

UN GUIDE COMPLET SUR L'APPROCHE SCRUM (SBOK[®] GUIDE)

Quatrième édition

Inclut deux chapitres sur le passage à l'échelle de
Scrum pour les gros projets et l'entreprise



Un Guide Complet sur l'Approche Scrum (*Guide SBOK®*)

Quatrième édition

**Inclut deux chapitres sur le passage à l'échelle de
Scrum pour les gros projets et l'entreprise**

Un guide complet pour réaliser des projets en utilisant l'approche Scrum

© 2022 SCRUMstudy®, une marque VMEdU, Inc. Tous droits réservés.

Library of Congress Cataloging-in-Publication Data

A Guide to the Scrum Body of Knowledge (Guide SBOK®) – Un guide complet sur l'approche Scrum – Quatrième édition

Références bibliographiques et index inclus.

1. Approche Scrum. I. SCRUMstudy®. II. *Guide SBOK®*

2013950625

Publié par :

SCRUMstudy®, une marque VMEdU, Inc.
12725 W. Indian School Road, Suite F-112
Avondale, Arizona 85392 – États-Unis
Email : sbok@scrumstudy.com
Site Web : www.scrumstudy.com

“SBOK”, le logo SCRUMstudy, “SFC”, “SDC”, “SMC”, “SAMC”, “SPOC”, “SSMC”, “SSPOC”, et “ESMC” sont des marques déposées SCRUMstudy® (une marque VMEdU, Inc.) Pour une liste complète des marques SCRUMstudy®, veuillez contacter le service juridique SCRUMstudy® (SCRUMstudy® Legal Department).

A Guide to the Scrum Body of Knowledge (Guide SBOK®) est destiné à des fins éducatives. SCRUMstudy® et VMEdU, Inc ne garantissent pas que ce guide soit adapté à d'autres fins et n'offre aucune garantie explicite ou implicite quelle qu'elle soit et n'accepte aucune responsabilité en cas d'erreurs ou d'omissions. Aucune responsabilité n'est acceptée en cas de dommages accidentels ou consécutifs relatifs ou survenant suite à l'utilisation des informations contenues ci-après.

SCRUMstudy® reçoit volontiers des corrections ou commentaires concernant ses livres. N'hésitez pas à envoyer vos commentaires concernant les erreurs typographiques, de formatage ou autres. Vous pouvez faire une copie de la page concernée dans le livre, identifier l'erreur et l'envoyer à l'adresse ci-dessus ou l'envoyer par email à sbok@scrumstudy.com.

Cet ouvrage ne peut être reproduit ou transmis sous quelque forme que ce soit ; électronique, manuelle, photocopie, enregistrement ou par quelque système de stockage et d'extraction d'information que ce soit, sans la permission préalable écrite de l'éditeur.

10 9 8 7 6 5 4 3 2

PRÉFACE

A Guide to the Scrum Body of Knowledge (Guide SBOK®) – Un guide complet sur l'approche Scrum fournit les orientations nécessaires à la mise en œuvre réussie de Scrum, l'approche agile la plus populaire pour le développement de produit et la réalisation de projets. Telle qu'elle est définie dans le *Guide SBOK®*, l'approche Scrum peut s'appliquer aux portefeuilles, programmes ou projets de toute taille et de tout niveau de complexité. Elle peut s'appliquer efficacement dans tout type de secteur afin de créer un produit, un service ou un autre résultat.

Le *Guide SBOK®* est destiné à être utilisé comme guide de référence ou comme manuel à la fois par les praticiens Scrum expérimentés et les autres praticiens de développement de produits ou de services, ainsi que par les personnes sans expérience ou connaissances préalables concernant la gestion de projet ou l'approche Scrum. Cette nouvelle édition du *Guide SBOK®* fournit des perspectives supplémentaires quant aux meilleures pratiques Scrum, en particulier concernant le passage à l'échelle. Deux chapitres ont été ajoutés au *Guide SBOK®* afin d'aborder spécifiquement le passage à l'échelle de Scrum pour les gros projets (chapitre 13) et le passage à l'échelle de Scrum pour l'entreprise (chapitre 14). Alors que la popularité et la mise en application de l'approche Scrum augmentent et évoluent mondialement, notre objectif est de partager les enseignements et les meilleures pratiques dans le cadre du *Guide SBOK®*.

Le *Guide SBOK®* exploite les connaissances et les observations accumulées au fil de milliers de projets, à travers différentes organisations et différents secteurs. Cette Quatrième édition vient compléter les contributions collectives d'experts Scrum et de spécialistes en matière de réalisation de projets. Plus particulièrement, le feedback reçu de la part de la communauté Scrum mondiale a joué un rôle important pour identifier les améliorations et ajouts à apporter au *Guide SBOK®*. Son développement a été un réel effort collaboratif de la part d'un grand nombre d'experts et praticiens issus de différentes disciplines.

L'adoption du cadre de travail du *Guide SBOK®* permet d'harmoniser la façon dont Scrum est mis en œuvre dans les organisations à travers le monde et aide considérablement ces dernières à améliorer leur retour sur capital investi. De plus, il encourage davantage de réflexion et de débat concernant l'application de Scrum à de nombreux types de projets. Ces projets contribueront à leur tour à l'élargissement et à l'enrichissement des connaissances et par conséquent aux futures mises à jour de ce guide.

Bien que le *Guide SBOK®* soit un guide et un cadre de travail complet pour la réalisation de projets en utilisant l'approche Scrum, son contenu est organisé afin d'en faciliter la consultation, et ce quelles que soient les connaissances préalables du lecteur sur le sujet. J'espère que chaque lecteur apprendra de ce livre et l'appréciera autant que ses nombreux auteurs et réviseurs ont appris et apprécié de rassembler les connaissances et le savoir collectifs qu'il contient.



Tridibesh Satpathy, Auteur principal, *Guide SBOK®*

TABLE DES MATIÈRES

| | | |
|-------|---|----|
| 1. | INTRODUCTION | 1 |
| 1.1 | Aperçu de l'approche Scrum..... | 2 |
| 1.1.1 | Un bref historique de l'approche Scrum..... | 3 |
| 1.2 | Pourquoi utiliser l'approche Scrum ? | 4 |
| 1.2.1 | Passage à l'échelle de Scrum | 5 |
| 1.3 | Objectifs du <i>Guide SBOK</i> ® | 6 |
| 1.4 | Cadre du <i>Guide SBOK</i> ® | 7 |
| 1.4.1 | Comment utiliser le <i>Guide SBOK</i> ® ?..... | 8 |
| 1.4.2 | Principes Scrum | 8 |
| 1.4.3 | Aspects Scrum | 10 |
| 1.4.4 | Processus Scrum | 15 |
| 1.4.5 | Scrum for Large Projects..... | 19 |
| 1.4.6 | Scrum for the Enterprise | 19 |
| 1.5 | L'approche Scrum comparée à la gestion de projet traditionnelle | 20 |
| 2. | PRINCIPLES..... | 21 |
| 2.1 | Introduction | 21 |
| 2.2 | Guide des rôles..... | 22 |
| 2.3 | Contrôle empirique des processus | 22 |
| 2.3.1 | Transparence..... | 23 |
| 2.3.2 | Inspection..... | 24 |
| 2.3.3 | Adaptation..... | 25 |
| 2.4 | Auto-organisation..... | 27 |
| 2.4.1 | Avantages de l'auto-organisation..... | 27 |
| 2.5 | Collaboration..... | 29 |
| 2.5.1 | Avantages de la collaboration dans les projets Scrum..... | 30 |
| 2.5.2 | Importance du regroupement dans la collaboration..... | 31 |
| 2.5.3 | Collaboration dans les équipes distribuées | 32 |
| 2.6 | Priorisation basée sur la valeur..... | 34 |
| 2.7 | Time-boxing | 36 |
| 2.7.1 | Time-boxes Scrum | 36 |
| 2.8 | Développement itératif | 38 |

| | | |
|--------|---|----|
| 2.9 | L'approche Scrum comparée à la gestion de projet traditionnelle | 40 |
| 3. | ORGANISATION | 41 |
| 3.1 | Introduction | 41 |
| 3.2 | Guide des rôles | 42 |
| 3.3 | Rôles des projets Scrum | 42 |
| 3.3.1 | Rôles principaux | 42 |
| 3.3.2 | Rôles non essentiels | 44 |
| 3.4 | Product Owner | 46 |
| 3.4.1 | Voix du client (VOC) | 48 |
| 3.5 | Scrum Master | 48 |
| 3.6 | Équipe Scrum | 50 |
| 3.6.1 | Sélection du personnel | 51 |
| 3.6.2 | Taille de l'équipe Scrum | 51 |
| 3.7 | Scrum dans les grands projets, les programmes et les portfolios | 52 |
| 3.7.1 | Fonctionnement de Scrum dans un grand projet | 52 |
| 3.7.2 | Rôles principaux supplémentaires dans les grands projets | 53 |
| 3.7.3 | Fonctionnement de Scrum dans un environnement d'entreprise | 55 |
| 3.7.4 | Rôles essentiels supplémentaires dans un environnement d'entreprise | 56 |
| 3.7.5 | Exemples de projets, programmes et portfolios | 58 |
| 3.8 | Résumé des responsabilités | 59 |
| 3.9 | L'approche Scrum comparée à la gestion de projet traditionnelle | 60 |
| 3.10 | Théories RH populaires et leur importance dans Scrum | 61 |
| 3.10.1 | Le modèle de Tuckman sur la dynamique des groupes | 61 |
| 3.10.2 | Gestion des conflits | 62 |
| 3.10.3 | Techniques de gestion des conflits | 63 |
| 3.10.4 | Styles de leadership | 64 |
| 3.10.5 | La pyramide des besoins de Maslow | 66 |
| 3.10.6 | Théorie X et théorie Y | 67 |
| 4. | JUSTIFICATION COMMERCIALE | 69 |
| 4.1 | Introduction | 69 |
| 4.2 | Guide des rôles | 70 |
| 4.3 | Livraison de la valeur ajoutée | 70 |

| | | |
|-------|--|----|
| 4.3.1 | Responsabilités du Product Owner concernant la justification commerciale | 72 |
| 4.3.2 | Responsabilités des autres rôles Scrum concernant la justification commerciale | 72 |
| 4.4 | Importance de la justification commerciale | 73 |
| 4.4.1 | Paramètres utilisés pour déterminer la justification commerciale | 74 |
| 4.4.2 | Justification commerciale et cycle du projet | 74 |
| 4.5 | Techniques de justification commerciale | 76 |
| 4.5.1 | Estimation de la valeur du projet..... | 77 |
| 4.5.2 | Planification de la valeur | 78 |
| 4.5.3 | Classement relatif des priorités | 81 |
| 4.5.4 | Représentation des Stories..... | 82 |
| 4.6 | Justification de valeur continue | 82 |
| 4.6.1 | Analyse de valeur acquise | 83 |
| 4.6.2 | Diagramme de flux cumulatif (DFC)..... | 85 |
| 4.7 | Confirmer la réalisation des bénéfiques | 86 |
| 4.7.1 | Prototypes, simulations et démonstrations..... | 86 |
| 4.8 | Résumé des responsabilités | 86 |
| 4.9 | L'approche Scrum comparée à la gestion de projet traditionnelle | 87 |
| 5. | QUALITÉ..... | 89 |
| 5.1 | Introduction | 89 |
| 5.2 | Guide des rôles | 90 |
| 5.3 | Définir la qualité | 90 |
| 5.3.1 | Qualité et portée | 90 |
| 5.3.2 | Qualité et valeur commerciale | 91 |
| 5.4 | Critères d'acceptation et backlog de produit priorisé | 92 |
| 5.4.1 | Écriture des critères d'acceptation | 93 |
| 5.4.2 | Définition de « Ready » (Prêt) | 94 |
| 5.4.3 | Définition de <i>terminé</i> | 95 |
| 5.4.4 | Définition minimum de <i>terminé</i> | 96 |
| 5.4.5 | Acceptation ou rejet des éléments du backlog de produit priorisé | 96 |
| 5.5 | Gestion de la qualité dans Scrum | 97 |
| 5.5.1 | Planification de la qualité | 97 |
| 5.5.2 | Assurance qualité et contrôle de la qualité | 99 |

| | | |
|-------|---|-----|
| 5.5.3 | Cycle Plan-Do-Check-Act (PDCA) | 100 |
| 5.6 | Résumé des responsabilités | 101 |
| 5.7 | L'approche Scrum comparée à la gestion de projet traditionnelle | 102 |
| 6. | CHANGEMENT | 103 |
| 6.1 | Introduction | 103 |
| 6.2 | Guide des rôles | 104 |
| 6.3 | Aperçu | 104 |
| 6.3.1 | Demandes de changement rejetées et approuvées | 105 |
| 6.4 | Le changement dans Scrum | 107 |
| 6.4.1 | Équilibrer flexibilité et stabilité | 107 |
| 6.4.2 | Atteindre la flexibilité | 107 |
| 6.5 | Intégration des changements | 113 |
| 6.5.1 | Changements apportés à un sprint | 113 |
| 6.6 | Changements apportés aux portefeuilles et aux programmes | 118 |
| 6.6.1 | Dans un portefeuille | 119 |
| 6.6.2 | Dans un programme | 119 |
| 6.7 | Résumé des responsabilités | 121 |
| 6.8 | L'approche Scrum comparée à la gestion de projet traditionnelle | 122 |
| 7. | RISQUES | 123 |
| 7.1 | Introduction | 123 |
| 7.2 | Guide des rôles | 124 |
| 7.3 | Qu'est-ce que le risque ? | 124 |
| 7.3.1 | Différences entre les risques et les problèmes | 124 |
| 7.3.2 | Attitude face aux risques | 125 |
| 7.4 | Procédure de gestion des risques | 126 |
| 7.4.1 | Identification des risques | 126 |
| 7.4.2 | Évaluation des risques | 127 |
| 7.4.3 | Priorisation des risques | 131 |
| 7.4.4 | Atténuation des risques | 132 |
| 7.4.5 | Communication des risques | 133 |
| 7.5 | Minimiser les risques avec l'approche Scrum | 134 |
| 7.6 | Les risques dans les portefeuilles et les programmes | 135 |

| | | |
|-------|---|-----|
| 7.6.1 | Dans un portefeuille | 135 |
| 7.6.2 | Dans un programme | 136 |
| 7.7 | Résumé des responsabilités | 137 |
| 7.8 | L'approche Scrum comparée à la gestion de projet traditionnelle | 138 |
| 8. | INITIATION | 139 |
| 8.1 | Création de la vision de projet | 143 |
| 8.1.1 | Contributions | 145 |
| 8.1.2 | Outils | 146 |
| 8.1.3 | Résultats | 148 |
| 8.2 | Identification du Scrum Master et des parties prenantes de l'entreprise | 149 |
| 8.2.1 | Contributions | 152 |
| 8.2.2 | Outils | 154 |
| 8.2.3 | Résultats | 155 |
| 8.3 | Formation de l'équipe Scrum | 156 |
| 8.3.1 | Contributions | 157 |
| 8.3.2 | Outils | 159 |
| 8.3.3 | Résultats | 160 |
| 8.4 | Développement des epics | 162 |
| 8.4.1 | Contributions | 163 |
| 8.4.2 | Outils | 166 |
| 8.4.3 | Résultats | 169 |
| 8.5 | Création du backlog de produit priorisé | 170 |
| 8.5.1 | Contributions | 172 |
| 8.5.2 | Outils | 173 |
| 8.5.3 | Résultats | 177 |
| 8.6 | Planification de la release | 179 |
| 8.6.1 | Contributions | 180 |
| 8.6.2 | Outils | 182 |
| 8.6.3 | Résultats | 183 |
| 8.7 | Lancement de la phase de diagramme de flux de données | 184 |
| 9. | PLANIFICATION ET ESTIMATION | 185 |
| 9.1 | Création des user stories | 190 |

| | | |
|--------|---|-----|
| 9.1.1 | Contributions..... | 191 |
| 9.1.2 | Outils | 193 |
| 9.1.3 | Résultats | 195 |
| 9.2 | Estimation des user stories..... | 196 |
| 9.2.1 | Contributions..... | 198 |
| 9.2.2 | Outils | 199 |
| 9.2.3 | Résultats | 202 |
| 9.3 | Validation des user stories..... | 202 |
| 9.3.1 | Contributions..... | 203 |
| 9.3.2 | Outils | 204 |
| 9.3.3 | Résultats | 205 |
| 9.4 | Identification des tâches | 207 |
| 9.4.1 | Contributions..... | 208 |
| 9.4.2 | Outils | 209 |
| 9.4.3 | Résultats | 211 |
| 9.5 | Estimation des Tâches..... | 213 |
| 9.5.1 | Contributions..... | 214 |
| 9.5.2 | Outils | 216 |
| 9.5.3 | Résultats | 217 |
| 9.6 | Mise à jour du Sprint Backlog | 217 |
| 9.6.1 | Contributions..... | 218 |
| 9.6.2 | Outils | 219 |
| 9.6.3 | Résultats | 220 |
| 9.7 | Diagramme de flux de données de la phase de planification et d'estimation | 222 |
| 10. | IMPLÉMENTATION | 223 |
| 10.1 | Création des éléments livrables..... | 227 |
| 10.1.1 | Contributions..... | 229 |
| 10.1.2 | Outils | 232 |
| 10.1.3 | Résultats | 233 |
| 10.2 | Organisation des mêlées quotidiennes..... | 235 |
| 10.2.1 | Contributions..... | 236 |
| 10.2.2 | Outils | 238 |

| | | |
|--------|--|-----|
| 10.2.3 | Résultats | 239 |
| 10.3 | Raffinage du Product Backlog priorisé..... | 241 |
| 10.3.1 | Contributions..... | 242 |
| 10.3.2 | Outils | 244 |
| 10.3.3 | Résultats | 245 |
| 10.4 | Diagramme de flux de données de la phase d'implémentation | 246 |
| 11. | REVUE ET RÉTROSPECTIVE | 247 |
| 11.1 | Démonstration et validation du sprint..... | 250 |
| 11.1.1 | Contributions..... | 252 |
| 11.1.2 | Outils | 254 |
| 11.1.3 | Résultats | 255 |
| 11.2 | Rétrospective de sprint | 257 |
| 11.2.1 | Contributions..... | 258 |
| 11.2.2 | Outils | 258 |
| 11.2.3 | Résultats | 261 |
| 11.3 | Diagramme de flux de données de la phase de revue et de rétrospective..... | 262 |
| 12. | RELEASE | 263 |
| 12.1 | Livraison des produits | 265 |
| 12.1.1 | Contributions..... | 266 |
| 12.1.2 | Outils | 268 |
| 12.1.3 | Résultats | 268 |
| 12.2 | Rétrospective de release | 269 |
| 12.2.1 | Contributions..... | 270 |
| 12.2.2 | Outils | 271 |
| 12.2.3 | Résultats | 272 |
| 12.3 | Diagramme de flux de données de la phase de release | 273 |
| 13. | MISE À L'ÉCHELLE DE SCRUM POUR LES GRANDS PROJETS..... | 275 |
| 13.1 | Impact des grands projets sur les processus fondamentaux Scrum | 277 |
| 13.1.1 | Lancement | 277 |
| 13.1.2 | Planification et estimation..... | 283 |
| 13.1.3 | Implémentation..... | 286 |
| 13.1.4 | Révision et rétrospective | 289 |

| | | |
|--------|---|-----|
| 13.1.5 | Release | 291 |
| 13.2 | Inputs et outputs supplémentaires pour les grands projets | 293 |
| 13.2.1 | Organisation de Scrum pour les grands projets* | 293 |
| 13.2.2 | Plan de collaboration des Product Owners* | 293 |
| 13.2.3 | Plan de collaboration des Scrum Masters* | 294 |
| 13.2.4 | Ressources partagées* | 294 |
| 13.2.5 | Spécialisation de l'équipe* | 295 |
| 13.2.6 | Environnement et calendrier de l'environnement* | 296 |
| 13.2.7 | Plan de préparation de release* | 296 |
| 13.3 | Outils supplémentaires pour les grands projets | 296 |
| 13.3.1 | Plan de communication pour les grands projets | 296 |
| 13.3.2 | Planification des ressources pour les grands projets* | 297 |
| 13.3.3 | Identification de l'environnement* | 298 |
| 13.3.4 | Affectation du Product Backlog priorisé* | 298 |
| 13.3.5 | Réunion Scrum of Scrums (SoS)* | 299 |
| 13.3.6 | Méthodes de préparation de release (Release Preparation Methods)* | 300 |
| 13.3.7 | Sprint de préparation de release | 300 |
| 13.3.8 | Outil de projet Scrum | 301 |
| 14. | La mise à l'échelle de Scrum pour l'entreprise | 303 |
| 14.1 | Impact des programmes ou des portfolios sur les processus fondamentaux Scrum | 305 |
| 14.1.1 | Lancement | 305 |
| 14.1.2 | Planification et estimation | 309 |
| 14.1.3 | Implémentation | 310 |
| 14.1.4 | Révision et rétrospective | 310 |
| 14.1.5 | Release | 311 |
| 14.2 | Processus supplémentaires pour mettre Scrum à l'échelle de l'entreprise (programme/portfolio) | 312 |
| 14.3 | Création/mise à jour d'équipes pour les programmes ou portfolios | 316 |
| 14.3.1 | Inputs | 317 |
| 14.3.2 | Outils | 318 |
| 14.3.3 | Outputs | 318 |
| 14.4 | Création/mise à jour des éléments de programme ou de portfolio | 320 |
| 14.4.1 | Inputs | 321 |

| | | |
|---|---|-----|
| 14.4.2 | Outils | 322 |
| 14.4.3 | Outputs..... | 324 |
| 14.5 | Révision et mise à jour du Scrum Guidance Body..... | 326 |
| 14.5.1 | Inputs..... | 327 |
| 14.5.2 | Outils | 327 |
| 14.5.3 | Outputs..... | 328 |
| 14.6 | Création/raffinage du Program ou Portfolio Backlog priorisé..... | 329 |
| 14.6.1 | Inputs..... | 331 |
| 14.6.2 | Outils | 333 |
| 14.6.3 | Outputs..... | 335 |
| 14.7 | Création/mise à jour de release de programme ou de portfolio..... | 336 |
| 14.7.1 | Inputs..... | 337 |
| 14.7.2 | Outils | 338 |
| 14.7.3 | Outputs..... | 339 |
| 14.8 | Rétrospective de release de programme ou de portfolio | 340 |
| 14.8.1 | Inputs..... | 340 |
| 14.8.2 | Outils | 342 |
| 14.8.3 | Outputs..... | 342 |
| ANNEXE A. UN APERÇU DU CONCEPT AGILE | | 343 |
| ANNEXE B. AUTEURS ET CONTRIBUTEURS DU <i>GUIDE SBOK</i> ® | | 353 |
| ANNEXE C. MISES À JOUR DE LA QUATRIÈME ÉDITION | | 357 |
| GLOSSAIRE | | 361 |
| INDEX..... | | 405 |

LIST OF FIGURES

| | |
|--|-----|
| Figure 1-1: Flux de l'approche Scrum pour un sprint..... | 2 |
| Figure 1-2: Cadre du Guide SBOK® | 7 |
| Figure 1-3: Principes Scrum..... | 9 |
| Figure 1-4: Organisation de Scrum | 12 |
| Figure 2-1: La transparence dans Scrum | 24 |
| Figure 2-2: Inspection dans Scrum | 25 |
| Figure 2-3: L'adaptation dans Scrum | 26 |
| Figure 2-4: Les défis relatifs à la gestion de projet traditionnelle | 27 |
| Figure 2-5: Les objectifs d'une équipe autoorganisée | 29 |
| Figure 2-6: Les avantages de la collaboration dans les projets Scrum..... | 31 |
| Figure 2-7: Priorisation basée sur la valeur..... | 35 |
| Figure 2-8: Durées des réunions Scrum | 38 |
| Figure 2-9: L'approche Scrum comparée à l'approche en cascade traditionnelle | 39 |
| Figure 3-1: Aperçu des rôles Scrum | 44 |
| Figure 3-2: Traits désirables pour les rôles Scrum principaux | 51 |
| Figure 3-3: Questions posées lors d'une réunion Scrum of Scrums..... | 54 |
| Figure 3-4: Scrum dans les organisations pour les projets, programmes ou portefeuilles | 59 |
| Figure 3-5: Les phases du modèle de Tuckman sur le développement des groupes | 62 |
| Figure 3-6: La pyramide des besoins de Maslow..... | 66 |
| Figure 4-1: Livraison de la valeur ajoutée dans Scrum comparée aux projets traditionnels | 71 |
| Figure 4-3: Justification commerciale et cycle du projet | 76 |
| Figure 4-4: Représentation de la chaîne de valeur | 79 |
| Figure 4-5: Analyse Kano | 81 |
| Figure 4-6: Représentation des Stories..... | 82 |
| Figure 4-7: Exemple de diagramme de flux cumulatif (DFC) | 85 |
| Figure 5-1: Diagramme de flux des incréments du projet..... | 93 |
| Figure 5-2: Définition de terminé en cascade | 96 |
| Figure 5-3: Définition de terminé en cascade | 100 |
| Figure 6-1: Exemple de processus d'approbation des changements..... | 106 |
| Figure 6-2: Mise à jour du backlog de produit priorisé avec les changements approuvés | 106 |
| Figure 6-3: Caractéristiques Scrum pour atteindre la flexibilité..... | 108 |
| Figure 6-4: Motivations des parties prenantes de l'entreprise pour demander des changements..... | 109 |
| Figure 6-5: Motivations de l'équipe Scrum principale pour effectuer des demandes de changement | 110 |
| Figure 6-6: Intégration des changements dans Scrum..... | 114 |
| Figure 6-7: Impact des changements prévus sur la durée du sprint | 116 |
| Figure 6-8: Intégrer les changements dans les portefeuilles et les programmes | 120 |
| Figure 7-1: Exemple d'arbre de probabilité | 128 |
| Figure 7-2: Exemple de diagramme de Pareto | 129 |
| Figure 7-3: Exemple de grille de probabilité et d'impact | 130 |
| Figure 7-4: Processus de priorisation des risques | 131 |
| Figure 7-5: Exemple de risk burndown chart | 134 |
| Figure 7-6: Gestion des risques dans les portefeuilles et les programmes | 136 |
| Figure 8-1 : Aperçu de la phase d'initiation | 141 |

| | |
|---|-----|
| Figure 8-2 : Aperçu de la phase d'initiation (éléments essentiels)..... | 142 |
| Figure 8-3 : Création de la vision de projet — Contributions, outils et résultats | 143 |
| Figure 8-4 : Création de la vision de projet — diagramme de flux de données | 144 |
| Figure 8-5 : Processus de l'analyse des lacunes | 148 |
| Figure 8-6 : Identification du Scrum Master et des parties prenantes de l'entreprise — Contributions, outils et résultats | 150 |
| Figure 8-7 : Identification du Scrum Master et des parties prenantes de l'entreprise — Diagramme de flux de données | 151 |
| Figure 8-8 : Formation de l'équipe Scrum — Contributions, outils et résultats | 156 |
| Figure 8-9 : Formation de l'équipe Scrum — Diagramme de flux de données | 157 |
| Figure 8-10 : Développement des epics — Contributions, outils et résultats | 162 |
| Figure 8-11 : Développement des epics — Diagramme de flux de données | 163 |
| Figure 8-12 : Création du backlog de produit priorisé — Contributions, outils et résultats | 171 |
| Figure 8-13 : Création du backlog de produit priorisé — Diagramme de flux de données | 171 |
| Figure 8-14 : Planification de la release — Contributions, outils et résultats | 179 |
| Figure 8-15 : Planification de la release — Diagramme de flux de données | 180 |
| Figure 8-16 : Phase d'initiation — Diagramme de flux de données..... | 184 |
| Figure 9-1 : Aperçu de la phase de planification et d'estimation | 188 |
| Figure 9-2 : Aperçu de la phase de planification et d'estimation (éléments essentiels)..... | 189 |
| Figure 9-3 : Création des user stories — Contributions, outils et résultats | 190 |
| Figure 9-4 : Création des user stories — diagramme de flux de données..... | 191 |
| Figure 9-5 : Estimation des user stories — Contributions, outils et résultats | 197 |
| Figure 9-6 : Estimation des user stories — diagramme de flux de données | 197 |
| Figure 9-7 : Validation des user stories — Contributions, outils et résultats..... | 202 |
| Figure 9-8 : Validation des user stories — diagramme de flux de données | 203 |
| Figure 9-9: Scrumboard classique | 206 |
| Figure 9-10: Scrumboard avec quatre sections | 206 |
| Figure 9-11 : Identification des tâches — Contributions, outils et résultats | 207 |
| Figure 9-12 : Identification des tâches — Diagramme de flux de données..... | 208 |
| Figure 9-13: Scrumboard with Identified Tasks..... | 212 |
| Figure 9-14 : Estimation des tâches — Contributions, outils et résultats | 213 |
| Figure 9-15 : Estimation des tâches — Diagramme de flux de données | 214 |
| Figure 9-16: Mise à jour du Sprint Backlog — Contributions, outils et résultats..... | 217 |
| Figure 9-17 : Mise à jour du Sprint Backlog — Diagramme de flux de données..... | 218 |
| Figure 9-18: Sprint Burndown Chart..... | 221 |
| Figure 9-19: Sprint Burnup Chart | 222 |
| Figure 9-20 : Phase de planification et d'estimation — Diagramme de flux de données | 222 |
| Figure 10-1 : Aperçu de la phase d'implémentation | 225 |
| Figure 10-2 : Aperçu de la phase d'implémentation (éléments essentiels) | 226 |
| Figure 10-3 : Création des éléments livrables — Contributions, outils et résultats | 227 |
| Figure 10-4 : Création des éléments livrables — Diagramme de flux de données | 228 |
| Figure 10-5: Scrumboard | 230 |
| Figure 10-6 : Mêlées quotidiennes — Contributions, outils et résultats | 235 |
| Figure 10-7 : Mêlées quotidiennes — Diagramme de flux de données | 236 |
| Figure 10-8 : Raffinage du Product Backlog priorisé — Contributions, outils et résultats | 241 |
| Figure 10-9: Raffinage du Product Backlog priorisé — Diagramme de flux de données..... | 242 |

| | |
|---|-----|
| Figure 10-10 : Phase d'implémentation — Diagramme de flux de données..... | 246 |
| Figure 11-1 : Aperçu de la phase de revue et de rétrospective | 249 |
| Figure 11-2 : Aperçu de la phase de revue et rétrospective (éléments essentiels) | 250 |
| Figure 11-3 : Démonstration et validation du sprint — Contributions, outils et résultats | 251 |
| Figure 11-4 : Démonstration et validation du sprint — Diagramme de flux de données | 252 |
| Figure 11-5 : Rétrospective de sprint — Contributions, outils et résultats..... | 257 |
| Figure 11-6 : Rétrospective de sprint — Diagramme de flux de données..... | 257 |
| Figure 11-7 : Phase de revue et rétrospective — Diagramme de flux de données..... | 262 |
| Figure 12-1 : Aperçu de la phase de release | 264 |
| Figure 12-2 : Aperçu de la phase de release (éléments essentiels) | 265 |
| Figure 12-3 : Livraison des produits — Contributions, outils et résultats..... | 265 |
| Figure 12-4 : Livraison des produits — Diagramme de flux de données..... | 266 |
| Figure 12-5 : Rétrospective de projet — Contributions, outils et résultats | 269 |
| Figure 12-6 : Rétrospective de projet — Diagramme de flux de données | 270 |
| Figure 12-7 : Phase de release — Diagramme de flux de données..... | 273 |
| Figure 14-1: La mise à l'échelle de Scrum pour l'entreprise..... | 315 |
| Figure 14-2: Create/Update Program or Portfolio Teams—Inputs, Tools, and Outputs..... | 316 |
| Figure 14-3 : Création/mise à jour des éléments de programme ou de portfolio—inputs, outils et outputs... | 320 |
| Figure 14-4 : Réunion Scrum of Scrums of Scrums (SoSoS)..... | 323 |
| Figure 14-5 : Révision et mise à jour du Scrum Guidance Body—inputs, outils et outputs | 326 |
| Figure 14-6 : Création/raffinage du Program ou Portfolio Backlog priorisé—inputs, outils et outputs | 330 |
| Figure 14-7: Création/mise à jour de release de programme ou de portfolio—inputs, outils et outputs | 336 |
| Figure 14-8 : Rétrospective de release de programme ou de portfolio—inputs, outils et outputs..... | 340 |

LIST OF TABLEAU

| | |
|--|-----|
| Tableau 1-1: Résumé des processus Scrum fondamentaux | 15 |
| Tableau 1-2: Réunions et processus Scrum..... | 19 |
| Tableau 1-3: L'approche Scrum comparée à la gestion de projet traditionnelle | 20 |
| Tableau 3-1: Responsabilités du Product Owner dans les processus Scrum..... | 47 |
| Tableau 3-2: Responsabilités du Scrum Master dans les processus Scrum..... | 49 |
| Tableau 3-3: Responsabilités de l'équipe Scrum dans les processus Scrum | 50 |
| Tableau 3-4: Résumé des responsabilités pertinentes à l'organisation | 60 |
| Tableau 4-1: Hiérarchie des responsabilités concernant la justification commerciale..... | 72 |
| Tableau 4-2: Formules de valeur acquise | 83 |
| Tableau 4-3: Résumé des responsabilités correspondant à la justification commerciale | 87 |
| Tableau 5-1: Résumé des responsabilités correspondant à la qualité | 102 |
| Tableau 6-1: Résumé des responsabilités relatives aux changements | 121 |
| Tableau 7-1: Résumé des responsabilités correspondant aux risques | 137 |
| Tableau 13-1: Impact des grands projets sur les processus Scrum fondamentaux - Phase d'initiation..... | 283 |
| Tableau 13-2: Impact des grands projets sur les processus Scrum fondamentaux - Phase de planification et d'estimation | 285 |
| Tableau 13-3: Impact des grands projets sur les processus fondamentaux Scrum—phase d'implémentation | 288 |
| Tableau 13-4: Impact des grands projets sur les processus Scrum fondamentaux - Phase de révision et de rétrospection..... | 290 |
| Tableau 13-5: Impact des grands projets sur les processus Scrum fondamentaux - Phase de Release..... | 292 |
| Tableau 14-1: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase d'initiation..... | 309 |
| Tableau 14-2: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase de Implémentation..... | 310 |
| Tableau 14-3: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase de révision et de rétrospective | 311 |
| Tableau 14-4: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase de release | 311 |

1. INTRODUCTION

A Guide to the Scrum Body of Knowledge (Guide SBOK®) – Un guide complet sur l'approche Scrum fournit les orientations nécessaires à la mise en œuvre réussie de Scrum, l'approche Agile la plus populaire pour la gestion de projet et le développement de produit. Ce guide présente un cadre de travail global qui inclut les principes, les aspects et les processus de Scrum.

Telle qu'elle est définie dans le *Guide SBOK®*, l'approche Scrum peut s'appliquer aux situations suivantes :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou autres résultats destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* tel qu'il est utilisé dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat d'un projet. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce premier chapitre décrit l'objectif et le cadre du *Guide SBOK®* et introduit les principaux concepts de Scrum. Il contient un résumé des principes de l'approche Scrum, de ses aspects et de ses processus. Le chapitre 2 développe les six principes Scrum fondamentaux sur lesquels s'appuie l'approche Scrum. Les chapitres 3 à 7 développent les cinq aspects Scrum devant être pris en compte quel que soit le projet : l'organisation, la justification commerciale, la qualité, le changement et le risque. Les chapitres 8 à 12 traitent des 19 processus Scrum impliqués dans la réalisation d'un projet Scrum. Ces processus font partie des cinq phases Scrum : Initiation, planification & estimation, implémentation, revue & rétrospective, et release. Ces phases détaillent les contributions et résultats associés à chaque processus ainsi que les différents outils qui peuvent être utilisés pour chacun. Certaines contributions, outils et résultats sont obligatoires et sont désignés comme tels, d'autres sont facultatifs et dépendent de la taille d'un projet donné, des exigences organisationnelles et/ou des directives énoncées par le Scrum Guidance Body (SGB) de l'organisation. Les chapitres 13 à 14 constituent un nouvel ajout au *Guide SBOK®*. Ils offrent des indications sur le passage à l'échelle de Scrum pour les gros projets et pour les entreprises.

Ce chapitre est divisé comme suit :

1.1 Aperçu de l'approche Scrum

1.2 Pourquoi utiliser l'approche Scrum ?

1.3 Objectifs du *Guide SBOK®*

1.4 Cadre du *Guide SBOK®*

1.5 L'approche Scrum comparée à la gestion de projet traditionnelle

1.1 Aperçu de l'approche Scrum

Un projet Scrum implique un effort de collaboration visant à créer un nouveau produit, service ou tout autre résultat, tel qu'il a été défini dans l'énoncé de vision du projet. Les projets sont sujets à des contraintes de temps, de coûts, de portée, de qualité, de ressources, de capacités organisationnelles et à d'autres limitations qui les rendent difficiles à planifier, à exécuter, à gérer et finalement à aboutir. Cependant, la mise en œuvre réussie des résultats d'un projet terminé offre à une organisation des avantages commerciaux non négligeables. Il est donc nécessaire pour les organisations de choisir et d'appliquer une approche de gestion de projet adaptée.

L'approche Scrum est l'une des méthodes agile les plus populaires. C'est un cadre adaptable, itératif, rapide, flexible et efficace conçu pour offrir une valeur ajoutée significative rapidement et tout au long d'un projet. Bien que le cadre Scrum, tel qu'il est défini dans le guide SBOK®, soit principalement utilisé pour la réalisation de projets est principalement utilisé pour réaliser des projets et créer des produits, Scrum peut également être utilisé pour gérer la maintenance continue des produits et services, pour suivre les problèmes et pour gérer les changements.

Scrum garantit la transparence de la communication et crée un environnement de responsabilité collective et de progression continue. Tel qu'il est défini dans le *Guide SBOK®*, le cadre Scrum est structuré de façon à supporter le développement de produits et de services pour tout secteur d'activité et tout type de projet, quelle que soit sa complexité.

L'une des principales for

ces de l'approche Scrum repose sur des équipes polyvalentes, autoorganisées et responsables qui répartissent leur temps sur des cycles de travail courts et intenses appelés *sprints*. La figure 1-1 présente un aperçu du flux d'un projet Scrum.

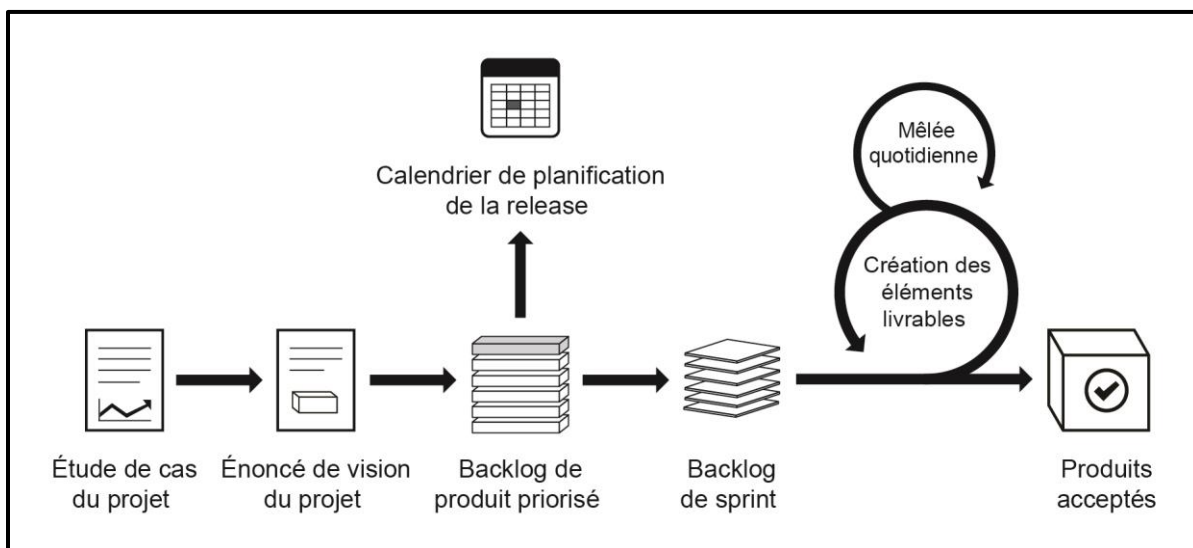


Figure 1-1: Flux de l'approche Scrum pour un sprint

Le cycle Scrum débute par une réunion des parties prenantes de l'entreprise durant laquelle la vision du projet est créée. Le Product Owner développe ensuite un backlog de produit priorisé écrit à partir des user stories et contenant une liste priorisée des exigences commerciales et des exigences du projet.

Chaque sprint débute par une réunion de planification de sprint durant laquelle les user stories prioritaires sont étudiées en vue d'être incluses dans le sprint. Un sprint dure généralement entre une et quatre semaines et engage le travail de l'équipe Scrum afin de créer des produits potentiellement livrables ou des incréments de produit. Pendant le sprint des mêlées quotidiennes, des réunions courtes et très ciblées sont organisées avec les membres de l'équipe où chacun explique ses progrès quotidiens. Le Product Owner peut évaluer les livrables terminés pendant le Sprint et peut accepter les livrables qui répondent aux critères d'acceptation prédéfinis. Vers la fin du sprint, une réunion de revue de sprint est organisée, durant laquelle le Product Owner et les parties prenantes de l'entreprise concernées assistent à une démonstration des produits. Le Product Owner accepte les produits uniquement si ceux-ci remplissent les critères d'acceptation prédéfinis. Le cycle de sprint se conclut par une réunion de rétrospective du sprint durant laquelle l'équipe discute des moyens d'améliorer les processus et ses performances pour le sprint suivant.

1.1.1 Un bref historique de l'approche Scrum

Au milieu des années 80, Hirotaka Takeuchi et Ikujiro Nonaka définissent une stratégie de développement de produit flexible et multidimensionnelle où l'équipe de développement travaille en groupe afin d'atteindre un objectif commun. Ils décrivent une stratégie de développement de produit innovante qu'ils appellent approche holistique ou « rugby », « où toute l'équipe essaie de parcourir la distance en se faisant passer la balle ». Leur approche est basée sur des études de cas de fabrication pour différents secteurs d'activités. Takeuchi et Nonaka proposent que le développement de produits ne ressemble pas à une course en relais séquentielle, mais ressemble d'avantage au jeu du rugby où les membres de l'équipe travaillent ensemble, se passant la balle tout en se déplaçant en groupe sur le terrain. Le concept de mêlée de rugby (« scrum » en anglais) (phase de jeu durant laquelle un groupe de joueurs se forme pour recommencer le jeu) a été introduit à cette approche afin de décrire la proposition des auteurs selon laquelle le développement d'un produit implique « l'avancement de la mêlée sur le terrain ».

Depuis lors, de nombreux spécialistes, experts et auteurs continuent à affiner la conceptualisation et le cadre de Scrum sur la base des meilleures pratiques. Une étape importante dans le parcours de Scrum a été la création de la version originale du *guide SBOK®* en 2013. Au fil du temps, le *guide SBOK®* a été continuellement amélioré sur la base des révisions et des commentaires fournis par plusieurs milliers de praticiens Scrum et Agile, y compris plus de 5000 professeurs dans plus de 110 pays qui enseignent les pratiques Scrum et Agile en utilisant le cadre du *guide SBOK®*. Cette quatrième édition est le fruit de nouveaux raffinages qui garantissent que le *guide SBOK®* reste valide et pertinent dans un monde en constante évolution.

Le *guide SBOK®* est désormais la norme industrielle pour les entreprises et les praticiens intéressés par l'implémentation des pratiques Scrum ou Agile. Ces dernières années, l'approche Scrum a gagné en popularité et elle est maintenant l'approche de développement de projet privilégiée pour de nombreuses organisations dans le monde entier.

Pour faciliter son application dans des contextes multinationaux, le *guide SBOK®* a été traduit en plusieurs langues, notamment en espagnol, portugais, français, italien, arabe, chinois et japonais. Pour plus d'informations sur l'accès aux versions traduites, veuillez consulter le site www.scrumstudy.com.

1.2 Pourquoi utiliser l'approche Scrum ?

Ci-dessous figurent certains des avantages fondamentaux à l'application de Scrum quel que soit le projet :

1. **Adaptabilité** — le contrôle empirique des processus et la livraison en itération rendent les projets adaptables et ouverts à l'intégration de changements.
2. **Transparence** — tous les éléments contenant des informations, tels que le scrumboard ou le sprint burndown chart sont partagés, créant ainsi un environnement de travail ouvert.
3. **Feedback continu** — le feedback continu est communiqué lors des *mêlées quotidiennes* et du processus de *démonstration et validation du sprint*.
4. **Progression continue** — les produits sont améliorés progressivement sprint après sprint à travers le processus *Raffinage du Product Backlog priorisé*.
5. **Livraison continue de la valeur ajoutée** — le processus itératif de *livraison des produits* permet la livraison continue de valeur ajoutée au client aussi souvent que nécessaire.
6. **Rythme soutenable** — les processus Scrum sont conçus de façon à permettre aux personnes impliquées de travailler à un rythme soutenable, qu'elles pourraient, en théorie, maintenir indéfiniment.
7. **Livraison anticipée de forte valeur ajoutée** — la *création du backlog de produit priorisé* permet de garantir que les exigences à plus forte valeur pour le client sont satisfaites en priorité.
8. **Processus de développement efficace** — le time-boxing et la minimisation des tâches non-essentielles permettent d'atteindre un meilleur niveau d'efficacité.
9. **Motivation** — la motivation des employés est accrue par les *mêlées quotidiennes* et le processus de *rétrospective de sprint*.
10. **Résolution rapide des problèmes** — la collaboration et le regroupement des équipes polyvalentes permettent une résolution plus rapide des problèmes.
11. **Produits performants** — la *création d'un carnet de produit priorisé* et les revues régulières après la création des produits garantissent la livraison de produits performants au client.

12. **Orientation vers le client** — L'accent placé sur la valeur commerciale et l'approche collaborative vis-à-vis des parties prenantes de l'entreprise garantissent une stratégie axée sur le client.
13. **Environnement de confiance élevée** — les *mêlées quotidiennes* et le processus de *rétrospective de sprint* encouragent la transparence et la collaboration, permettant ainsi de développer un environnement de confiance élevée et de réduire les frictions entre les employés.
14. **Responsabilité collective** — le processus de *validation des user stories* permet aux membres de l'équipe de s'approprier le projet et leur travail, permettant ainsi d'améliorer la qualité de travail.
15. **Haute vélocité** — un cadre collaboratif permet à des équipes polyvalentes très compétentes de déployer tout leur potentiel et d'atteindre un haut niveau de vélocité.
16. **Environnement innovant** — les processus de *rétrospective de sprint* et de *rétrospective de projet* créent des conditions d'introspection, d'apprentissage et d'adaptabilité menant ainsi à un environnement de travail innovant et créatif.

1.2.1 Passage à l'échelle de Scrum

Dans l'idéal, une équipe Scrum, pour être efficace, se compose de six à dix membres. Cette pratique est sans doute à l'origine de l'idée reçue selon laquelle Scrum ne s'applique qu'à de petits projets. Cependant, elle peut facilement être mise à l'échelle et appliquée efficacement à des projets plus larges, à des programmes ou à des portefeuilles. Dans le cas où la taille de l'équipe Scrum est supérieure à dix personnes, plusieurs équipes Scrum peuvent être formées afin de collaborer sur le projet. La logique des consignes et principes de cette approche peut s'appliquer à des projets de toute envergure, quelle que soit la zone géographique ou l'organisation. Les gros projets peuvent impliquer plusieurs équipes Scrum travaillant en parallèle. Ils nécessitent la synchronisation et la facilitation des flux d'informations ainsi que l'amélioration de la communication. Les projets importants et complexes sont souvent réalisés dans le cadre d'un programme ou d'un portefeuille.

Les informations sur le passage à l'échelle de Scrum pour les gros projets sont disponibles dans le chapitre 13 et celles sur le passage à l'échelle pour l'entreprise dans le chapitre 14.

1.3 Objectifs du *Guide SBOK*[®]

Le cadre Scrum s'est avéré être le cadre préféré pour la réalisation de projets afin d'apporter systématiquement une grande valeur commerciale et d'améliorer le retour sur investissement. L'importance accordée par Scrum à la livraison de valeur ajoutée encourage l'équipe Scrum à fournir des résultats aussi tôt que possible lors d'un projet.

Le *Guide SBOK*[®] a été conçu dans le but de créer un guide essentiel pour aider les organisations et les praticiens en gestion de projet qui souhaitent appliquer Scrum, ainsi que pour accompagner ceux qui pratiquent déjà Scrum et souhaitent améliorer leurs processus. Ce guide est basé sur l'expérience de milliers de projets à travers de nombreux secteurs d'activités et organisations. De nombreux experts Scrum et praticiens en gestion de projet ont contribué à son développement.

Le *Guide SBOK*[®] est particulièrement utile :

- pour les membres de l'équipe Scrum principale, notamment :
 - les Product Owners souhaitant comprendre pleinement l'approche Scrum, plus particulièrement les questions relatives aux clients et aux parties prenantes de l'entreprise ainsi qu'à la justification commerciale, à la qualité, au changement et au risque associés aux projets Scrum.
 - les Scrum Masters souhaitant apprendre leur rôle spécifique concernant la supervision de la mise en application de Scrum dans des projets.
 - les membres de l'équipe Scrum souhaitant mieux comprendre les processus Scrum et les outils associés disponibles afin de créer le produit ou le service d'un projet.
- comme guide complet pour tous les praticiens travaillant sur des projets Scrum quel que soit le secteur d'activité ou l'organisation.
- comme source de références pour toute personne travaillant en relation avec l'équipe Scrum principale, notamment le Product Owner du portefeuille, le Scrum Master du portefeuille, le Product Owner du programme, le Scrum Master du programme, le Scrum Guidance Body et Parties prenantes de l'entreprise (c-à-d le sponsor, le client et les utilisateurs).
- comme manuel pour toute personne sans aucune expérience ou connaissance préalable de l'approche Scrum souhaitant en apprendre davantage à ce sujet.

Le contenu du *Guide SBOK*[®] est également utile aux personnes se préparant à passer un des examens de certification SCRUMstudy[®] suivant :

- Scrum Developer Certified (SDC[®])
- Scrum Master Certified (SMC[®])
- Scaled Scrum Master Certified (SSMC[®])
- SCRUMstudy Agile Master Certified (SAMC[®])
- Product Owner Scrum Certifié – Scrum Product Owner Certified (SPOC[®])
- Scaled Scrum Product Owner Certified (SSPOC[®])
- Expert Scrum Master Certified (ESMC[®])

1.4 Cadre du *Guide SBOK*[®]

Le *Guide SBOK*[®] est composé de trois domaines de base :

1. les **principes** sont abordés dans le chapitre 2. Celui-ci passe en revue les six principes fondateurs sur lesquels l'approche Scrum est basée.
2. les **aspects** sont abordés dans les chapitres 3 à 7. Ces chapitres décrivent les cinq aspects qui constituent les éléments fondamentaux pour tous les projets Scrum.
3. les **processus** sont abordés dans les chapitres 8 à 12. Ces chapitres incluent les dix-neuf processus Scrum fondamentaux et les contributions, outils et résultats qui leur sont associés. Le chapitre 13 couvre les inputs, les outils et les outputs supplémentaires nécessaires pour adapter Scrum aux grands projetstandis que le chapitre 14 traite des processus supplémentaires nécessaires pour adapter Scrum aux besoins de l'entreprise.

La figure 1-2 illustre le cadre du guide SBOK[®] et montre comment les principes, les aspects et les processus interagissent les uns avec les autres—ces trois éléments sont d'égale importance pour la compréhension et l'application du cadre du Scrum.

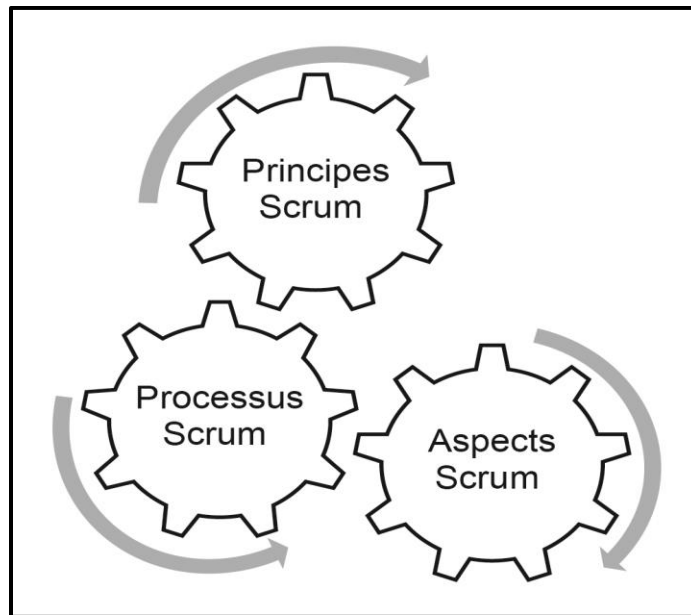


Figure 1-2: Cadre du Guide SBOK[®]

1.4.1 Comment utiliser le *Guide SBOK*® ?

Le *Guide SBOK*® peut être utilisé comme guide de référence ou comme manuel à la fois par les praticiens Scrum expérimentés et les autres praticiens de développement de produits ou de services, ainsi que par les personnes sans expérience ou connaissances préalables concernant la gestion de projet ou l'approche Scrum. Les contenus sont organisés de façon à faciliter la consultation par les trois rôles de l'équipe Scrum principale : Le Scrum Master, le Product Owner et l'équipe Scrum.

Les chapitres qui abordent les six principes Scrum (chapitre 2) et les cinq aspects Scrum (chapitres 3 à 7) incluent un guide des rôles. Ce guide des rôles offre des informations concernant la pertinence de chaque section du chapitre pour les rôles de l'équipe Scrum principale.

De façon à faciliter l'application de Scrum, le *Guide SBOK*® identifie clairement les contributions, outils et résultats obligatoires de ceux qui ne le sont pas ou sont seulement facultatifs. Les contributions, outils et résultats accompagnés d'un astérisque (*) sont obligatoires ou considérés comme étant essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont facultatifs. Pour les personnes qui découvrent l'approche Scrum, il est recommandé de se concentrer principalement sur les contributions, outils et résultats obligatoires. Les praticiens plus expérimentés pourront entièrement lire les chapitres sur les processus afin de tirer profit des contributions, outils et résultats recommandés facultatifs qui y sont suggérés.

Scrum est une approche qui n'est pas destinée à être normative, ce qui signifie qu'elle est flexible dans sa mise en application. Tous les processus Scrum fondamentaux expliqués dans le *Guide SBOK*® (chapitres 8 à 12) sont requis pour chaque projet Scrum, mais ne sont appliqués qu'en fonction des besoins spécifiques relatifs à l'organisation, au projet, au produit ou à l'équipe. Des inputs, des outils et des outputs supplémentaires s'appliqueraient lors de la mise à l'échelle de Scrum pour les grands projets (chapitre 13) (chapitre 13) et des processus supplémentaires s'appliqueraient lors de la mise à l'échelle de Scrum pour l'entreprise (chapitre 14).

1.4.2 Principes Scrum

Les principes Scrum sont les directives fondamentales à l'application de Scrum et doivent obligatoirement être utilisés pour tous les projets. Les six principes Scrum présentés dans le chapitre 2 sont :

1. le contrôle empirique des processus
2. l'auto-organisation
3. la collaboration
4. la priorisation basée sur la valeur
5. le time-boxing
6. le développement itératif

La figure 1-3 illustre les six principes Scrum.

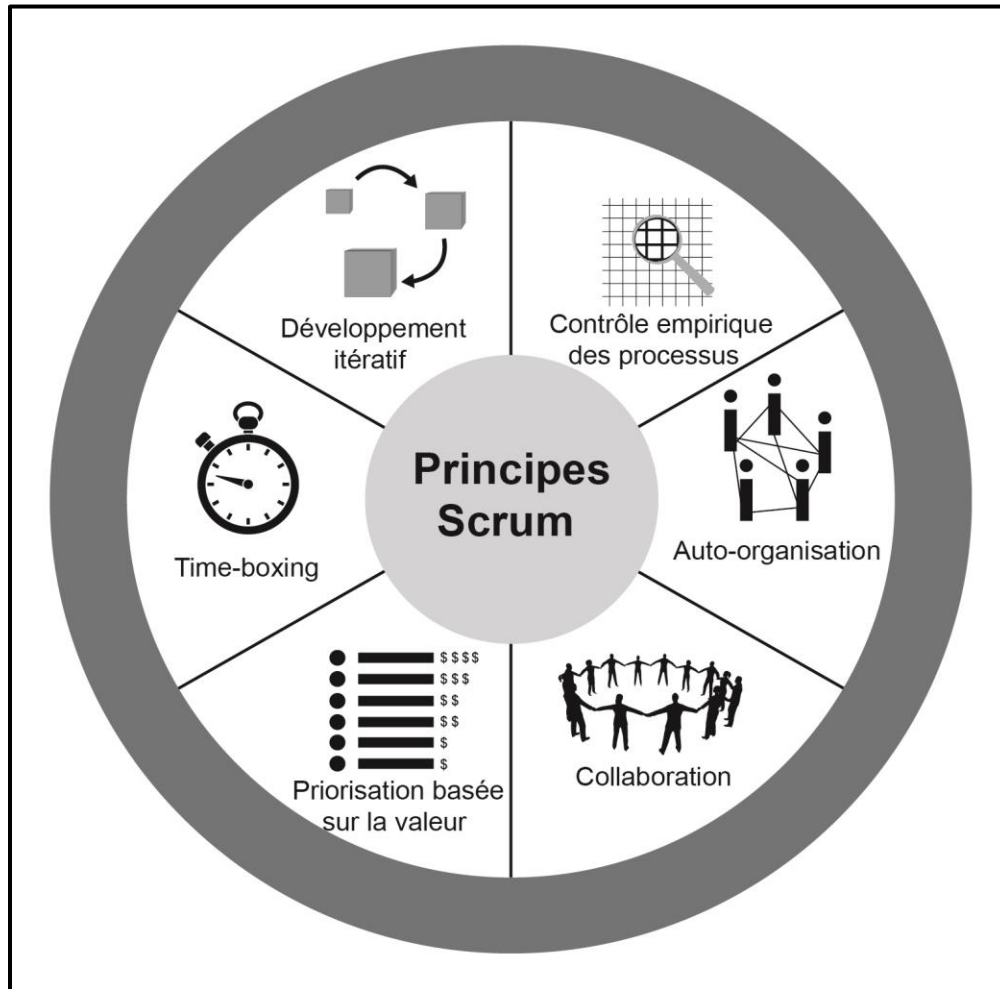


Figure 1-3: Principes Scrum

Les principes Scrum s'appliquent à tout type de projet, quelle que soit l'organisation. Ils doivent être respectés de façon à assurer la mise en œuvre efficace de l'approche Scrum. Les principes Scrum sont non négociables et doivent être appliqués comme indiqué dans le *Guide SBOK*[®]. Garder ces principes intacts et les utiliser de façon appropriée inspire la confiance dans l'approche Scrum en ce qui concerne la réalisation des objectifs du projet. Cependant, les aspects et processus Scrum sont modifiables pour permettre de répondre aux exigences spécifiques au projet ou à l'organisation.

1. **Contrôle empirique des processus** — ce principe met en avant la philosophie qui se trouve au cœur de l'approche Scrum, basée sur les trois idées principales de transparence, d'inspection et d'adaptation. Le contrôle empirique des processus facilite l'apprentissage par l'expérimentation, en particulier lorsque le problème n'est pas bien défini ou lorsqu'il n'existe pas de solutions claires.
2. **Auto-organisation** — ce principe repose sur les travailleurs actuels qui offrent une bien meilleure valeur ajoutée lorsqu'ils sont autoorganisés. Cela contribue à une meilleure acceptation de l'équipe, au partage des responsabilités et à un environnement créatif et innovant favorable à la croissance.

3. **Collaboration** — ce principe repose sur les trois dimensions principales associées au travail collaboratif : la conscience, l'expression et l'appropriation. Il recommande également de réalisation du projet comme un processus partagé de création de valeur ajoutée avec des équipes qui travaillent et interagissent ensemble, afin d'offrir la meilleure valeur ajoutée.
4. **Priorisation basée sur la valeur** — ce principe souligne l'objectif de Scrum qui vise à offrir la meilleure valeur commerciale et ce dès le début du projet et en continuant tout au long de celui-ci.
5. **Time-boxing** — ce principe décrit la manière dont le temps est considéré comme une contrainte limitée dans l'approche Scrum et doit être utilisé efficacement afin de gérer l'organisation et la réalisation d'un projet. Les éléments limités dans le temps dans Scrum incluent les sprints, les mêlées quotidiennes (dits « standups »), les réunions de planification de sprint, les réunions de révision de sprint, et les réunions de rétrospective de sprint.
6. **Développement itératif** — ce principe définit le développement en itérations et met l'accent sur comment mieux gérer les changements et créer des produits qui satisfont les exigences du client. Il délimite aussi les responsabilités du Product Owner et de l'organisation concernant le développement itératif.

1.4.3 Aspects Scrum

Les aspects Scrum doivent être abordés et gérés tout au long de chaque projet Scrum. Les cinq aspects Scrum présentés dans les chapitres 3 à 7 sont :

1.4.3.1 Organisation

Comprendre les rôles et responsabilités définis dans un projet Scrum est essentiel pour s'assurer de l'application réussie de Scrum. Les rôles Scrum sont regroupés en deux grandes catégories :

1. **Rôles principaux** — ce sont les rôles nécessaires à la réalisation du produit ou du service du projet. Les individus auxquels les rôles principaux sont attribués participent pleinement au projet et sont responsables de la réussite de chaque itération ainsi que du projet dans son ensemble.

Ces rôles incluent :

- Le **Product Owner**, c'est la personne responsable d'atteindre la valeur commerciale maximum du projet. Il est aussi chargé d'exprimer les exigences du client et de soutenir la justification commerciale du projet. Le Product Owner représente la voix du client.
- Le **Scrum Master**, c'est un facilitateur qui s'assure que l'équipe Scrum dispose d'un environnement favorable à la réussite du projet. Le Scrum Master guide, facilite et

enseigne les pratiques Scrum à toutes les personnes participant au projet. Il élimine les obstacles pour l'équipe et garantit le respect des processus Scrum.

- **L'équipe Scrum**, c'est le groupe ou l'équipe responsable de l'interprétation des exigences spécifiées par le Product Owner et de créer les incréments de produit du projet.

2. Rôles non essentiels — ce sont les rôles qui ne sont pas obligatoirement requis pour le projet Scrum, ils peuvent inclure des membres de l'équipe qui sont intéressés par le projet. Ces personnes n'ont pas de rôle défini dans l'équipe de projet, peuvent interagir avec l'équipe mais ne sont pas responsables de la réussite du projet. Les rôles non essentiels doivent être pris en compte dans n'importe quel projet Scrum.

Les rôles non essentiels incluent :

- **Les parties prenantes** de l'entreprise sont un sous-ensemble de toutes les parties prenantes d'un projet Scrum - les parties prenantes comprennent tous les individus et groupes affectés par le projet Scrum, tant à l'intérieur qu'à l'extérieur de l'organisation (par exemple, tous les rôles principaux et secondaires, les fournisseurs, les groupes internes, les experts, etc.)
- Le **Scrum Guidance Body (SGB)**, c'est un rôle facultatif qui consiste généralement en un ensemble de documents et/ou en un groupe d'experts qui sont participant à la définition des objectifs relatifs à la qualité, aux législations gouvernementales, à la sécurité et à d'autres paramètres organisationnels clés. Le SGB guide le travail effectué par le Product Owner, le Scrum Master et l'équipe Scrum.
- Les **fournisseurs**, ce rôle inclut les individus ou organisations externes qui fournissent des produits et/ou des services qui ne font pas partie des compétences de base de l'organisation du projet.

La figure 14 illustre l'organisation Scrum.

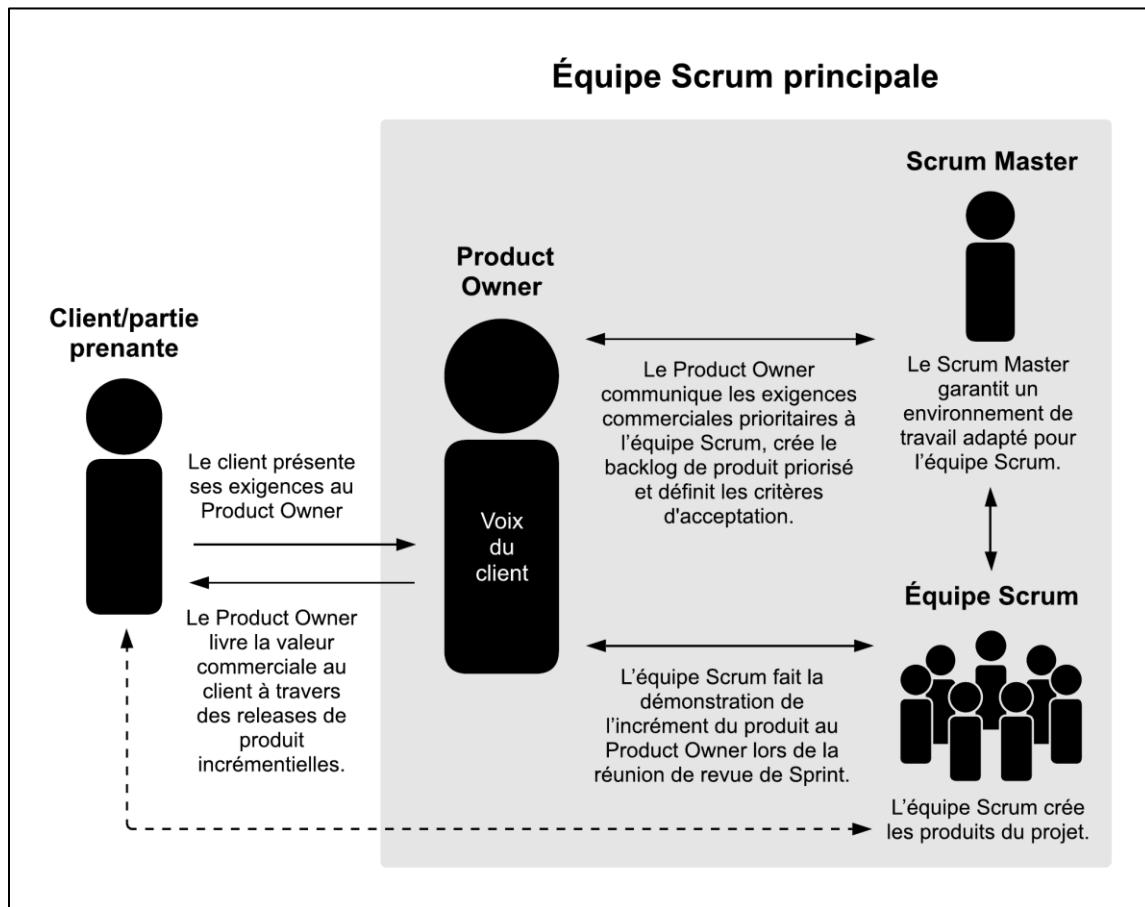


Figure 1-4: Organisation de Scrum

L'organisation de Scrum aborde également les exigences relatives à la structure de l'équipe pour les gros projets, les programmes ou les portefeuilles.

1.4.3.2 Justification commerciale

Il est important pour chaque organisation de réaliser une évaluation commerciale adéquate avant de commencer n'importe quel projet. Cela aide les décideurs clés à comprendre les besoins commerciaux associés à un changement de produit, à un nouveau produit ou service, ainsi que la justification en faveur de la validation d'un projet et de sa viabilité.

Dans Scrum, la justification commerciale est basée sur le concept de livraison de la valeur ajoutée. L'une des caractéristiques fondamentales de tout projet est l'incertitude relative à son aboutissement et à ses résultats. Il est impossible de garantir la réussite d'un projet une fois fini, peu importe sa taille ou son niveau de complexité. Au vu de cette incertitude quant à la réussite, l'approche Scrum tente d'offrir des résultats aussi tôt que possible au cours du projet.

La livraison anticipée des résultats, et donc de la valeur ajoutée, offre une opportunité de réinvestissement et atteste de la valeur du projet pour parties prenantes de l'entreprise.

L'adaptabilité de l'approche Scrum permet de modifier les objectifs et processus du projet si sa justification commerciale vient à changer. Il est important de noter que, même si le Product Owner est le principal responsable de la justification commerciale, les autres membres de l'équipe y contribuent aussi de manière non négligeable.

1.4.3.3 Qualité

Dans Scrum, la qualité est définie comme la capacité à terminer un produit répondant aux critères d'acceptation et produisant la valeur commerciale attendue par le client.

L'approche Scrum adopte une approche de progression continue afin de garantir qu'un projet répond aux exigences qualitatives. Les membres de l'équipe tirent les enseignements de leurs expériences et de la participation des parties prenantes de l'entreprise afin de maintenir le backlog de produit priorisé constamment à jour en tenant compte de la modification des exigences. Le backlog de produit priorisé n'est fini qu'à la clôture ou conclusion du projet. Tous les changements des exigences reflètent les changements de l'environnement commercial interne et/ou externe. Ils permettent ainsi à l'équipe de travailler et de s'adapter continuellement afin de répondre à ces exigences.

Puisque Scrum consiste à réaliser le travail en incréments durant les sprints, les erreurs ou défauts sont identifiés plus rapidement, grâce à la répétition des tests de qualité, plutôt que lorsque le produit ou service est presque terminé. De plus, les tâches importantes relatives à la qualité (p. ex. le développement, les tests et la documentation) sont réalisées au cours du même sprint et par la même équipe, garantissant ainsi la qualité inhérente de chaque incrément créé au cours d'un sprint. Dans les projets Scrum, de tels incréments de produits, lorsqu'ils sont potentiellement livrables, sont dit *terminés*.

Cette progression continue et les tests répétés optimisent donc la probabilité d'atteindre le niveau de qualité attendu au cours d'un projet Scrum. Les discussions constantes entre l'équipe Scrum principale et parties prenantes de l'entreprise (y compris les clients et les utilisateurs) avec de vrais incréments de produits livrés à la fin de chaque sprint permettent de constamment réduire l'écart entre les attentes du client vis-à-vis du projet et les incréments réalisés.

Le Scrum Guidance Body est également habilité à offrir des orientations en matière de qualité applicables à tous les projets au sein de l'organisation.

1.4.3.4 Changement

Chaque projet, quelle que soit la méthode ou l'approche utilisée, est soumis au changement. Les membres de l'équipe du projet doivent impérativement comprendre que les processus de développement Scrum sont conçus afin de prendre en compte les changements. En appliquant les principes Scrum pour les processus de gestion des changements, les organisations essaient d'optimiser les avantages provenant des changements et de minimiser les impacts négatifs.

Un principe fondamental de l'approche Scrum est le constat que a) au cours d'un projet, les parties prenantes de l'entreprise (p. ex. clients, utilisateurs et sponsors) changent d'avis concernant de ce qu'elles veulent et ce dont elles ont besoin (parfois appelé « modification des exigences ») et b) il est très difficile, voire impossible, pour parties prenantes de l'entreprise de définir toutes les exigences lors de la phase d'initiation d'un projet.

Les projets Scrum tirent parti du changement en utilisant des sprints courts et itératifs qui tiennent compte du feedback des clients sur chaque incrément créé durant un sprint. Cela permet aux clients d'interagir régulièrement avec les membres de l'équipe Scrum, de voir les incréments de produit lorsque ceux-ci sont prêts et de modifier les exigences plus tôt au cours du sprint, si cela est nécessaire.

Les équipes de gestion du portefeuille ou du programme peuvent également répondre aux demandes de changement applicables à leur niveau et ayant trait aux projets Scrum.

1.4.3.5 Risque

Le risque est défini comme un évènement ou une série d'évènements incertains pouvant affecter les objectifs d'un projet et contribuer à sa réussite comme à son échec. Les risques qui pourraient impacter positivement le projet sont considérés comme des opportunités, tandis que les menaces sont des risques qui pourraient affecter le projet de façon négative. Les risques doivent être gérés de façon dynamique. La gestion des risques est un processus itératif qui doit débuter durant la phase d'initiation du projet et continuer tout au long du cycle du projet. Ce processus doit respecter des étapes standardisées de façon à garantir l'identification et l'évaluation des risques ainsi que la mise en place et l'exécution d'un plan d'action adapté.

Les risques sont identifiés, évalués et les actions pour y répondre sont mises en place sur la base de deux facteurs : la probabilité que chaque risque se produise et le possible impact en cas d'occurrence. Les risques à forte valeur de probabilité et d'impact (déterminée en multipliant les deux facteurs entre eux) sont abordés en priorité et avant ceux à plus faible valeur. Généralement, une fois le risque identifié, il est important de comprendre celui-ci en termes de causes probables et d'effets potentiels si le risque venait à se concrétiser.

1.4.4 Processus Scrum

Les processus Scrum traitent des activités spécifiques et du flux d'un projet Scrum. Cependant, les processus Scrum ne sont généralement pas séquentiels mais plutôt itératifs par nature et peuvent se chevaucher les uns les autres. Au total, il existe dix-neuf processus Scrum fondamentaux qui s'appliquent à tous les projets. Ces processus sont regroupés en cinq phases et présentés dans les chapitres 8 à 12 du *Guide SBOK®*, comme illustré dans le tableau 1-1.

| Chapitre | Phase | Processus Scrum fondamentaux |
|----------|--|--|
| 8 | Phase d'initiation | <ol style="list-style-type: none"> 1. Création de la vision de projet 2. Identification du Scrum Master et Parties prenantes de l'entreprise 3. Formation de l'équipe Scrum 4. Développement des epics 5. Création du backlog de produit priorisé 6. Planification de la release |
| 9 | Phase de planification et d'estimation | <ol style="list-style-type: none"> 7. Création des user stories 8. Estimation des user stories 9. Validation des user stories 10. Identification des tâches 11. Estimation des tâches 12. Mise à jour du Sprint Backlog |
| 10 | Phase d'implémentation | <ol style="list-style-type: none"> 13. Création des incréments de produit 14. Mêlées quotidiennes 15. Raffinage du Product Backlog priorisé |
| 11 | Phase de revue et de rétrospective | <ol style="list-style-type: none"> 16. Démonstration et validation du sprint 17. Rétrospective de sprint |
| 12 | Phase de release | <ol style="list-style-type: none"> 18. Livraison des produits 19. Rétrospective de release |

Tableau 1-1: Résumé des processus Scrum fondamentaux

Ces phases décrivent chaque processus en détail, y compris les contributions, outils et résultats qui leur sont associés. Pour chaque processus, certaines contributions, outils et résultats sont obligatoires (ceux avec un astérisque [*] après leur nom), tandis que d'autres sont facultatifs. Le choix d'inclure ou non les contributions, les outils et/ou résultats facultatifs dépend du projet, de l'organisation ou du secteur d'activité. Les contributions, outils et résultats suivis d'un astérisque sont considérés comme obligatoires ou essentiels à l'application de Scrum dans n'importe quelle organisation.

Pour les projets Scrum à grande échelle qui requièrent une coordination entre plusieurs équipes, il est nécessaire de disposer d'inputs, d'outils et d'outputs supplémentaires. Ils sont définis dans le chapitre 13— Mise à l'échelle de Scrum pour les grands projets.

En outre, des processus supplémentaires sont définis lors de l'implémentation de Scrum au niveau de l'entreprise. Ces points sont abordés au chapitre 14—Mise à l'échelle de Scrum pour l'entreprise.

1.4.4.1 Phase d'initiation

1. *Création de la vision de projet* — l'étude de cas du projet est passée en revue pour créer un énoncé de vision du projet qui servira d'inspiration et permettra de recentrer le projet tout entier. Le Product Owner est également identifié durant ce processus.
2. *Identification du Scrum Master et Parties prenantes de l'entreprise* — le Scrum Master et Parties prenantes de l'entreprise sont identifiés grâce à des critères de sélection spécifiques.
3. *Formation de l'équipe Scrum* — les membres de l'équipe Scrum sont identifiés. Généralement, c'est le Product Owner qui a la responsabilité principale de la sélection des membres de l'équipe mais il le fait souvent en collaboration avec le Scrum Master.
4. *Développement des epics* — l'énoncé de vision du projet sert de base au développement des epics. Des réunions de groupes d'utilisateurs peuvent être organisées afin de discuter des epics pertinents.
5. *Création d'un backlog de produit priorisé* — les epics sont redéfinis, élaborés puis priorisés afin de créer un backlog de produit priorisé pour le projet. La définition de *terminé* est également établie à ce moment-là.
6. *Planification de la release* — l'équipe Scrum principale passe en revue les user stories dans le backlog de produit priorisé afin de développer un calendrier de planification de la release. Ce dernier consiste principalement en un calendrier de déploiement échelonné qui peut être partagé avec parties prenantes de l'entreprise du projet. La durée du sprint est également décidée lors de ce processus.

1.4.4.2 Phase de planification et d'estimation

7. *Création des user stories* — les user stories et leurs critères d'acceptation sont créés. En règle générale, les user stories sont écrites par le Product Owner et sont conçues pour s'assurer que les exigences du client sont clairement expliquées et sont compréhensibles pour toutes parties prenantes de l'entreprise. Des exercices d'écriture des user stories peuvent avoir lieu et impliquent la création des user stories par les membres de l'équipe Scrum. Les user stories sont ajoutées au backlog de produit priorisé.
8. *Estimation des user stories* — le Product Owner clarifie les user stories de façon à permettre au Scrum Master et à l'équipe Scrum d'estimer les efforts requis pour développer la fonctionnalité décrite dans chaque user story.

9. *Validation des user stories* — l'équipe Scrum s'engage à développer les user stories approuvées par le Product Owner pour un sprint. Ce processus produit des user stories validées.
10. *Identification des tâches* — les user stories validées sont divisées en tâches spécifiques et rassemblées dans une liste des tâches.
11. *Estimation des tâches* — l'équipe Scrum principale estime les efforts requis afin de réaliser chaque tâche de la liste lors d'un atelier d'estimation des tâches. Le résultat de ce processus est une liste des tâches et efforts estimés.
12. *Mise à jour du Sprint Backlog* — au cours de la réunion de planification de sprint, l'équipe Scrum principale crée le backlog de sprint contenant toutes les tâches devant être réalisées au cours d'un sprint.

1.4.4.3 Phase d'implémentation

13. *Création des incréments de produit* — l'équipe Scrum travaille sur les tâches du backlog de sprint afin de créer des incréments de produit. Un scrumboard est souvent utilisé afin de suivre les activités et le travail réalisé. Les difficultés ou problèmes rencontrés par l'équipe Scrum peuvent être ajoutés à un registre des obstacles.
14. *Mêlées quotidiennes* — une réunion très ciblée et limitée dans le temps est organisée chaque jour. Elle est appelée *mêlée quotidienne*. C'est un forum permettant aux membres de l'équipe Scrum de se tenir informés de leurs progrès et des obstacles auxquels ils font face.
15. *Raffinage du Product Backlog priorisé* — l'Raffinage du Product Backlog priorisé est un processus au cours duquel le backlog priorisé est constamment mis à jour et entretenu. Une réunion de revue du backlog de produit priorisé peut être organisée pour discuter des modifications et mises à jour du backlog. Celles-ci sont ajoutées au backlog de produit priorisé, le cas échéant.

1.4.4.4 Phase de revue et rétrospective

16. *Démonstration et validation du Sprint* — L'équipe Scrum fait la démonstration des incréments de produit au Product Owner et aux parties prenantes de l'entreprise concernées lors d'une réunion de revue de sprint. Le but de cette réunion est de s'assurer que le Product Owner approuve et accepte les incréments de produit créés durant le sprint.
17. *Rétrospective de sprint* — Le Scrum Master et l'équipe Scrum se réunissent pour discuter des enseignements tirés lors du sprint. Ces informations sont documentées sous forme d'enseignements qui peuvent être appliqués lors des futurs sprints. Cette discussion résulte souvent en des améliorations actionnables convenues ou des mises à jour des recommandations du Scrum Guidance Body.

1.4.4.5 Phase de release

18. *Livraison des produits* — Les produits acceptés sont livrés ou transférés aux parties prenantes de l'entreprise concernées. Un accord officiel sur les produits finis documente la réussite du sprint.
19. *Rétrospective de release* — Ce processus conclue le projet. Les parties prenantes de l'entreprise de l'organisation et l'équipe Scrum principale se rassemblent afin d'effectuer une Rétrospective de release et d'identifier, de documenter et d'assimiler les enseignements du projet. Généralement, ces leçons mènent à la documentation des améliorations actionnables convenues à mettre en place au cours des projets à venir.

1.4.4.6 Réunions ou cérémonies Scrum

Les réunions ou cérémonies Scrum jouent un rôle essentiel dans la mise en œuvre efficace du cadre de Scrum et sont un moyen clé par lequel les principes de Scrum sont implémentés. Les réunions Scrum importantes et les processus associés dans lesquels ces réunions se déroulent sont résumés dans le tableau 1-2.

| Réunions Scrum | Processus Scrum |
|--|---|
| Réunion de vision du projet | <ul style="list-style-type: none"> • Création du processus de vision du projet |
| Réunions de groupes d'utilisateurs | <ul style="list-style-type: none"> • Développement des Epics • Création des User Stories |
| Réunions de groupe thématique | <ul style="list-style-type: none"> • Développement des Epics • Création des User Stories |
| Sessions ou réunions de planification de release | <ul style="list-style-type: none"> • Planification de release |
| Réunions de révision de Product Backlog priorisé | <ul style="list-style-type: none"> • Estimation des User Stories • Raffinage du Product Backlog priorisé |
| Réunions de planification de sprint | <ul style="list-style-type: none"> • Estimation des User Stories • Validation des User Stories • Identification des tâches • Estimation des tâches • Mise à jour du Sprint Backlog |
| Mêlée quotidienne | <ul style="list-style-type: none"> • Gestion des mêlées quotidiennes |

| | |
|-------------------------------------|--|
| Réunion de révision de sprint | <ul style="list-style-type: none"> • Présentation et validation du sprint |
| Réunion de rétrospective de sprint | <ul style="list-style-type: none"> • Rétrospective de sprint |
| Réunion de rétrospective de release | <ul style="list-style-type: none"> • Rétrospective de release |

Tableau 1-2: Réunions et processus Scrum

1.4.5 Scrum for Large Projects

Lorsqu'il s'agit de grands projets nécessitant les efforts de plusieurs (quatre ou plus) équipes Scrum avec plusieurs Product Owners et plusieurs Scrum Masters, les processus fondamentaux définis dans les chapitres 8 à 12 restent valables, mais certaines considérations supplémentaires et mises à jour des inputs, des outils et des outputs peuvent être nécessaires. Cela peut inclure des besoins supplémentaires de coordination et de synchronisation. Les impacts sur les processus fondamentaux Scrum lors de la mise à l'échelle de Scrum à de grands projets sont décrits au chapitre 13.

La définition de ce qui constitue un grand projet dépend généralement de l'organisation et/ou de la complexité des projets entrepris. Un critère clé pour déterminer si un projet est considéré comme petit ou grand est de savoir si le projet nécessite plusieurs Scrum Masters et/ou plusieurs Product Owners. Si le projet ne nécessite qu'un Scrum Master et un Product Owner, ces personnes peuvent normalement gérer tous les efforts supplémentaires de communication et de synchronisation requis par le projet.

1.4.6 Scrum for the Enterprise

Lorsqu'on applique Scrum au niveau de l'entreprise (par exemple à un programme ou à un portfolio), il peut y avoir plusieurs centaines d'équipes Scrum, avec plusieurs milliers de personnes travaillant sur plusieurs projets au sein de programmes et/ou des portfolios dans l'ensemble de l'organisation. L'application de Scrum au niveau d'un programme ou d'un portfolio aura certains impacts sur les projets sous-jacents. En général, les projets Scrum doivent toujours être exécutés en utilisant les processus fondamentaux Scrum abordés dans les chapitres 8 à 12 pour les petits projets classiques, tout en incorporant les considérations supplémentaires décrites dans le chapitre 13 pour les grands projets (qui ont plusieurs Product Owners et/ou Scrum Masters).

Certains des défis qui se posent au niveau du programme ou du portfolio sont similaires à ceux qui se présentent lors de l'application de Scrum à un grand projet. La synchronisation entre les équipes et la collaboration globale sont généralement les plus grands défis d'un grand projet Scrum—c'est également un défi lorsqu'on applique Scrum au niveau d'un programme ou d'un portfolio. Cependant, les plus grands défis lors de l'application de Scrum au niveau d'un programme ou d'un portfolio se situent généralement du côté de l'entreprise car les priorités de l'entreprise des différents projets entrent en conflit les unes avec les autres et peuvent également entrer en conflit avec les objectifs généraux du programme ou du portfolio. Ces priorités et ces objectifs doivent être alignés.

Lors de l'implémentation de Scrum au niveau de l'entreprise, il n'y a pas seulement des inputs, des outils et des outputs supplémentaires, comme dans un projet Scrum de grande envergure, mais aussi des processus spécifiques supplémentaires qui sont nécessaires pour faire face aux efforts supplémentaires de priorisation, d'alignement et de coordination. Ces considérations supplémentaires sont abordées au chapitre 14.

1.5 L'approche Scrum comparée à la gestion de projet traditionnelle

Le tableau 1-3 résume les nombreuses différences qui existent entre l'approche Scrum et les approches traditionnelles de la gestion de projet.

| | Scrum | Gestion de projet traditionnelle |
|----------------------------------|--|---|
| Accent sur | Les personnes | Les processus |
| Documentation | Minimale, lorsqu'elle est nécessaire | Complète |
| Style de processus | Itératif | Linéaire |
| Planification préalable | Faible | Forte |
| Priorisation des exigences | Basée sur la valeur commerciale et régulièrement mise à jour | Fixée dans le plan de projet |
| Assurance Qualité | Axée sur le client | Axée sur les processus |
| Organisation | Autoorganisée | Contrôlée |
| Style de gestion | Leadership décentralisé | Centralisée |
| Changement | Mises à jour du backlog de produit | Système formel de gestion des changements |
| Leadership | Collaboratif, leader serviteur | De commandement et de contrôle |
| Mesure de la performance | Valeur commerciale | Conformité au plan |
| Retour sur capital investi (RCI) | Tôt/pendant toute la durée du projet | Fin du projet |
| Implication du client | Forte tout au long du projet | Varie suivant le cycle du projet |

Tableau 1-3: L'approche Scrum comparée à la gestion de projet traditionnelle

2. PRINCIPES

2.1 Introduction

Les principes Scrum constituent la base sur laquelle le cadre de Scrum est fondé. Les principes Scrum peuvent s'appliquer à n'importe quel type de projet ou d'organisation et doivent être respectés de façon à assurer la mise en œuvre appropriée de Scrum. Les aspects et processus Scrum sont modifiables de façon à répondre aux exigences du projet ou des organisations, mais les principes Scrum sont non négociables et doivent être appliqués tels qu'ils sont décrits dans l'approche présentée par *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*. Le fait de garder ces principes intacts et de les utiliser de façon appropriée permet de gagner la confiance de l'utilisateur de l'approche Scrum quant à la réalisation des objectifs du projet. Les principes sont considérés comme étant les directives fondamentales à la mise en œuvre de l'approche Scrum.

Tels qu'ils sont définis dans le *Guide SBOK®*, les *principes* s'appliquent aux situations suivantes :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat d'un projet. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce chapitre est divisé comme suit :

2.2 Guide des rôles — cette section donne un aperçu des sections ou paragraphes pertinents à chacun des rôles Scrum principaux de Product Owner, de Scrum Master et de l'équipe Scrum.

2.3 Contrôle empirique des processus — cette section décrit le premier principe et les trois idées principales de transparence, d'inspection et d'adaptation.

2.4 Auto-organisation — cette section met en avant le second principe de l'approche Scrum qui repose sur les travailleurs actuels qui offrent une bien meilleure valeur ajoutée lorsqu'ils sont autoorganisés. Cela contribue à une meilleure acceptation de l'équipe, au partage des responsabilités et à un environnement créatif et innovant favorable à la croissance.

2.5 Collaboration — cette section met l'accent sur le troisième principe Scrum où le développement du produit est un processus partagé de création de valeur qui nécessite que toutes parties prenantes de l'entreprise travaillent et interagissent ensemble afin d'offrir la meilleure valeur ajoutée.

Cette section explore également les dimensions fondamentales du travail collaboratif : la conscience, l'expression et l'appropriation.

2.6 Priorisation basée sur la valeur — cette section présente le quatrième principe Scrum qui souligne la volonté de l'approche Scrum d'offrir la valeur commerciale maximale dans un délai minimal.

2.7 Time-boxing — cette section explique le cinquième principe Scrum qui considère le temps comme une limite contraignante. Elle aborde aussi le sprint, la mêlée quotidienne et les autres réunions relatives au sprint, telles que la réunion de planification de sprint et la réunion de revue de sprint, qui sont toutes limitées dans le temps.

2.8 Développement itératif — cette section aborde le sixième principe Scrum qui met l'accent sur la façon dont le développement itératif aide à mieux gérer les changements et à concevoir des produits qui répondent aux exigences du client.

2.9 L'approche Scrum comparée à la gestion de projet traditionnelle — cette section souligne les différences fondamentales entre les principes Scrum et les principes traditionnels de la gestion de projet (modèle en cascade ou « waterfall » en anglais) et explique en quoi l'approche Scrum est aujourd'hui mieux adaptée à un monde en constante évolution.

2.2 Guide des rôles

Toutes les sections de ce chapitre sont importantes pour tous les rôles de l'équipe Scrum principale: Product Owner, Scrum Master et équipe Scrum. Il est essentiel que toutes parties prenantes de l'entreprise comprennent les principes Scrum afin que la mise en œuvre de l'approche Scrum soit un succès quelle que soit l'organisation.

2.3 Contrôle empirique des processus

Dans l'approche Scrum, les décisions sont prises basées sur l'observation et l'expérimentation plutôt que sur une planification détaillée d'avance. Le contrôle empirique des processus facilite l'apprentissage par l'expérimentation, en particulier lorsque le problème n'est pas bien défini ou lorsqu'il n'existe pas de solutions claires. Le contrôle empirique des processus repose sur trois principes fondamentaux : la transparence, l'inspection et l'adaptation.

2.3.1 Transparence

La transparence permet à toutes les facettes de tout projet Scrum d'être observées par tous. Cela encourage un flux d'information simple et transparent à travers l'organisation dans son ensemble et crée un environnement de travail ouvert. Dans Scrum, la transparence est illustrée à travers les points suivants :

- un énoncé de vision de projet visible par toutes parties prenantes de l'entreprise et l'équipe Scrum
- un backlog de produit priorisé ouvert avec des user stories priorisées et consultables par tous, à la fois au sein et en dehors de l'équipe Scrum
- Un Sprint Backlog qui peut être utilisé pour dresser la liste de toutes les tâches, associées aux histoires d'utilisateur validées, qui doivent être exécutées par l'équipe Scrum dans le Sprint en cours
- Un calendrier de planification de release qui peut être utilisé pour coordonner le travail de plusieurs équipes Scrum et d'autres parties prenantes de l'entreprise
- une visibilité claire des progrès de l'équipe grâce à l'utilisation d'un scrumboard, d'un burndown chart et d'autres éléments d'informations
- Réunions de planification de sprint au cours desquelles l'équipe Scrum estime l'effort nécessaire pour livrer les User Stories de première priorité et valide un ensemble de User Stories à réaliser au cours du sprint
- les mêlées quotidiennes au cours desquelles tous les membres de l'équipe doivent rendre compte de ce qu'ils ont fait le jour précédent, de ce qu'ils prévoient de faire aujourd'hui et de tous problèmes les empêchant d'effectuer leurs tâches pour le sprint en cours
- les réunions de revue de Sprint organisées au cours du processus de *démonstration et validation de sprint*, au cours desquelles l'équipe Scrum fait la démonstration au Product Owner et aux parties prenantes de l'entreprise des incréments de produit du sprint potentiellement livrables
- Réunions de rétrospective de sprint tenues après les réunions de révision de sprint le dernier jour du sprint, au cours desquelles l'équipe Scrum discute des possibilités d'amélioration pour les prochains sprints
- Une réunion ou une session de planification de release est organisée pour permettre à l'équipe Scrum d'avoir une vue d'ensemble des releases prévues et du calendrier de livraison du produit qu'elle développe

La figure 2-1 illustre le concept de transparence dans Scrum.

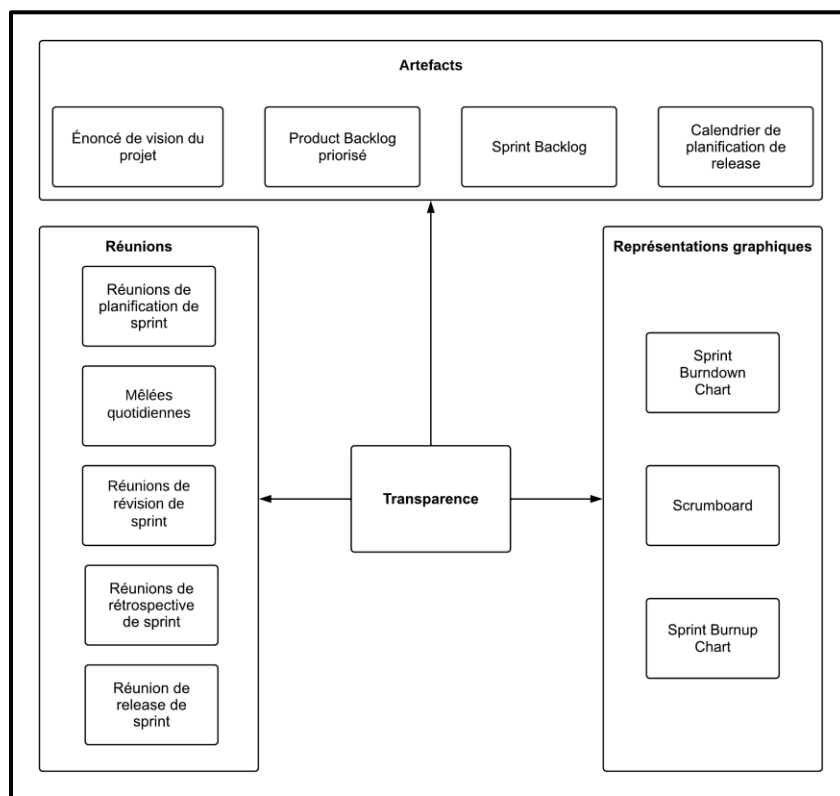


Figure 2-1: La transparence dans Scrum

2.3.2 Inspection

Dans Scrum, l'inspection est illustrée à travers les points suivants :

- l'utilisation d'un scrumboard et d'éléments d'informations communs qui montrent les progrès de l'équipe Scrum dans la réalisation des tâches pour le sprint en cours.
- le recueil du feedback des clients et autres parties prenantes de l'entreprise pendant les processus de *développement des epics*, de *création d'un backlog de produit priorisé* et de *planification des releases*.
- L'inspection et la validation des incréments de produits par le Product Owner et le client lors du processus de *démonstration et validation du sprint*.

La figure 2-2 illustre le concept d'inspection dans Scrum.

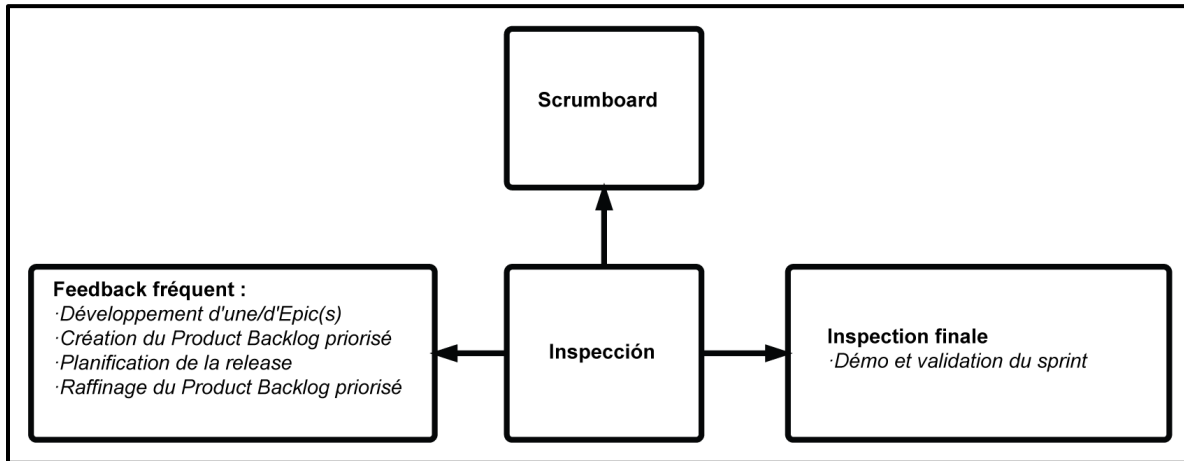


Figure 2-2: Inspection dans Scrum

2.3.3 Adaptation

On parle d'adaptation quand l'équipe Scrum principale et parties prenantes de l'entreprise apprennent grâce à la transparence ainsi qu'à l'inspection et s'adaptent en améliorant le travail qu'elles effectuent. Voici quelques exemples d'adaptation :

- lors des mêlées quotidiennes, les membres de l'équipe Scrum discutent ouvertement des obstacles qui les empêchent d'effectuer leurs tâches et cherchent à s'aider les uns les autres. au sein de l'équipe Scrum, les membres les plus expérimentés guident ceux qui ont moins d'expérience en termes de connaissance du projet ou de connaissances technologiques.
- l'identification des risques est effectuée et réitérée tout au long du projet. les risques identifiés contribuent à plusieurs processus Scrum dont la *création du backlog de produit priorisé*, Raffinage du Product Backlog priorisé et la *démonstration et validation du sprint*.
- les améliorations peuvent également résulter en demandes de changement, discutées et approuvées durant les processus de *développement des epics*, de *création du backlog de produit priorisé* et Raffinage du Product Backlog priorisé.
- le Scrum Guidance Body interagit avec les membres de l'équipe Scrum lors des processus de *création des user stories*, de *estimation des tâches*, de *création des incréments de produit* et Raffinage du Product Backlog priorisé afin de conseiller et d'apporter son expertise le cas échéant.
- durant le processus de *rétrospective de sprint*, les améliorations actionnables convenues sont décidées sur la base des résultats du processus de *démonstration et validation du sprint*.
- Dans la réunion de rétrospective de release, les participants documentent les enseignements et effectuent des vérifications en cherchant des opportunités pour améliorer les processus et aborder les inefficacités.

La figure 2-3 illustre le concept d'adaptation dans Scrum.

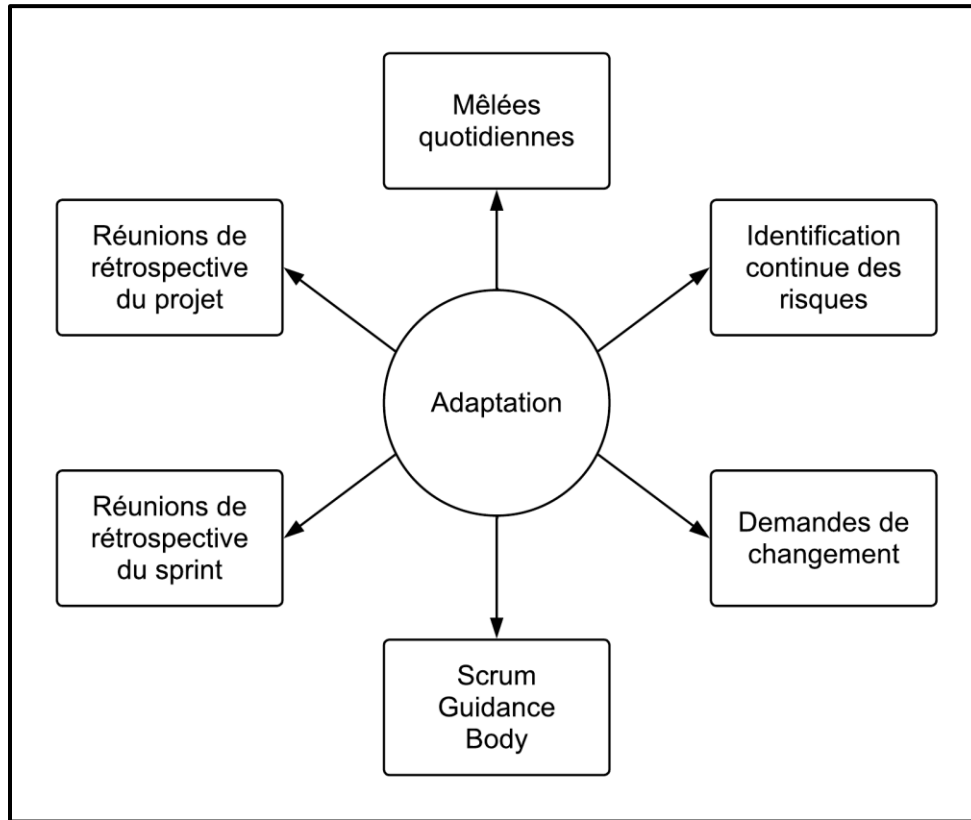


Figure 2-3: L'adaptation dans Scrum

Avec d'autres méthodes, telles que le modèle traditionnel en cascade, un effort de planification préalable considérable est nécessaire et le client ne passe généralement pas en revue les composants du produit avant la fin d'une phase ou la fin du projet dans son ensemble. Cette méthode présente souvent des risques élevés pour la réussite du projet car elle peut potentiellement avoir un impact plus important sur la livraison du projet et son acceptation par le client. L'interprétation et la compréhension que le client a du produit fini peut varier grandement de ce qui a été compris et conçu par l'équipe et cela peut ne pas être remarqué avant tard dans le développement du projet.

La figure 2-4 montre un exemple de ces défis.

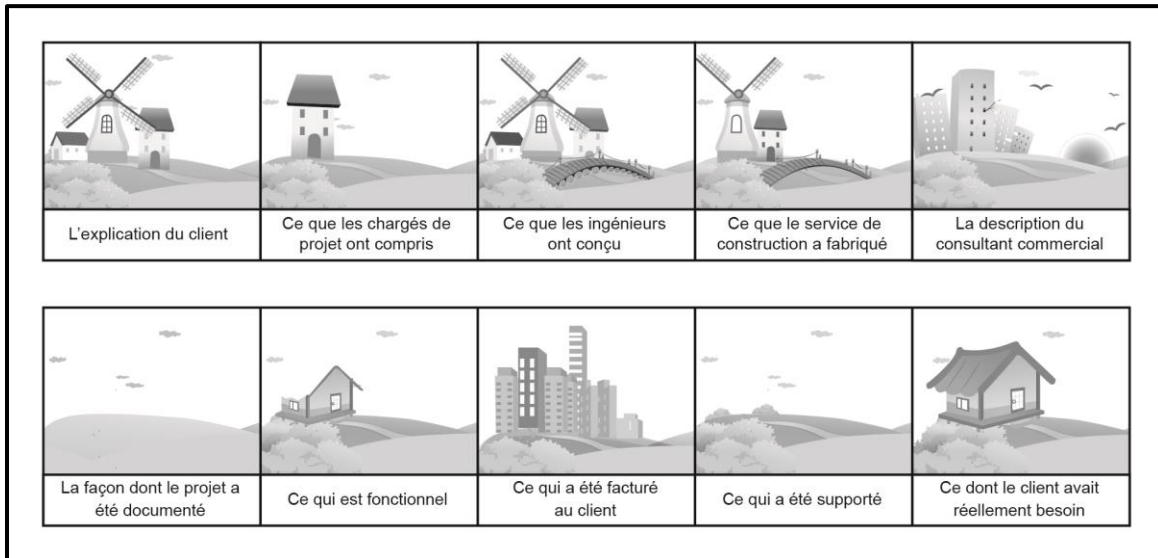


Figure 2-4: Les défis relatifs à la gestion de projet traditionnelle

2.4 Auto-organisation

L'approche Scrum repose sur le principe que les employés sont motivés et cherchent à accepter des responsabilités plus importantes. Ils offrent donc une bien meilleure valeur ajoutée lorsqu'ils sont auto-organisés.

Le style de leadership privilégié dans l'approche Scrum est celui de « leadership de soutien » (leader de soutien) et met en avant la réalisation des résultats à travers l'attention portée aux besoins de l'équipe. Voir la section 3.10.4 pour une discussion des différents styles de leadership et de gestion.

2.4.1 Avantages de l'auto-organisation

L'auto-organisation est un principe essentiel de l'approche Scrum. Elle mène à :

- l'implication de l'équipe et au partage de la responsabilité
- la motivation, qui entraîne un meilleur niveau de performance de l'équipe
- un environnement créatif et innovant favorable à la croissance
- Sélection de l'approche la plus simple et la meilleure pour satisfaire les exigences définies

L'auto-organisation ne signifie pas que les membres de l'équipe sont autorisés à agir comme ils le veulent. Cela signifie seulement que lorsque la vision de produit est définie dans le processus de *création de la vision de projet*, le Product Owner, le Scrum Master et l'équipe Scrum sont identifiés.

Aussi, l'équipe Scrum principale elle-même travaille étroitement avec parties prenantes de l'entreprise concernées afin de mieux affiner les exigences lors des processus de *développement des epics* et de *création des user stories*. L'expertise de l'équipe est utilisée pour évaluer les contributions nécessaires à l'exécution du travail prévu pour le projet. Cette évaluation et cette expertise sont appliquées à tous les aspects techniques et de gestion du projet lors du processus de *création des incréments de produit*.

Bien que la priorisation soit principalement effectuée par le Product Owner qui représente la voix du client, l'équipe Scrum autoorganisée est impliquée dans répartition et l'estimation des tâches lors des processus *d'identification et d'estimation des tâches*. Lors de ces processus, chaque membre de l'équipe est responsable de déterminer le travail qu'il ou elle devra effectuer. Pendant la réalisation d'un sprint, les membres de l'équipe pourraient avoir besoin d'aide pour réaliser leurs tâches. L'approche Scrum répond à ce problème par le biais d'interactions obligatoires et régulières entre les membres de l'équipe lors des mêlées quotidiennes. L'équipe Scrum elle-même interagit avec d'autres équipes lors des réunions Scrum of Scrums (SoS) et peut chercher à obtenir des conseils supplémentaires de la part du Scrum Guidance Body le cas échéant.

Enfin, lors du processus de *démonstration et validation du sprint*, l'équipe Scrum et le Scrum Master travaillent en étroite collaboration afin de faire la démonstration et de faire accepter l'incrément de produit créé durant le sprint. Puisque les produits sont potentiellement livrables (et que le backlog de produit est priorisé en fonction de la valeur créée par chaque user story), le Product Owner et le client peuvent clairement visualiser et énoncer la valeur créée par chaque sprint. À leur tour, les équipes Scrum ont la satisfaction de voir leur dur travail accepté par le client et les autres parties prenantes de l'entreprise .

Les objectifs principaux des équipes autoorganisées sont les suivants :

- comprendre la vision du projet et en quoi le projet offre de la valeur ajoutée à l'organisation
- estimer les user stories lors du processus *d'estimation des user stories* et s'assigner les tâches durant le processus de *Mise à jour du Sprint Backlog*
- identifier les tâches indépendamment lors du processus *d'identification des tâches*
- appliquer et tirer parti de leur expertise en tant qu'équipe polyvalente afin de travailler sur les tâches lors du processus de *création des incréments de produit*
- offrir des résultats tangibles acceptés par le client et les autres parties prenantes de l'entreprise lors du processus de *démonstration et validation du sprint*
- aborder et résoudre les problèmes individuels en équipe lors des *mêlées quotidiennes*
- clarifier tous les écarts ou doutes. Être également ouverts à apprendre de nouvelles choses
- continuellement perfectionner ses connaissances et ses compétences en interagissant régulièrement avec l'équipe
- maintenir la stabilité des membres de l'équipe pendant toute la durée du projet en ne changeant aucun membre, sauf si cela ne peut être évité

La figure 2-5 illustre les objectifs d'une équipe autoorganisée.

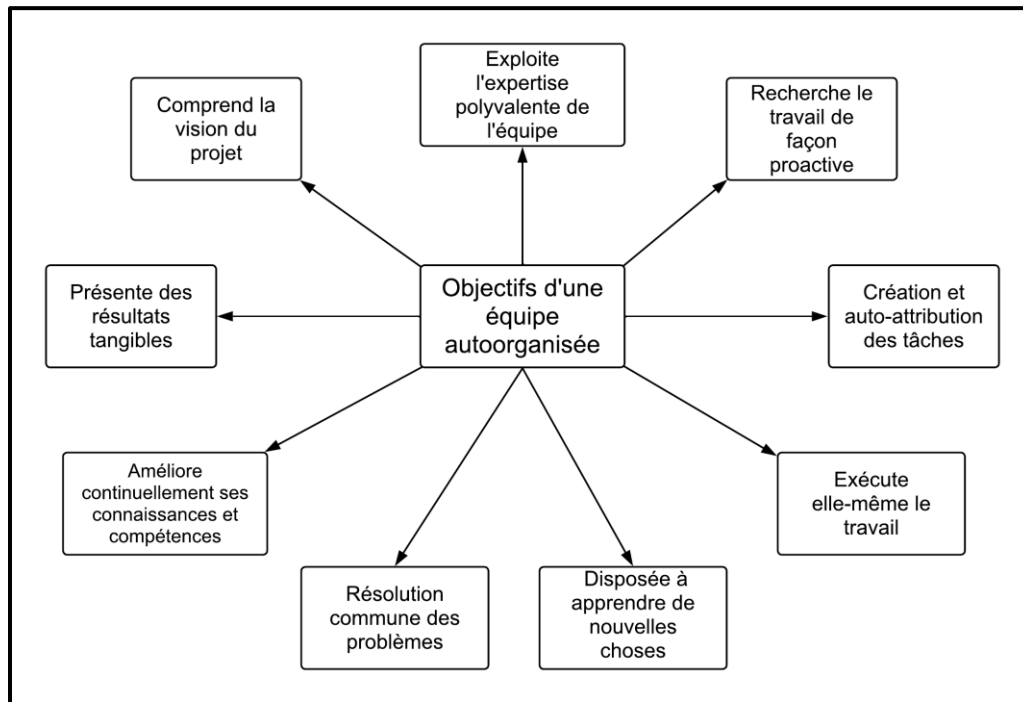


Figure 2-5: Les objectifs d'une équipe autoorganisée

2.5 Collaboration

Dans l'approche Scrum, la collaboration fait référence à l'équipe Scrum principale qui travaille avec parties prenantes de l'entreprise pour créer et valider les produits du projet répondant aux objectifs définis dans la vision du projet. Ici, il est important de faire la différence entre coopération et collaboration. On parle de coopération lorsque le résultat du travail consiste en la somme des efforts produits par les différents membres de l'équipe. On parle de collaboration lorsqu'une équipe travaille ensemble de façon à tirer avantage des contributions de chacun pour produire un meilleur résultat. Pour parvenir à une collaboration totale, il est important d'établir la confiance entre tous les membres de l'équipe et entre l'équipe et les parties prenantes de l'entreprise.

Les dimensions fondamentales du travail collaboratif sont les suivantes :

- **Conscience** — les individus qui travaillent ensemble doivent être conscient du travail des uns et des autres.
- **Organisation** — les individus qui collaborent doivent diviser le travail en unités, répartir les unités entre les différents membres, puis lorsque le travail a été effectué, le réintégrer.
- **Appropriation** — adapter la technologie à sa propre situation. La technologie peut être utilisée d'une façon complètement différente de celle imaginée par les concepteurs.

2.5.1 Avantages de la collaboration dans les projets Scrum

Le Manifeste Agile (Fowler & Highsmith, 2001) accorde plus d'importance à « la collaboration avec les clients plus que la négociation contractuelle ». Ainsi, le cadre Scrum adopte une approche où les membres de l'équipe Scrum principale (Product Owner, Scrum Master et équipe Scrum) collaborent ensemble et avec parties prenantes de l'entreprise afin de créer des produits qui offrent la plus grande valeur possible au client. La collaboration se produit tout au long du projet.

La collaboration permet d'assurer la réalisation des avantages suivants pour un projet :

1. les besoins de changements dus à des exigences mal formulées sont minimisés. Par exemple, lors des processus de *création de la vision de projet*, de *développement des epics* et de *création du backlog de produit priorisé*, le Product Owner collabore avec parties prenantes de l'entreprise afin de créer respectivement la vision de projet, les epics et le backlog de produit priorisé. Cela permet d'assurer que le travail requis afin de terminer le projet est clair pour les membres de l'équipe Scrum principale. L'équipe Scrum collabore continuellement avec le Product Owner et parties prenantes de l'entreprise grâce à un backlog de produit priorisé transparent afin de créer les produits du projet. Les *mêlées quotidiennes*, les processus d'*Raffinage du Product Backlog priorisé* et de *rétrospective de sprint* offrent à l'équipe Scrum principale l'opportunité de discuter de ce qui a été fait et de collaborer sur ce qui doit être fait. Ainsi, le nombre de demandes de changements et de modifications de la part du client est minimisé.
2. les risques sont identifiés et traités efficacement. Par exemple, les risques pour le projet sont identifiés et évalués par les membres de l'équipe Scrum principale lors des processus de *développement des epics*, de *création des incréments de produit* et lors des *mêlées quotidiennes*. Les outils de réunion Scrum tels que les *mêlées quotidiennes*, les réunions de planification de sprint, les réunions de revue du backlog de produit priorisé, etc. offrent à l'équipe des opportunités non seulement pour identifier et évaluer les risques mais aussi pour mettre en œuvre les actions permettant de répondre aux risques à forte priorité.
3. le véritable potentiel de l'équipe est atteint. Par exemple, les *mêlées quotidiennes* offrent à l'équipe Scrum l'opportunité de collaborer avec et de comprendre les forces et faiblesses de ses membres. Si un membre de l'équipe a manqué un délai, les membres de l'équipe Scrum s'organisent de façon collaborative afin d'effectuer cette tâche et d'atteindre les objectifs établis pour la réalisation du sprint.
4. les enseignements garantissent la progression continue. Par exemple, l'équipe Scrum utilise le processus de *rétrospective de sprint* afin d'identifier ce qui s'est bien ou mal passé lors du sprint précédent. Cela offre au Scrum Master l'opportunité de travailler avec l'équipe pour modifier et améliorer l'équipe pour le prochain sprint planifié. Cela permet aussi de s'assurer que la collaboration sera encore plus efficace lors du prochain sprint.

La figure 2-6 illustre le concept de collaboration dans les projets Scrum.

2

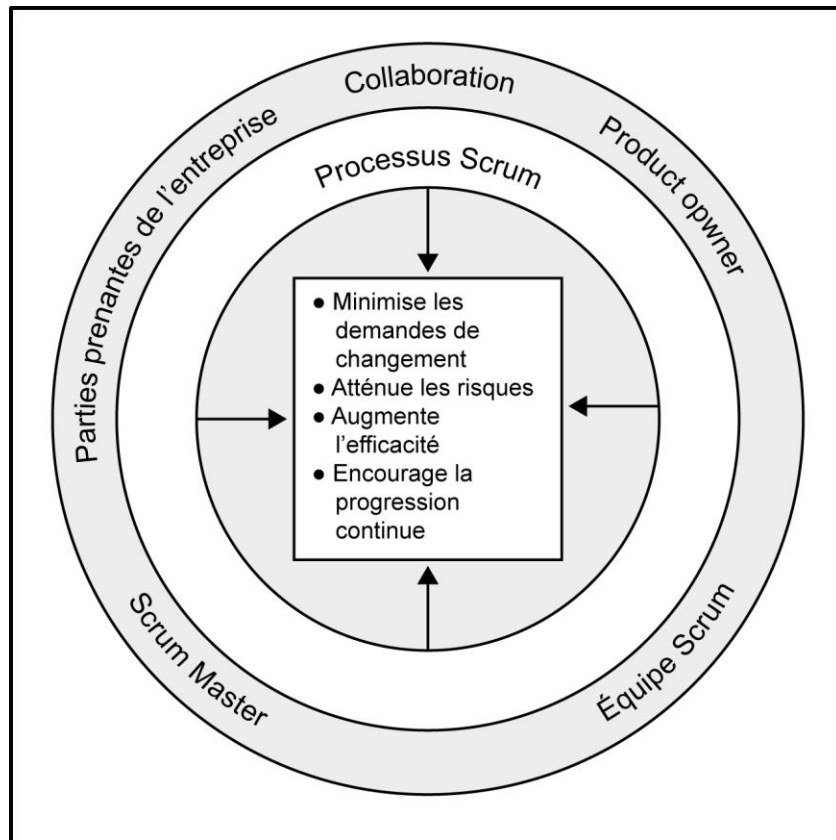


Figure 2-6: Les avantages de la collaboration dans les projets Scrum

2.5.2 Importance du regroupement dans la collaboration

Pour la plupart des pratiques Scrum un haut débit de communication est requis. Dans ce but, il est préférable que les membres de l'équipe soient regroupés. Le regroupement permet aux membres de l'équipe d'interagir à la fois de façon formelle et informelle. Ainsi, les membres de l'équipe sont toujours disponibles pour la coordination, la résolution de problèmes et l'apprentissage. Voici quelques avantages associés au regroupement :

- les questions obtiennent rapidement une réponse.
- les problèmes sont résolus immédiatement.
- il y a moins de friction lors des opérations.
- la confiance est gagnée et offerte beaucoup plus rapidement.

Les outils de collaboration qui peuvent être utilisés par les équipes regroupées ou dispersées sont les suivants :

1. **Équipes regroupées** (c-à-d. les équipes travaillant dans le même bureau) — pour l'application de Scrum il est préférable d'avoir des équipes regroupées. Lorsque l'équipe est regroupée, les moyens de communication privilégiés incluent les interactions en face-à-face, les war rooms (ou salles de décision), les scrumbords, les affichages, les tables partagées, etc.
2. **Équipes dispersées** (c-à-d. les équipes qui travaillent physiquement à différents endroits) — bien que les équipes regroupées soient privilégiées, par moments, l'équipe Scrum peut être dispersée en raison de la sous-traitance, de la délocalisation, d'emplacements géographiques différents, d'options de travail à domicile, etc. Les outils qui permettent la collaboration au sein des équipes dispersées incluent la vidéo-conférence, la messagerie instantanée, les chats, les réseaux sociaux, les écrans partagés et les outils logiciels permettant de simuler la fonctionnalité des scrumbords, les affichages, etc.

2.5.3 Collaboration dans les équipes distribuées

Bien que la préférence aille aux équipes colocalisées, l'équipe Scrum peut parfois être distribuée. Les membres de l'équipe Scrum peuvent travailler depuis plusieurs lieux physiques (comme différentes villes ou même différents pays) et/ou peuvent également travailler à domicile. Même lorsque les équipes sont situées dans un même lieu, les membres de l'équipe doivent avoir la possibilité de travailler à distance en cas de circonstances atténuantes susceptibles d'affecter la capacité de l'équipe à travailler dans un environnement situé dans un même lieu. Dans de telles situations, il peut être nécessaire de s'assurer que les membres de l'équipe Scrum sont organisés pour travailler efficacement avec l'équipe distribuée.

2.5.3.1 Outil de projet Scrum

L'utilisation d'un outil de projet Scrum collaboratif est fortement recommandé pour s'assurer que les membres de l'équipe peuvent travailler de manière productive, en particulier lorsque les membres de l'équipe ne sont pas colocalisés sur le lieu de travail. L'outil devrait idéalement permettre de :

- Définir efficacement tous les rôles de Scrum et fournir une fonctionnalité de messagerie/collaboration permettant à tous les membres de l'équipe d'interagir les uns avec les autres ;
- Créer et travailler sur des artefacts Scrum importants, tels que le Product Backlog priorisé, le Sprint Backlog, le Scrumboard etc ;
- Fournir le flux de travail pour travailler à travers les différents processus Scrum impliqués dans les phases de lancement, de planification, d'implémentation, de rétrospective et de release ;
- S'étendre au niveau de l'organisation ou de l'entreprise (si Scrum est implémenté au sein d'une grande organisation/entreprise) ;

- Planifier les réunions liées à Scrum, telles que les réunions de planification de release, les mêlées quotidiennes (dits « standups »), les réunions de planification de sprint, les réunions de révision de sprint, les réunions de rétrospective de sprint, les réunions de révision de Product Backlog, etc. Toutefois, les réunions proprement dites peuvent se dérouler dans un outil de vidéoconférence distinct;
- Permettre aux membres de l'équipe Scrum de communiquer facilement entre eux en ligne, soit en tête-à-tête, soit par le biais de groupes distribués et/ou de forums de discussion (puisque les équipes Scrum colocalisées efficaces communiquent régulièrement entre elles) ; (Notez toutefois que, contrairement aux équipes colocalisées, les membres des équipes distribuées doivent comprendre que les autres membres de l'équipe ne seront peut-être pas disponibles au même moment pour une communication instantanée.)
- Enregistrer les enseignements tirés (des rétrospectives, etc.) avec des rapports appropriés générés à chaud;
- Incorporer l'automatisation afin que tous les modèles ou les conseils du Scrum Guidance Body soient disponibles pour toutes les équipes Scrum de l'organisation (par exemple, la définition de « Ready » (Prêt) ou la définition de « Done » (Terminé));
- Permettre à l'AGS d'évaluer les comportements liés à Scrum (tels que le nombre maximum de membres de l'équipe, la durée des sprints, etc.) et
- Cloner à partir de projets similaires les Epics et les User Stories—cela permettra aux membres de l'équipe Scrum de passer moins de temps à créer de la documentation inutile ou en double et de tirer des leçons de l'expérience de tâches similaires achevées. (Cela est particulièrement bénéfique lorsque les équipes Scrum utilisent des processus d'implémentation similaires pour créer des catégories de produits identiques, par exemple, une agence de publicité qui crée des publicités imprimées pour différents clients, une entreprise de construction qui crée des dessins pour des activités de construction de routes similaires, et ainsi de suite).

Les avantages de l'utilisation d'un outil de projet Scrum pour les équipes distribuées sont les suivants:

- Faciliter le travail des membres de l'équipe Scrum à tout moment et de n'importe quel endroit
- Automatiser les rapports, les chats, le calendrier, les flux de travail, etc
- Appliquer les directives standard dans l'ensemble de l'organisation en automatisant les recommandations du SGB
- Efficacité accrue grâce à la diminution du temps consacré à la création de documents répétitifs ou inutiles lors du clonage de projets similaires (par exemple, clonage d'Epics et de User Stories)
- Travailler avec une équipe plus diversifiée (travaillant parfois dans des pays différents) permet souvent d'intégrer des perspectives et des expériences locales

- Moins de défis logistiques par rapport aux défis qui peuvent survenir en s'assurant que tous les membres de l'équipe travaillent depuis un même lieu. Cela peut également permettre d'économiser du temps et des coûts pour les dépenses liées aux déplacements, aux lieux de travail coûteux, etc.

Il est important pour les équipes distribuées d'accorder une attention particulière aux principes de Scrum afin de s'assurer qu'ils sont respectés. L'accent doit être mis sur la mise en place d'un environnement de travail transparent, collaboratif et de confiance.

2.6 Priorisation basée sur la valeur

L'approche Scrum est motivée par l'objectif visant à offrir la plus grande valeur possible dans un délai minimal. L'un des outils les plus efficaces pour offrir la plus grande valeur sur une durée la plus courte possible est la priorisation.

La priorisation se définit comme le fait de déterminer l'ordre et de séparer ce qui doit être fait maintenant de ce qui peut être fait plus tard. La priorisation est loin d'être un nouveau concept dans le domaine de la gestion de projet. Le modèle traditionnel de gestion de projet en cascade propose d'utiliser plusieurs outils différents afin d'établir la priorité des tâches. Du point de vue du chef de projet, la priorisation est essentielle car certaines tâches doivent être accomplies en premier afin d'accélérer le processus de développement et d'atteindre les objectifs du projet. Certaines des techniques traditionnelles de priorisation des tâches impliquent d'établir des échéances pour les tâches déléguées en utilisant des matrices de priorisation.

L'approche Scrum, cependant, utilise la priorisation basée sur la valeur comme un des principes fondamentaux permettant de motiver la structure et la fonctionnalité de Scrum dans son ensemble. Ce système de priorisation contribue à l'efficacité des projets grâce à son adaptabilité et au développement du produit ou service en incréments. Plus important encore, l'approche Scrum a pour objectif d'offrir au client un produit ou service de valeur dans un délai court et de façon continue lors du développement du projet.

La priorisation est effectuée par le Product Owner lorsqu'il priorise les user stories dans le backlog de produit priorisé. Le backlog de produit priorisé contient une liste de toutes les exigences requises afin de concrétiser le projet.

Une fois que le Product Owner a reçu les exigences commerciales du client, celles-ci sont écrites sous la forme d'Epics et de User Stories (un format spécifique pour la saisie des exigences). Le Product Owner travaille avec le client et d'autres parties prenantes de l'entreprise pour déterminer quelles exigences commerciales apportent une valeur commerciale maximale. Parfois, un client peut insister pour que toutes les User Stories soient catégorisées comme hautement prioritaires. Même une liste de User Stories hautement prioritaires doit se voir attribuer des priorités relatives. Le Product Owner doit comprendre ce que le client veut et valorise de façon à organiser les éléments du Product Backlog priorisé (User Stories) en fonction de leur importance relative. Cette liste s'appelle le Product Backlog priorisé et doit contenir toutes les exigences du projet. Prioriser un Backlog implique de déterminer l'importance de chaque User Story. Les exigences à forte valeur sont identifiées et remontées en haut du Product Backlog priorisé.

Les processus dans lesquels le principe de l'établissement des priorités en fonction de la valeur est mis en pratique sont la création du Product Backlog priorisé et le raffinage du Product Backlog priorisé.

Une fois que le Product Owner a reçu les exigences commerciales de la part du client et les a écrites sous la forme d'user stories réalisables, il travaille avec le client et le sponsor afin de comprendre quelles exigences offrent le plus de valeur commerciale. Le Product Owner doit comprendre ce que le client veut et valorise de façon à organiser les éléments du backlog de produit priorisé (user stories) en fonction de leur importance relative. Parfois, un client peut mandater toutes les user stories avec un haut niveau de priorité. Bien que cela puisse être vrai, même une liste d'user stories à haut niveau de priorité doit être priorisée en elle-même. Prioriser un backlog implique de déterminer l'importance de chaque user story. Les exigences à forte valeur sont identifiées et remontées en haut du backlog de produit priorisé. Le principe de priorisation basée sur la valeur est appliqué lors des processus de *création du backlog de produit priorisé* et d'*Raffinage du Product Backlog priorisé*.

Parallèlement, le Product Owner doit travailler avec l'équipe Scrum afin de comprendre les risques et incertitudes du projet étant donné que celui-ci pourrait avoir à faire face aux conséquences qui leur sont associées. Cela doit être pris en compte lors de la priorisation des user stories en fonction de leur valeur (voir le chapitre *Risques* pour plus d'informations). L'équipe Scrum alerte également le Product Owner de toute dépendance survenant lors de la réalisation. Ces dépendances doivent être prises en compte lors de la priorisation. La priorisation peut être basée sur une estimation subjective de la valeur commerciale ou de la rentabilité projetée, ou bien elle peut se baser sur des résultats et analyses du marché en utilisant des outils qui incluent notamment les entretiens avec les clients, les sondages, ainsi que les modèles financiers et techniques analytiques.

Le Product Owner doit traduire les contributions et besoins des parties prenantes de l'entreprise du projet afin de créer le backlog de produit priorisé. Ainsi, lors de la priorisation des user stories dans le backlog de produit priorisé, les trois facteurs suivants sont pris en compte (voir la figure 2-7) :

1. la valeur
2. le risque et l'incertitude
3. les dépendances

La priorisation résulte donc en des produits qui satisfont les exigences du client avec l'objectif d'offrir la meilleure valeur dans un délai le plus court possible.

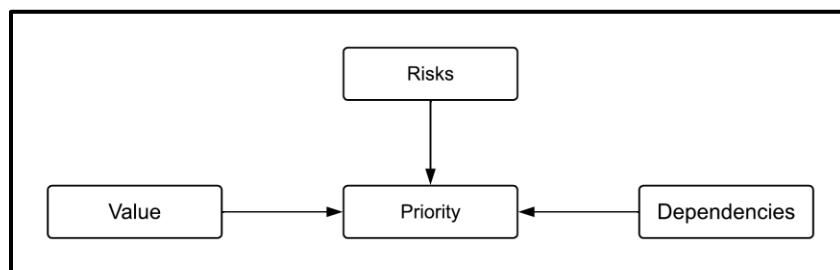


Figure 2-7: Priorisation basée sur la valeur

2.7 Time-boxing

L'approche Scrum considère le temps comme étant une des contraintes les plus importantes pour la gestion d'un projet. Afin de répondre à cette contrainte de temps, Scrum introduit le concept de « time-boxing » (« boîte de temps » en français) qui propose d'attribuer une certaine durée à chaque processus et activité lors d'un projet Scrum. Cela permet de garantir que les membres de l'équipe Scrum ne cherchent pas à effectuer trop ou trop peu de travail sur une période de temps spécifique et ne gaspillent pas leur temps et leur énergie sur un travail pour lequel ils ne disposent que de peu de précisions.

Ci-dessous figurent quelques avantages du time-boxing :

- processus de développement efficace
- moins de frais
- Haut niveau de vélocité des équipes
- Des équipes plus concentrées
- Des membres de l'équipe bien préparés

Le time-boxing peut être utilisé dans de nombreux processus Scrum, par exemple, lors des mêlées quotidiennes la durée de la réunion est limitée dans le temps. Dans certains cas, le time-boxing peut être utilisé afin d'empêcher l'amélioration excessive d'un élément (ou « gold-plating »).

Le time-boxing est une pratique essentielle à l'approche Scrum et doit être appliqué avec soin. Il doit être utilisé de façon appropriée. Un time-boxing arbitraire pourrait entraîner la démotivation de l'équipe et avoir comme conséquence de créer un environnement d'appréhension.

2.7.1 Time-boxes Scrum

- **Sprint** — un sprint est une itération limitée dans le temps, entre une et quatre semaines, durant laquelle le Scrum Master guide, facilite et protège l'équipe Scrum contre les obstacles internes et externes lors du processus de *création des incréments de produit*. Cela aide à éviter la dérive de la vision qui pourrait à son tour affecter les objectifs du sprint. Pendant ce temps, l'équipe travaille à convertir les exigences contenues dans le backlog de produit priorisé en fonctionnalités potentiellement livrables. Afin de tirer un maximum d'avantages d'un projet Scrum, il est toujours recommandé de limiter la durée d'un sprint à 4 semaines, sauf pour les projets avec des exigences stables auquel cas la durée des sprints peut s'étendre jusqu'à 6 semaines. Pour tirer le maximum de bénéfices d'un projet Scrum et offrir une flexibilité maximale pour le changement, la durée d'un sprint doit être la plus courte possible. En même temps, le sprint doit être suffisamment long pour que l'équipe soit en mesure de créer un produit livrable fonctionnel ou expédiable qui puisse être examiné et approuvé par le Product Owner.

- **Réunion de planification de Sprint** — Cette réunion se tient avant le début du sprint, lors des processus de *validation des User Stories*, *d'identification des tâches*, *d'estimation des tâches*, et de *mise à jour du Sprint Backlog*. Elle est limitée dans le temps à deux heures pour chaque semaine de la durée du sprint. Par exemple, pour un sprint d'un mois/quatre semaines, la case de temps pour une réunion de planification de sprint devrait être de huit heures. La réunion de planification de sprint répond aux objectifs suivants:
 1. Définition des objectifs — durant la première partie de la réunion, le Product Owner explique à l'équipe Scrum les user stories possédant le plus haut niveau de priorité ou d'exigence dans le backlog de produit priorisé. L'équipe Scrum, en collaboration avec le Product Owner s'engage ensuite sur les user stories qui définissent l'objectif du sprint.
 2. Identification et estimation des tâches — l'équipe Scrum décrit ensuite « comment » réaliser les éléments sélectionnés dans le backlog de produit priorisé afin d'atteindre l'objectif du sprint. Les user stories validées ainsi que les tâches et efforts estimés associés sont inclus au backlog de sprint pour être suivis.
- **Mêlée quotidienne** — la mêlée quotidienne est une réunion quotidienne courte, limitée à 15 minutes. Les membres de l'équipe se retrouvent pour rendre compte des progrès du projet et répondre aux trois questions suivantes :
 1. Qu'ai-je fait depuis la dernière réunion ?
 2. Qu'ai-je prévu de faire d'ici à la prochaine réunion ?
 3. Le cas échéant, à quels blocages et obstacles fais-je face en ce moment ?

Cette réunion est tenue par l'équipe dans le cadre du processus d'*organisation des mêlées quotidiennes*.

- **Réunion de revue de sprint** — La réunion de révision de sprint est limitée à une heure pour chaque semaine de la durée du sprint. Par exemple, pour un sprint de quatre semaines, la case de temps pour la réunion de révision de sprint devrait être de quatre heures. La réunion de revue de sprint est limitée à quatre heures pour un sprint d'un mois. Au cours de cette réunion, qui fait partie du processus de *démonstration et validation du sprint*, l'équipe Scrum présente au Product Owner les incréments de produit du sprint en cours. Le Product Owner compare le produit (ou incrément de produit) aux critères d'acceptation et accepte ou rejette les user stories réalisées.
- **Réunion de rétrospective du sprint** — La réunion de rétrospective de sprint est limitée dans le temps à une heure par semaine de la durée du sprint. Par exemple, pour un sprint de quatre semaines, la case de temps pour la réunion de rétrospective de sprint devrait être de quatre heures. Cette réunion se déroule dans le cadre du processus de rétrospective de sprint. Pendant cette réunion, l'équipe Scrum se rassemble afin de passer en revue et de réfléchir au sprint passé en termes des processus suivis, des outils employés, des stratégies de collaboration et de communication et d'autres aspects pertinents au projet.

L'équipe discute de ce qui s'est bien ou mal passé lors du précédent sprint, le but étant de tirer les enseignements et d'améliorer les sprints à venir. Certaines opportunités d'amélioration ou de meilleures pratiques émergeant lors de cette réunion peuvent également être mises à jour dans le cadre des documents du Scrum Guidance Body.

La figure 2-8 illustre les durées limitées des réunions Scrum.

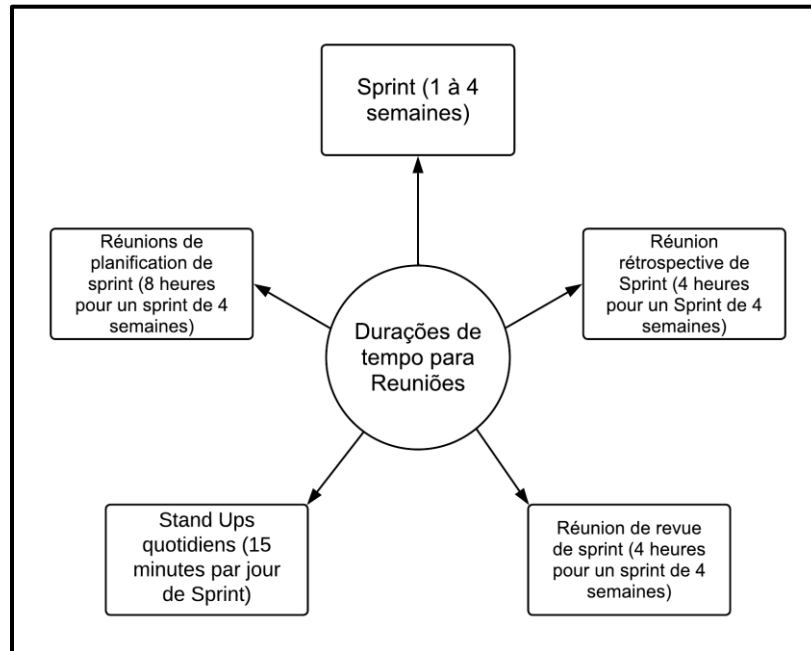


Figure 2-8: Durées des réunions Scrum

2.8 Développement itératif

L'approche Scrum est motivée par l'objectif visant à offrir la plus grande valeur possible dans un délai minimal. Afin de réaliser cela concrètement, l'approche Scrum se base sur le développement itératif des produits. Pour les projets les plus complexes, le client peut ne pas être en capacité de définir des exigences très concrètes, ou ne pas savoir à quoi le produit fini pourrait ressembler. Le modèle itératif est plus flexible et permet de garantir l'inclusion dans le projet de tout changement demandé par le client. Il peut être continuellement nécessaire d'écrire des user stories pour toute la durée du projet. Lors des phases d'écriture initiales, les user stories sont des fonctionnalités de haut niveau. Ces user stories sont aussi appelées epics. Les epics sont généralement trop larges pour être complétés par l'équipe en un seul sprint. Ils sont donc divisés en user stories plus petites. Chacun des aspects complexe du projet est divisé et élaboré progressivement lors du processus d'*Raffinage du Product Backlog priorisé*. Les processus de *création des user stories* et d'*estimation* et de *validation des user stories* permettent d'ajouter de nouvelles exigences au backlog de produit priorisé. Le rôle du Product Owner est de garantir un RCI accru en se concentrant sur la valeur et sur la production continue de celle-ci lors de chaque sprint.

Le Product Owner doit avoir une excellente compréhension de la justification commerciale du projet et de la valeur ajoutée que le projet est censé apporter lorsqu'il rédige le backlog de produit priorisé et décide des incréments et de la valeur ajoutée livrée lors de chaque sprint. Les processus d'*identification des tâches*, d'*estimation des tâches* et de *Mise à jour du Sprint Backlog* permettent ensuite de produire le backlog de sprint que l'équipe utilisera pour développer les incréments de produit. Pour chaque sprint, le processus de *création des incréments de produit* est utilisé afin de développer les résultats du sprint. Le Scrum Master doit s'assurer que les processus Scrum sont respectés et encourager l'équipe à travailler de la façon la plus productive possible. L'équipe Scrum s'autoorganise dans le but de créer les incréments de produit à partir des user stories contenues dans le sprint backlog. Pour les gros projets, plusieurs équipes polyvalentes travaillent en parallèle sur les sprints, fournissant ainsi des solutions potentiellement livrables à la fin de chaque sprint. Une fois le sprint terminé, le Product Owner accepte ou rejette les incréments de produit en fonction des critères d'acceptation lors du processus de *démonstration et validation du sprint*.

La figure 2-9 illustre la façon dont un projet Scrum est réalisé de façon itérative en offrant de la valeur ajoutée tout au long du cycle du projet.

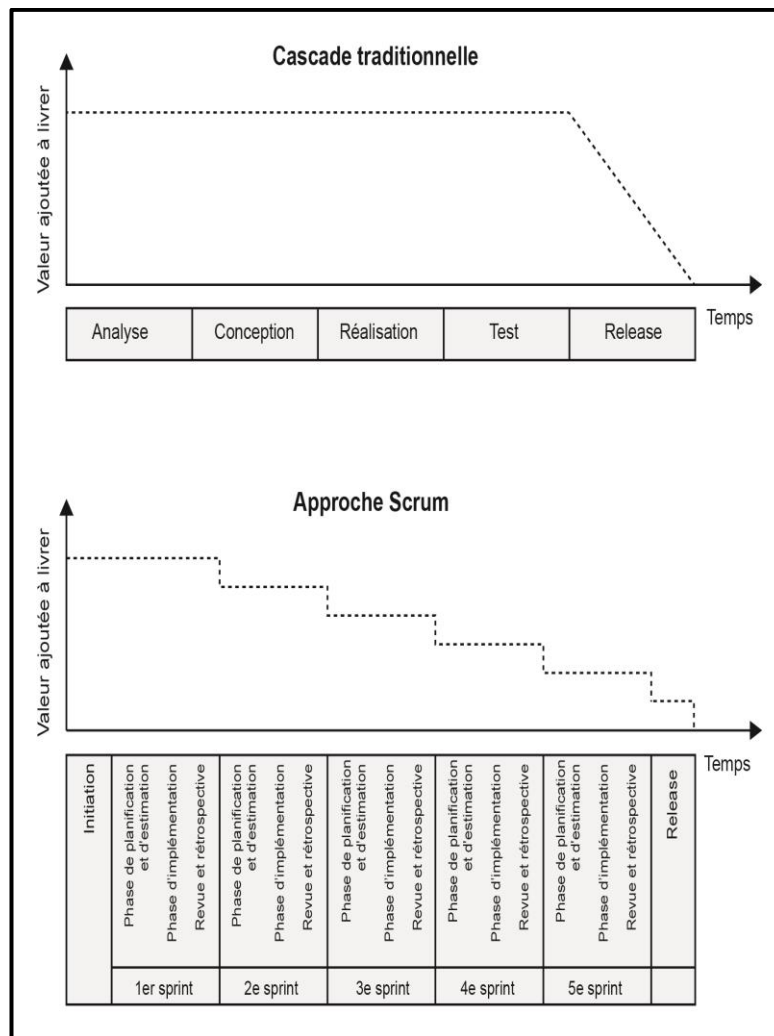


Figure 2-9: L'approche Scrum comparée à l'approche en cascade traditionnelle

L'avantage du développement itératif est qu'il permet de corriger le projet au fur et à mesure que les individus impliqués comprennent de mieux en mieux ce qui doit être livré avec le projet et assimilent ces enseignements de façon itérative. Le temps et les efforts requis pour atteindre la fin du projet sont ainsi considérablement réduits et l'équipe produit des incréments de produits qui sont mieux adaptés à l'environnement commercial final.

2.9 L'approche Scrum comparée à la gestion de projet traditionnelle

Dans la gestion de projet traditionnelle, l'accent est mis sur la planification préalable du projet et en particulier sur la définition de sa portée, de ses coûts et de son calendrier ainsi que sur la gestion de ces paramètres. La gestion de projet traditionnelle peut parfois conduire à des situations où le plan a réussi mais où le client n'est pas satisfait.

Le cadre Scrum est basé sur l'idée que les connaissances des travailleurs actuels offrent bien plus qu'une expertise technique et que d'essayer de tout planifier dans un monde en évolution permanente n'est pas efficace. Scrum encourage donc la prise de décision itérative basée sur les données. Dans l'approche Scrum, l'objectif principal est de livrer des produits qui satisfont les exigences du client sous la forme d'incrément livrables itératifs.

Afin de fournir la meilleure valeur dans un délai le plus court possible, Scrum promeut la priorisation et le time-boxing plutôt que la définition de la portée, des coûts et du calendrier du projet. L'auto-organisation est une caractéristique importante de l'approche Scrum qui permet aux individus réalisant de travail d'estimer et de s'approprier la responsabilité des tâches.

3. ORGANISATION

3.1 Introduction

Ce chapitre discute des différentes facettes relatives à l'organisation d'un projet Scrum, des rôles essentiels et non essentiels ainsi que de la formation d'équipes Scrum hautement performantes.

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, l'organisation s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat d'un projet. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce chapitre est divisé comme suit :

3.2 Guide des rôles — cette section identifie les sections ou paragraphes pertinents pour le Product Owner, le Scrum Master et l'équipe Scrum.

3.3 Rôles des projets Scrum — cette section aborde tous les rôles principaux essentiels et non essentiels associés à un projet Scrum.

3.4 Product Owner — cette section définit les principales responsabilités du Product Owner concernant un projet, programme ou portefeuille Scrum.

3.5 Scrum Master — cette section se focalise sur les principales responsabilités du Scrum Master dans le cadre d'un projet, programme ou portefeuille Scrum.

3.6 Équipe Scrum — cette section précise les principales responsabilités de l'équipe Scrum dans le cadre d'un projet Scrum.

3.7 Projets, programmes et portefeuilles Scrum — cette section se focalise sur la façon dont l'approche Scrum peut s'adapter afin d'être utilisée dans différents contextes de programmes ou de portefeuilles. Elle met également l'accent sur les responsabilités spécifiques des membres de l'équipe Scrum en matière de communication, d'intégration et de travail avec les équipes corporatives et de gestion des programmes.

3.8 Responsabilités — cette section décrit les responsabilités pertinentes au thème de l'organisation, pour toutes les personnes travaillant sur un projet, en fonction de leur rôle.

3.9 L'approche Scrum comparée à la gestion de projet traditionnelle — cette section explique les différences et avantages clés du modèle Scrum comparé au modèle de gestion de projet traditionnel en cascade.

3.10 Théories RH populaires et leur pertinence par rapport à Scrum — cette section liste certaines des théories RH les plus populaires et les plus utiles à tous les membres de l'équipe Scrum principale.

3.2 Guide des rôles

1. Product Owner — il est impératif que les Product Owners lisent ce chapitre dans son intégralité.
2. Scrum Master — le Scrum Master doit également se familiariser avec ce chapitre entier en portant une attention particulière aux sections 3.3, 3.5, 3.6, 3.8 et 3.10.4.
3. Équipe Scrum — l'équipe Scrum devrait surtout se concentrer sur les sections 3.3, 3.6 et 3.8.

3.3 Rôles des projets Scrum

Comprendre les rôles et responsabilités est essentiel pour s'assurer de l'application réussie de Scrum.

Les rôles Scrum sont regroupés en deux grandes catégories :

1. **Rôles principaux** — les rôles principaux sont les rôles qui sont obligatoirement requis pour la production du produit du projet, qui sont investis dans le projet et sont fondamentalement responsables de la réussite de chaque sprint et du projet dans son ensemble.
2. **Rôles non essentiels** — les rôles non essentiels sont les rôles qui ne sont pas obligatoirement requis pour le projet Scrum, ils peuvent inclure des membres de l'équipe qui sont intéressés par le projet, n'ont pas de rôle défini dans l'équipe de projet, peuvent interagir avec l'équipe mais ne sont pas responsables de la réussite du projet. Les rôles non essentiels doivent également être pris en compte dans n'importe quel projet Scrum.

3.3.1 Rôles principaux

Dans l'approche Scrum, il existe trois rôles principaux auxquels revient la responsabilité finale d'atteindre les objectifs du projet. Les rôles principaux sont le Product Owner, le Scrum Master et l'équipe Scrum. Ensemble ils forment l'équipe Scrum principale. Il est important de noter que parmi ces trois rôles, aucun n'a l'autorité sur l'autre.

1. Product Owner

Le Product Owner est la personne responsable d'optimiser la valeur commerciale du projet. Il est chargé d'exprimer les exigences du client et de soutenir la justification commerciale du projet. Le Product Owner représente la *voix du client*.

De la même façon qu'un projet a son Product Owner un programme peut avoir un Product Owner de programme et un portefeuille un Product Owner de portefeuille.

2. Scrum Master

Le Scrum Master est un facilitateur qui s'assure que l'équipe Scrum dispose d'un environnement favorable au développement réussi du produit. Le Scrum Master guide, facilite et enseigne les pratiques Scrum à toutes les personnes impliquées dans le projet. Il élimine les obstacles pour l'équipe et garanti le respect des processus Scrum.

Il est important de noter que le rôle du Scrum Master diffère largement de celui du Chef de projet dans un modèle de gestion de projet en cascade traditionnel et dans lequel le chef de projet travaille comme un gestionnaire ou responsable pour le projet. Le Scrum Master joue uniquement le rôle facilitateur et il est au même niveau hiérarchique que tous les autres membres de l'équipe Scrum. N'importe quel membre de l'équipe Scrum qui apprend à faciliter les projets Scrum peut devenir Scrum Master pour un projet ou pour un sprint.

De la même façon qu'un projet a son Scrum Master, un programme peut avoir un Scrum Master de programme et un portefeuille un Scrum Master de portefeuille.

3. Équipe Scrum

L'équipe Scrum est le groupe ou l'équipe responsable de l'interprétation des exigences commerciales spécifiées par le Product Owner, de l'estimation des user stories et de la création finale des produits du projet.

La figure 3-1 offre un aperçu des rôles de l'équipe Scrum principale.

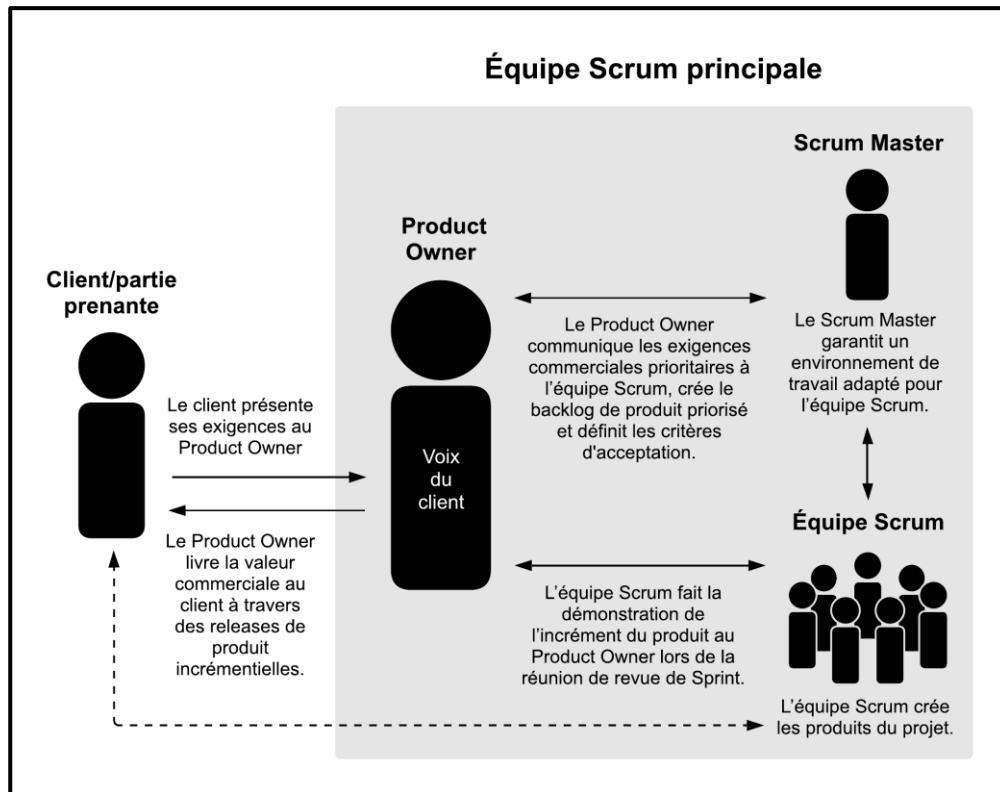


Figure 3-1: Aperçu des rôles Scrum

3.3.2 Rôles non essentiels

Les rôles non essentiels sont les rôles qui ne sont pas obligatoirement requis pour le projet Scrum, il ne sont pas toujours continuellement ou directement impliqués dans le processus Scrum. Cependant, il est important de connaître les rôles non essentiels car ceux-ci peuvent prendre une part importante dans certains projets Scrum.

Les rôles non essentiels peuvent notamment inclure les rôles suivant:

1. Parties prenantes de l'entreprise

Parties prenantes est un terme collectif qui inclut les clients, utilisateurs et les sponsors qui interagissent régulièrement avec le Product Owner, le Scrum Master et l'équipe Scrum afin d'apporter leur contribution et de faciliter la création du produit, du service ou d'un autre résultat du projet. Parties prenantes de l'entreprise influencent le projet tout au long de son développement. Parties prenantes de l'entreprise peuvent également jouer un rôle durant les processus de *développement d'epics*, de *création du backlog de produit priorisé*, de *planification de la release*, de *rétrospective de sprint* et d'autres processus Scrum importants.

La responsabilité de veiller à l'engagement des parties prenantes de l'entreprise incombe au Product Owner. Les actions suivantes sont recommandées pour maintenir l'engagement et le soutien des parties prenantes:

- Assurer une collaboration efficace et la participation des parties prenantes de l'entreprise au projet
- Évaluer en permanence l'impact sur l'entreprise
- Maintenir une communication régulière avec les parties prenantes de l'entreprise
- Gérer les attentes des parties prenantes de l'entreprise

- **Client**

Le client est la personne ou l'organisation qui acquiert le produit, le service ou tout autre résultat du projet. Quelle que soit l'organisation, suivant le projet, il peut y avoir à la fois des clients internes (c-à-d. au sein de l'organisation même) ou des clients externes (c-à-d. en dehors de l'organisation).

- **Utilisateurs**

Les utilisateurs sont les individus ou l'organisation qui utilisent directement le produit, service ou autre résultat du projet. Comme pour les clients, pour chaque organisation, il peut y avoir à la fois des utilisateurs internes ou externes. Dans certains secteurs, les clients et les utilisateurs sont les mêmes.

- **Sponsor**

Le sponsor est l'individu ou l'organisation qui offre les ressources et soutient le projet. C'est également la partie prenante à qui tout le monde doit rendre compte à la fin du projet. Dans certains cas, la même personne ou organisation peut jouer plusieurs rôles de partie prenante. Par exemple, le sponsor et le client peuvent être la même personne ou organisation.

Le sponsor doit comprendre le contexte financier lié à un produit ou un service et est généralement plus préoccupé par les résultats finaux que par les tâches individuelles. Il est important que le sponsor (ou les sponsors) qui financent le projet aient une idée claire des considérations suivantes :

- Avantages de l'application des pratiques Scrum pour le projet
- Échéances et coûts estimés du projet Scrum
- Les risques globaux liés au projet Scrum et les mesures pour les atténuer ou les éviter
- Dates de release et produits livrables finaux prévus

2. Services de soutien

Les services de soutien sont des groupes internes ou externes qui soutiennent ou sont touchés par le projet Scrum, par exemple, formation, logistique, marketing, finances, infrastructure, etc.

3. Fournisseurs

Les fournisseurs incluent les individus ou organisations externes qui fournissent des produits ou des services qui ne font pas partie des compétences de base de l'organisation du projet.

4. Scrum Guidance Body

Le Scrum Guidance Body (SGB) est un rôle facultatif, mais fortement recommandé pour formaliser les pratiques organisationnelles associées à Scrum. Il se compose généralement d'un groupe de documents et/ou d'un groupe d'experts qui sont normalement impliqués dans la définition des objectifs relatifs à la qualité, aux législations gouvernementales, à la sécurité et à d'autres paramètres organisationnels clés. Ces objectifs guident le travail effectué par le Product Owner, le Scrum Master et l'équipe Scrum. Le Scrum Guidance Body aide également à capter les meilleures pratiques pouvant s'appliquer à tous les projets Scrum de l'organisation.

Le Scrum Guidance Body ne prend aucune décision relative au projet. Au contraire, il agit comme une structure de soutien ou de conseil à tous les niveaux hiérarchiques de l'organisation du projet — portefeuille, programme et projet. S'il y a lieu, les équipes Scrum peuvent demander des conseils au Scrum Guidance Body.

3.4 Product Owner

Le Product Owner représente les intérêts de la communauté de parties prenantes de l'entreprise auprès de l'équipe Scrum. Le Product Owner est responsable d'assurer la communication claire des fonctionnalités requises du produit ou du service auprès de l'équipe Scrum, de définir les critères d'acceptation et d'assurer que ces critères sont respectés. En d'autres mots, le Product Owner est responsable de s'assurer que l'équipe Scrum offre de la valeur ajoutée. Le Product Owner doit constamment maintenir une double vision. Il doit comprendre et supporter les exigences et les intérêts de toutes parties prenantes de l'entreprise, tout en comprenant les besoins et le fonctionnement de l'équipe Scrum. Étant donné que le Product Owner doit comprendre les exigences et les priorités des parties prenantes de l'entreprise, y compris des clients et des utilisateurs, on dit souvent de ce rôle qu'il représente la *voix du client*.

Le tableau 3-1 résume les responsabilités du Product Owner lors des différents processus Scrum.

| Processus | Responsabilités du Product Owner |
|---|---|
| 8.1 Création de la vision de projet | <ul style="list-style-type: none"> • Définit la vision du projet • Aide à créer la charte du projet et le budget du projet |
| 8.2 Identification du Scrum Master et parties prenantes de l'entreprise | <ul style="list-style-type: none"> • Aide à l'identification du Scrum Master pour le projet • Identifie parties prenantes de l'entreprise |
| 8.3 Formation de l'équipe Scrum | <ul style="list-style-type: none"> • Aide à l'identification des membre de l'équipe Scrum • Aide au développement d'une stratégie de collaboration • Aide au développement du programme de consolidation d'équipe avec le Scrum Master |
| 8.4 Développement d'epics | <ul style="list-style-type: none"> • Crée des epics et personnages |
| 8.5 Création du backlog de produit priorisé | <ul style="list-style-type: none"> • Prioriser les éléments du Product Backlog priorisé • Définit les critères « Done » (Terminé) et répond à la définition de « Ready » |
| 8.6 Planification de la release | <ul style="list-style-type: none"> • Crée le calendrier de planification de la release • Aide à déterminer la durée du sprint |
| 9.1 Création des user stories | <ul style="list-style-type: none"> • Responsable de la création des User Stories • Définit les critères d'acceptation pour toutes les user stories |
| 9.2 Estimation des user stories | <ul style="list-style-type: none"> • Clarifie les user stories |
| 9.3 Validation des user stories | <ul style="list-style-type: none"> • Travaille avec l'équipe Scrum pour valider les user stories |
| 9.4 Identification des tâches | <ul style="list-style-type: none"> • Explique les user stories à l'équipe Scrum lors de la création de la liste des tâches |
| 9.5 Estimation des tâches | <ul style="list-style-type: none"> • Guide l'équipe Scrum lors de l'estimation des efforts requis pour les tâches et offre des clarifications |
| 9.6 Mise à jour du Sprint Backlog | <ul style="list-style-type: none"> • Clarifier les exigences pour l'équipe Scrum pendant que l'équipe crée le Sprint Backlog |
| 10.1 Création des incréments de produit | <ul style="list-style-type: none"> • Clarifie les exigences commerciales auprès de l'équipe Scrum |
| 10.3 Raffinage du Product Backlog priorisé | <ul style="list-style-type: none"> • Refines Prioritized Product Backlog |
| 11.1 Démonstration et validation du sprint | <ul style="list-style-type: none"> • Accepter/rejeter les produits livrables • Fournit les informations nécessaires au Scrum Master et aux équipes Scrum • Mise à jour du plan de release et du Backlog priorisé |
| 12.1 Livraison des produits | <ul style="list-style-type: none"> • Participe au déploiement des releases de produit et le coordonne avec le client |
| 12.2 Rétrospective de release | <ul style="list-style-type: none"> • Peut participer à la réunion de rétrospective de release |

Tableau 3-1: Responsabilités du Product Owner dans les processus Scrum

Les autres responsabilités du Product Owner incluent :

- de déterminer les exigences globales initiales du projet et de lancer les activités du projet. Cela peut inclure de communiquer avec le Product Owner du programme et/ou du portefeuille afin d'assurer que le projet respecte les orientations fournies par la haute direction.
- de représenter les utilisateurs du produit ou du service en témoignant d'une connaissance approfondie de la communauté d'utilisateurs.
- de sécuriser les ressources financières initiales et continues pour le projet.
- de se concentrer sur la création de la valeur ajoutée et du retour sur capital investi (RCI) global.
- d'évaluer la viabilité et d'assurer la livraison du produit ou service.

Le Product Owner ne représente pas toujours un client ou une entreprise externe. Par exemple, dans un projet informatique, les exigences telles que l'amélioration des performances, l'évolutivité, la testabilité, la fiabilité, la sécurité des informations et la conformité peuvent être détenues par les groupes technologiques au sein de l'organisation. Dans ce cas, les Product Owners peuvent également jouer le rôle d'architectes techniques, de responsables techniques, etc.

3.4.1 Voix du client (VOC)

En tant que représentant du client et des autres parties prenantes de l'entreprise, on dit du Product Owner qu'il est la voix du client puisqu'il assure que les besoins explicites et implicites du client sont traduits en user stories dans le backlog de produit priorisé et ensuite utilisés pour la création des incréments de produit pour le client.

3.5 Scrum Master

Le Scrum Master est le « leader de soutien » de l'équipe Scrum qui modère et anime les interactions de l'équipe en tant que coach et motivateur. Le Scrum Master est responsable de s'assurer que l'équipe dispose d'un environnement de travail productif en la protégeant contre les facteurs extérieurs, d'éliminer les obstacles et de faire appliquer les principes, aspects et processus Scrum.

Le tableau 3-2 résume les responsabilités du Scrum Master lors des différents processus Scrum.

| Processus | Responsabilités du Scrum Master |
|---|---|
| 8.2 Identification du Scrum Master et des parties prenantes de l'entreprise | <ul style="list-style-type: none"> Participe à l'identification des parties prenantes de l'entreprise du projet |
| 8.3 Formation de l'équipe Scrum | <ul style="list-style-type: none"> Facilite la sélection de l'équipe Scrum Facilite la création de la stratégie de collaboration et du programme de consolidation de l'équipe S'assure que les ressources de rechange sont disponibles pour le bon déroulement du projet |
| 8.4 Développement d'epics | <ul style="list-style-type: none"> Facilite la création des epics et personnages |
| 8.5 Création du backlog de produit priorisé | <ul style="list-style-type: none"> Aide le Product Owner dans la création du Product Backlog priorisé ainsi que dans la définition des critères « Done » (Terminé) et dans le respect de la définition de « Ready » (Prêt) |
| 8.6 Planification de la release | <ul style="list-style-type: none"> Coordonne la création du calendrier de planification de la release Aide le Product Owner et l'équipe Scrum à déterminer la durée du sprint |
| 9.1 Création des user stories | <ul style="list-style-type: none"> Facilite la création des User Stories et de leurs critères d'acceptation |
| 9.2 Estimation des user stories | <ul style="list-style-type: none"> Facilite les réunions de l'équipe Scrum pour l'estimation des user stories |
| 9.3 Validation des user stories | <ul style="list-style-type: none"> Facilite les réunions de l'équipe Scrum pour la validation des user stories |
| 9.4 Identification des tâches | <ul style="list-style-type: none"> Facilite la création de la liste des tâches par l'équipe Scrum pour le prochain sprint |
| 9.5 Estimation des tâches | <ul style="list-style-type: none"> Assiste l'équipe Scrum lors de l'estimation des efforts requis pour réaliser les tâches décidées pour le sprint |
| 9.6 Mise à jour du Sprint Backlog | <ul style="list-style-type: none"> Aide l'équipe Scrum à développer le Sprint Backlog et le Sprint Burndown Chart |
| 10.1 Création des incréments de produit | <ul style="list-style-type: none"> Supporte l'équipe Scrum lors de la création des incréments de produits décidés pour le sprint Aide à maintenir le scrumboard et le registre des obstacles à jour |
| 10.2 Mêlées quotidiennes | <ul style="list-style-type: none"> S'assure que le scrumboard et le registre des obstacles est maintenu à jour |
| 10.3 Raffinage du Product Backlog priorisé | <ul style="list-style-type: none"> Facilite les réunions de revue du backlog de produit priorisé |
| 11.1 Démonstration et validation du sprint | <ul style="list-style-type: none"> Facilite la présentation des incréments de produit réalisés par l'équipe Scrum pour approbation par le Product Owner |
| 11.2 Rétrospective du sprint | <ul style="list-style-type: none"> S'assure que l'environnement de projet idéal est disponible pour l'équipe Scrum pour les sprint à venir |
| 12.2 Rétrospective de release | <ul style="list-style-type: none"> Représente l'équipe Scrum principale pour tirer les enseignements du projet en cours, le cas échéant |

Tableau 3-2: Responsabilités du Scrum Master dans les processus Scrum

3.6 Équipe Scrum

L'équipe Scrum est aussi parfois appelée « équipe de développement » puisqu'elle est responsable du développement du produit, service ou autre résultat du projet. Il consiste en un groupe de personnes auto-organisées qui travaillent sur les User Stories dans le Sprint Backlog afin de créer les produits livrables du projet. Le tableau 3-3 résume les responsabilités de l'équipe Scrum lors des différents processus Scrum.

| Processus | Responsabilités de l'équipe Scrum |
|--|--|
| 8.3 Formation de l'équipe Scrum | <ul style="list-style-type: none"> Contribue à la création de la stratégie de collaboration et du programme de consolidation de l'équipe |
| 8.4 Développement d'epics | <ul style="list-style-type: none"> Assure la compréhension claire des epics et personnages |
| 8.5 Backlog de produit priorisé | <ul style="list-style-type: none"> Comprend les user stories dans le backlog de produit priorisé |
| 8.6 Planification de la release | <ul style="list-style-type: none"> Convient de la durée du sprint avec les autres membres de l'équipe Scrum principale Cherche à clarifier les nouveaux produits ou changements aux produits existants, le cas échéant, dans le backlog de produit priorisé affiné |
| 9.1 Création des user stories | <ul style="list-style-type: none"> Offre sa contribution au Product Owner pour la création des user stories |
| 9.2 Estimation des user stories | <ul style="list-style-type: none"> Estime les user stories approuvées par le Product Owner. |
| 9.3 Validation des user stories | <ul style="list-style-type: none"> Valide les user stories à réaliser pendant un sprint |
| 9.4 Identification des tâches | <ul style="list-style-type: none"> Développe une liste des tâches basée sur les user stories et les dépendances |
| 9.5 Estimation des tâches | <ul style="list-style-type: none"> Estime les efforts requis pour réaliser les tâches identifiées et, le cas échéant, met à jour la liste des tâches |
| 9.6 Mise à jour du Sprint Backlog | <ul style="list-style-type: none"> Définit les User Stories et les tâches à inclure dans le Sprint Backlog et suivies sur le Sprint Burndown Chart |
| 10.1 Création des incréments de produit | <ul style="list-style-type: none"> Crée des produits livrables Identifie les risques et implémente des actions d'atténuation des risques Identifie les obstacles à suivre dans le Impediments Log |
| 10.2 Mêlées quotidiennes | <ul style="list-style-type: none"> Met à jour le Scrumboard tout au long de chaque sprint Discute des problèmes rencontrés par les membres individuels et cherche des solutions pour motiver l'équipe |
| 10.3 Raffinage du Product Backlog priorisé | <ul style="list-style-type: none"> Participe aux réunions de révision de Product Backlog priorisé |
| 11.1 Démonstration et validation du sprint | <ul style="list-style-type: none"> Présente les produits livrables terminés au Product Owner pour approbation |
| 11.2 Rétrospective du sprint | <ul style="list-style-type: none"> Identifie les possibilités d'amélioration du sprint en cours et convient des améliorations à apporter pour le prochain sprint |
| 12.2 Rétrospective de release | <ul style="list-style-type: none"> Participe à la réunion de rétrospective de release |

Tableau 3-3: Responsabilités de l'équipe Scrum dans les processus Scrum

3.6.1 Sélection du personnel

La figure 3-2 liste les traits désirables pour les rôles Scrum principaux.

3

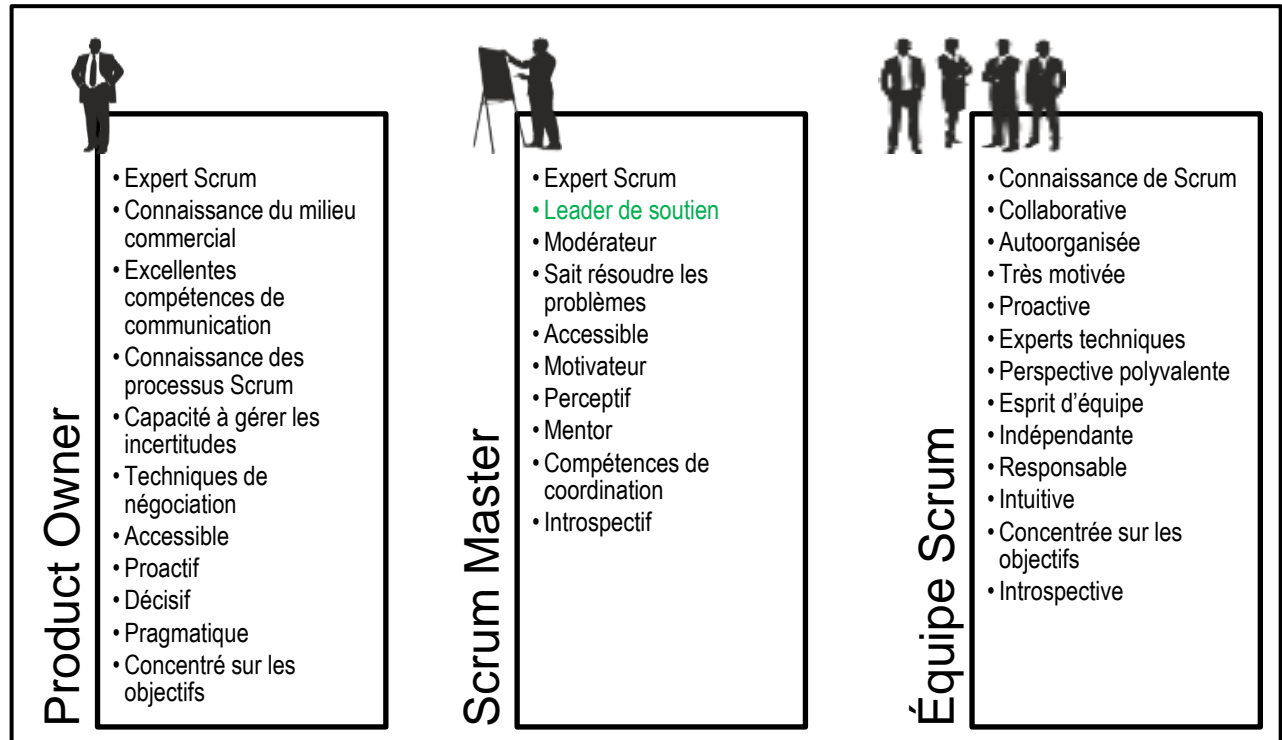


Figure 3-2: Traits désirables pour les rôles Scrum principaux

3.6.2 Taille de l'équipe Scrum

Il est important que l'équipe Scrum possède toutes les compétences essentielles requises afin d'effectuer le travail du projet. Il est également nécessaire d'avoir un haut niveau de collaboration afin d'optimiser la productivité, de façon à minimiser le niveau de coordination requise pour effectuer les tâches.

La taille idéale d'une équipe Scrum est de six à dix membres — suffisamment large pour garantir une gamme de compétences adéquate mais suffisamment petite pour collaborer facilement. L'objectif est d'avoir suffisamment de personnes dans l'équipe pour accomplir le travail, tout en restant en petit comité pour avoir une communication et une collaboration efficaces au sein de l'équipe. Cependant, il existe aussi des inconvénients. Un des principaux inconvénients est que les petites équipes sont plus largement affectées que les équipes plus importantes par la perte d'un membre, même si ce n'est que pour une courte durée. Afin de résoudre ce problème, il est possible pour les membres de l'équipe de posséder des connaissances et compétences expertes en dehors de leur rôle spécifique.

Cependant, cela peut être difficile et dépend du type de projet, du secteur et de la taille de l'organisation. Il est également recommandé d'avoir des remplaçants prêts à se substituer à toute personne susceptible de quitter l'équipe Scrum.

3.7 Scrum dans les grands projets, les programmes et les portefeuilles

3.7.1 Fonctionnement de Scrum dans un grand projet

Les processus Scrum fondamentaux définis dans les chapitres 8 à 12 sont valables pour les projets Scrum avec un Product Owner, un Scrum Master et une à trois équipes Scrum. Ces projets sont généralement considérés comme de petits projets Scrum.

Lorsqu'il s'agit de grands projets nécessitant les efforts de quatre équipes Scrum ou plus, avec plusieurs Product Owners et plusieurs Scrum Masters, les processus fondamentaux définis dans les chapitres 8 à 12 restent valables, mais certaines considérations supplémentaires et mises à jour des inputs, des outils et des outputs peuvent être nécessaires. Cela peut inclure des efforts supplémentaires de coordination et de synchronisation. Les impacts sur les processus fondamentaux Scrum lors de l'extension de Scrum à de grands projets sont décrits en détail au chapitre 13.

La définition de ce qui constitue un grand projet dépend généralement de l'organisation et/ou de la complexité des projets entrepris. Un critère clé pour déterminer si un projet est considéré comme petit ou grand est de savoir si le projet nécessite plusieurs Scrum Masters et/ou plusieurs Product Owners. Si le projet ne nécessite qu'un Scrum Master et un Product Owner, ces personnes peuvent normalement gérer tous les efforts supplémentaires de communication et de synchronisation requis par le projet.

Des inputs, des outils et des outputs supplémentaires peuvent être nécessaires pour les grands projets pour les raisons suivantes:

Product Owners

- Nécessité d'une collaboration entre les Product Owners lorsqu'ils travaillent avec les parties prenantes de l'entreprise, d'un raffinement du Product Backlog priorisé et d'une collaboration avec plusieurs équipes Scrum

Il est également important de noter qu'à mesure que Scrum est adapté aux grands projets, des services de soutien supplémentaires peuvent être nécessaires, tels que des architectes, des gestionnaires de produits, des services de conformité, de sécurité de l'information, des organes de gouvernance, etc.

Scrum Masters

- Nécessité d'une collaboration entre les Scrum Masters pour gérer les obstacles et synchroniser le travail des différentes équipes Scrum

Équipes Scrum

- Interaction et dépendances accrues entre les équipes Scrum au fur et à mesure que la complexité augmente pour un grand projet
- Nécessité de gérer les conflits, de résoudre les problèmes et de fixer les priorités entre les équipes Scrum
- Besoin de spécialisation, car certaines équipes Scrum peuvent avoir besoin de ressources spécialisées pour des tâches spécifiques (et ces compétences particulières ne sont pas nécessaires dans toutes les équipes Scrum)
- Nécessité de définir certaines lignes directrices et normes qui doivent être respectées par toutes les équipes Scrum (par exemple, les normes de sécurité au sein d'une entreprise ou les lignes directrices juridiques et gouvernementales pour des industries spécifiques) ; celles-ci peuvent être définies par le Scrum Guidance Body
- Nécessité de mettre en place un environnement ou une zone de travail pour le grand projet, qui serait ensuite utilisé par toutes les équipes Scrum
- Nécessité de coordonner les outputs de plusieurs équipes Scrum pour faciliter la release d'un grand projet.

3.7.2 Rôles principaux supplémentaires dans les grands projets

3.7.2.1 Chief Product Owner

Dans le cas de gros projets avec plusieurs équipes Scrum et plusieurs Product Owners, il est toujours nécessaire d'avoir une personne unique en charge des décisions commerciales quotidiennes. Cette personne/ce rôle est celui de Chief Product Owner. Ce rôle a la responsabilité de coordonner le travail de plusieurs Product Owners. Avec l'aide des Product Owners, le Chief Product Owner prépare et maintient le backlog de produit priorisé global pour le gros projet. Il l'utilise pour coordonner le travail des Product Owners et des équipes Scrum. Le Chief Product Owner est responsable du produit final du projet tandis que les Product Owners de chaque équipe individuelle sont uniquement responsables des composants et fonctionnalités développés par leurs équipes Scrum respectives.

Dans un gros projet, le Chief Product Owner aura comme tâche de prioriser les demandes concurrentes émises par les Product Owners en se basant sur son interaction avec parties prenantes de l'entreprise. La complexité de cette tâche augmente avec le nombre d'équipes Scrum et de Product Owners. Une part importante de la complexité de cette tâche est d'assurer que les différents composants sont correctement intégrés aux moments appropriés. Il est donc impératif de développer une liste des composants et des ressources nécessaires communes aux équipes durant le projet. Bien que le Chief Product Owner prenne les décisions commerciales finales, il collabore avec le Chief Scrum Master, les autres Product Owners et Scrum Masters afin de développer cette liste.

Le Chief Product Owner interagit également avec le Product Owner du programme afin d'assurer l'adéquation entre le gros projet et les buts et objectifs du programme.

Il est recommandé que le Chief Product Owner consulte les sections du *Guide SBOK®* définies dans le guide des rôles pour le Product Owner ainsi que le chapitre 13 qui décrit le passage à l'échelle de Scrum pour les gros projets.

3.7.2.2 Chief Scrum Master

Les gros projets requièrent plusieurs équipes Scrum travaillant en parallèle. Il se peut que les informations collectées par une équipe doivent être communiquées correctement à d'autres équipes — c'est le Chief Scrum Master qui est responsable de cette activité.

Le rôle de Chief Scrum Master est nécessaire afin d'assurer la bonne collaboration des équipes Scrum. La coordination des différentes équipes Scrum travaillant sur un projet se fait normalement lors des réunions Scrum of Scrums (SoS) (voir la section 13.2.2.1). Il n'existe aucune hiérarchie entre les Scrum Masters : ils sont tous pairs. Le Chief Scrum Master travaille seulement à un niveau multi-équipe, tandis que les Scrum Masters travaillent chacun au niveau d'une équipe unique.

La figure 3-3 présente les questions posées lors d'une réunion Scrum of Scrums (SoS).

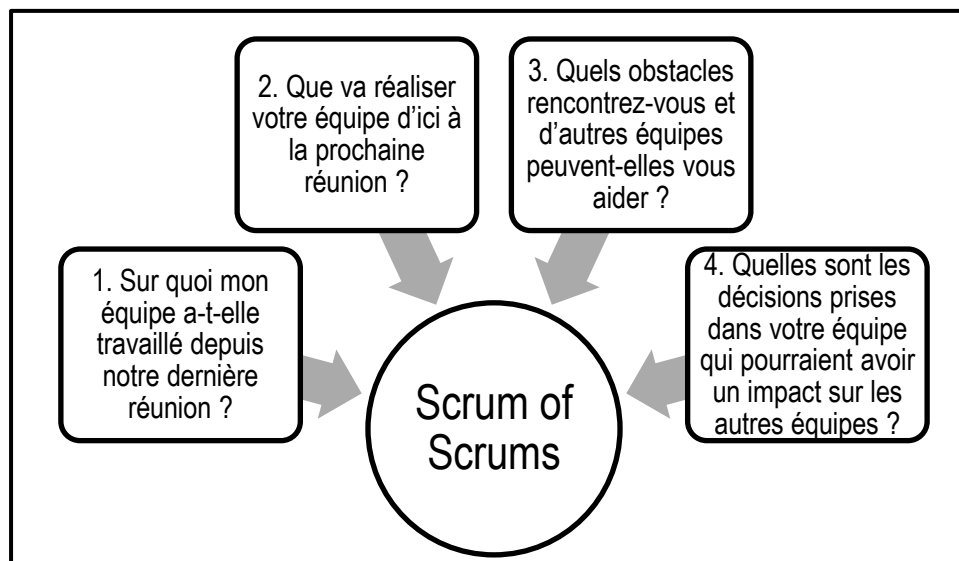


Figure 3-3: Questions posées lors d'une réunion Scrum of Scrums

Normalement, les problèmes inter-équipe sont abordés par les parties concernées lors d'une session immédiatement après la réunion Scrum of Scrums. Cette session est facilitée par le Chief Scrum Master.

Le Chief Scrum Master peut être choisi parmi les Scrum Masters du gros projet ou être une autre personne. Pour les très gros projets, il est recommandé d'avoir un Chief Scrum Master qui ne soit pas aussi le Scrum Master d'un projet individuel car les efforts requis pour le rôle de Chief Scrum Master l'empêcheront de pouvoir accorder suffisamment de temps au travail avec son équipe Scrum. Dans tous les cas, le Chief Scrum Master doit avoir suffisamment d'expertise Scrum pour être capable d'encourager la collaboration et d'aider et de coacher les autres dans la mise en œuvre de l'approche Scrum pour une livraison efficace des produits du projet.

En plus d'éliminer les obstacles et d'assurer un environnement de projet favorable pour les équipes Scrum, le Chief Scrum Master collabore avec le Chief Product Owner, les Scrum Masters et les Product Owners pour des activités telles que le développement des listes des ressources et composants communs nécessaires aux équipes du projet. Il facilite tout ce qui est en dehors du domaine d'une équipe Scrum seule.

Le Chief Scrum Master interagit également avec le Scrum Master du programme afin d'assurer l'adéquation entre le gros projet et les buts et objectifs du programme. Il est recommandé que le Chief Scrum Master consulte les sections du *Guide SBOK*[®] définies dans le guide des rôles pour le Scrum Master ainsi que le chapitre 13 qui décrit le passage à l'échelle de Scrum pour les gros projets.

3.7.3 Fonctionnement de Scrum dans un environnement d'entreprise

L'application des pratiques Scrum aux projets qui font partie d'un modèle d'entreprise implique une compréhension de la manière dont une entreprise est mise en place et gouvernée. Cela se fait généralement à l'aide de programmes et de portefeuilles.

Programme—un programme est un groupe de projets associés ayant pour objectif de fournir des résultats opérationnels tels qu'ils sont définis dans l'énoncé de la vision du programme. Le Program Backlog priorisé incorpore les Product Backlogs priorisés pour tous les projets du programme.

Portfolio—un programme est un groupe de programmes et/ou de projets associés ayant pour objectif de fournir des résultats opérationnels tels qu'ils sont définis dans l'énoncé de la vision du portfolio. Le Portfolio Backlog priorisé incorpore les Program Backlogs priorisés de tous les programmes du portfolio et le Product Backlog priorisé des projets autonomes qui font partie du portfolio.

Les problèmes et les questions rencontrés lors de l'utilisation de Scrum dans le cadre d'un programme ou d'un portfolio concernent principalement la coordination entre les nombreuses équipes. Cela peut conduire à un échec si ce dernier point n'est pas soigneusement géré. Les outils utilisés pour la communication doivent être adaptés aux besoins des nombreuses équipes impliquées dans un programme ou un portfolio. Chaque équipe Scrum doit s'occuper non seulement de la communication interne, mais aussi de la communication externe avec les autres équipes et les parties prenantes de l'entreprise concernées du programme ou du portfolio correspondant.

Lors de l'application de Scrum à la gestion de projets dans le cadre d'un programme ou d'un portfolio, il est fortement recommandé de respecter les principes généraux de Scrum présentés dans cette publication. Il est toutefois entendu que pour tenir compte de l'ensemble des activités et des interdépendances du programme ou du portfolio, des ajustements mineurs de l'ensemble des outils, ainsi que de la structure organisationnelle, peuvent être nécessaires. Si le Scrum Guidance Body existe, il peut être chargé d'examiner l'organisation à différents niveaux afin de comprendre et de définir l'application appropriée de Scrum, et d'agir comme un organisme de conseil pour tous ceux qui travaillent sur un projet, un programme ou un portfolio.

Les programmes et les portfolios ont des équipes distinctes avec des objectifs différents. Une équipe de gestion de programme vise à fournir des capacités et à réaliser certains objectifs qui contribuent à la réalisation des objectifs spécifiques du programme. En revanche, une équipe chargée du portfolio doit équilibrer les objectifs des différents programmes associés afin d'atteindre les objectifs stratégiques de l'organisation dans son ensemble.

Il est important de noter qu'à mesure que Scrum est adapté à l'entreprise, des services de soutien supplémentaires peuvent être nécessaires, tels que des architectes, des gestionnaires de produits, des services de conformité, de sécurité de l'information, des organes de gouvernance, etc.

Le chapitre 14 présente des informations détaillées sur la mise à l'échelle de Scrum pour l'entreprise.

3.7.4 Rôles essentiels supplémentaires dans un environnement d'entreprise

3.7.4.1 Program Product Owner

Le rôle de Program Product Owner est similaire au rôle de Product Owner, sauf qu'il vise à répondre aux besoins du programme ou de l'unité opérationnelle plutôt qu'à ceux d'une seule équipe Scrum.

Le Program Product Owner définit les objectifs stratégiques et les priorités du programme. Il ou elle est chargé(e) d'exprimer les exigences du client et de préserver la justification commerciale du projet. Le Program Product Owner gère également le Program Backlog priorisé. Il ou elle est responsable de la création et du raffinement des produits livrables au niveau du programme, ce qui nécessite une coordination entre les projets sous-jacents dans le programme. Le Program Product Owner est également responsable de la coordination avec les autres Program Product Owners lorsque d'autres programmes ont des dépendances communes et/ou des plans de release communs.

Le Program Product Owner se coordonne également avec le Portfolio Product Owner concerné pour s'assurer que le programme est aligné avec le portfolio correspondant. Le Program Product Owner interagit avec le Portfolio Product Owner pour assurer l'alignement du programme avec les buts et les objectifs du portfolio. Il ou elle est également impliqué(e) dans la nomination des Product Owners pour chaque projet individuel lié et s'assure que la vision, les objectifs, les résultats et les releases des projets associés s'alignent sur ceux du programme.

Les Program Product Owners doivent se référer aux sections du guide des rôles du *guide SBOK®* qui définissent le rôle du Product Owner. Des informations plus détaillées sur ce rôle sont présentées au chapitre 14, qui décrit la mise à l'échelle de Scrum pour l'entreprise.

3.7.4.2 Portfolio Product Owner

Le rôle du Portfolio Product Owner est similaire au rôle de Product Owner et également au rôle de Program Product Owner sauf qu'il vise à répondre aux besoins du portfolio ou de l'unité opérationnelle plutôt qu'aux besoins d'une seule équipe Scrum ou aux besoins d'un programme.

Le Portfolio Product Owner prend les décisions au niveau du portfolio. Il ou elle aura la meilleure perspective pour aider à décider comment organiser l'entreprise pour réaliser la vision définie. Le Portfolio Product Owner est responsable de la création et du raffinage du Portfolio Backlog priorisé.

Les Portfolio Product Owners doivent se référer aux sections du guide des rôles dans le *guide SBOK®* qui définissent le rôle du Product Owner. Des informations plus détaillées sur ce rôle sont présentées au chapitre 14, qui décrit la mise à l'échelle de Scrum pour l'entreprise.

3.7.4.3 Program Scrum Master

Le rôle de Program Scrum Master est similaire au rôle de Scrum Master, sauf qu'il vise à répondre aux besoins du programme ou de l'unité opérationnelle plutôt qu'à ceux d'une seule équipe Scrum.

Le Program Scrum Master est un médiateur qui veille à ce que toutes les équipes de projet du programme bénéficient d'un environnement propice à la réussite de leurs projets. Le Scrum Master du programme guide, anime et enseigne les pratiques Scrum à toutes les personnes impliquées dans le programme ; fournit des conseils aux Scrum Masters des projets. Il élimine les obstacles pour les différentes équipes de projet ; procède à la coordination avec le Scrum Guidance Body pour définir les objectifs liés à la qualité, aux réglementations gouvernementales, à la sécurité et à d'autres paramètres organisationnels clés et enfin, il s'assure que les processus Scrum sont effectivement suivis tout au long du programme. Il ou elle est un médiateur, qui résout les problèmes et élimine les obstacles au niveau du programme. En même temps, il ou elle est également responsable de la coordination entre tous les projets du programme et de la coordination avec d'autres programmes avec des dépendances partagées ou des plans de release partagés.

Le Program Scrum Master assure l'interface avec le Portfolio Scrum Master pour assurer l'alignement du programme avec les buts et objectifs du portfolio. Il ou elle participe également à la nomination des Scrum Masters pour les projets individuels et s'assure que la vision, les objectifs, les résultats et les releases de chaque projet du programme s'alignent sur ceux du programme.

Les Program Scrum Masters doivent se référer aux sections du guide des rôles du *guide SBOK®* qui définissent le rôle du Scrum Master. Des informations plus détaillées sur ce rôle sont présentées au chapitre 14, qui décrit la mise à l'échelle de Scrum pour l'entreprise.

3.7.4.4 Portfolio Scrum Master

Le rôle du Portfolio Scrum Master est similaire au rôle du Scrum Master, sauf qu'il vise à répondre aux besoins du portfolio ou de l'unité opérationnelle plutôt qu'à ceux d'une seule équipe Scrum.

Les Portfolio Scrum Masters doivent se référer aux sections du guide des rôles dans le *guide SBOK*[®] qui définissent le rôle du Scrum Master. Des informations plus détaillées sur ce rôle sont présentées au chapitre 14, qui décrit la mise à l'échelle de Scrum pour l'entreprise.

3.7.5 Exemples de projets, programmes et portfolios

Voici des exemples de projets, de programmes et de portfolios d'industries et secteurs différents :

Exemple 1: Entreprise de construction

- Projet—construction d'une maison
- Programme—construction d'un complexe de logements
- Portfolio—tous les projets de logement de l'entreprise

Exemple 2: Organisation aérospatiale

- Projet—construction du véhicule de lancement
- Programme—lancement réussi d'un satellite
- Portfolio—tous les programmes satellites actifs

Exemple 3: Société de technologie de l'information (TI)

- Projet—développement du module de panier d'achat
- Programme—développement d'un site web de commerce électronique entièrement fonctionnel
- Portfolio—tous les sites web développés par la société jusqu'à présent

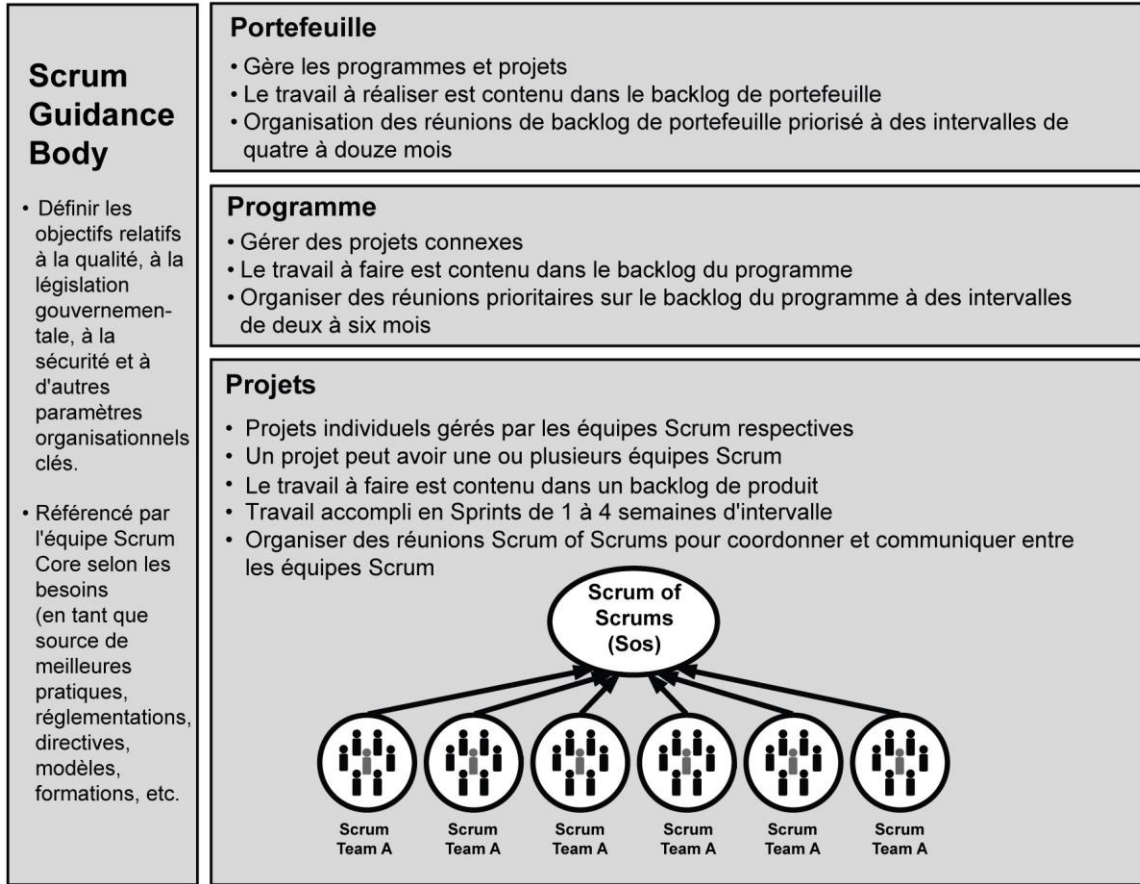


Figure 3-4: Scrum dans les organisations pour les projets, programmes ou portefeuilles

3.8 Résumé des responsabilités

| Rôle | Responsabilités |
|---------------------------------------|---|
| Équipe Scrum | <ul style="list-style-type: none"> • Assume la responsabilité collective et s'assure que les produits du projet sont créés conformément aux exigences • Assure le Product Owner et le Scrum Master que le travail attribué est réalisé conformément au plan |
| Product Owner/ Chief Product Owner | <ul style="list-style-type: none"> • Détermine les exigences globales initiales et lance le projet • Aide à nommer les personnes adéquates pour les rôles de Scrum Master et de l'équipe Scrum • Aide à sécuriser les ressources financières initiales et continues pour le projet • Détermine la vision du produit • Évalue la viabilité et assure la livraison du produit ou service • Garantit la transparence et la clarté des éléments du backlog de produit priorisé • Décide du contenu minimum nécessaire à la commercialisation de la release |

| | |
|-------------------------------------|---|
| | <ul style="list-style-type: none"> • Fournit les critères d'acceptation des user stories à développer pendant un sprint • Inspecte les itérations de produits • Décide de la durée du sprint |
| Scrum Master/ Chief Scrum Master | <ul style="list-style-type: none"> • S'assure que les processus Scrum sont respectés par tous les membres de l'équipe y compris le Product Owner • S'assure des progrès du développement du produit ou service et que les membres de l'équipe Scrum disposent des outils nécessaires afin d'effectuer le travail. • Supervise la réunion de planification de la release et programme les autres réunions |
| Product Owner du programme | <ul style="list-style-type: none"> • Définit les objectifs et priorités stratégiques pour les programmes |
| Scrum Master du programme | <ul style="list-style-type: none"> • Résout les problèmes et coordonne les réunions pour les programmes |
| Product Owner du portefeuille | <ul style="list-style-type: none"> • Définit les objectifs et priorités stratégiques pour les portefeuilles |
| Scrum Master du portefeuille | <ul style="list-style-type: none"> • Résout les problèmes et coordonne les réunions pour les portefeuilles |
| Parties prenantes de l'entreprise | <ul style="list-style-type: none"> • Un terme collectif qui inclut les clients, utilisateurs et sponsors • Interagit régulièrement avec le Product Owner, le Scrum Master et l'équipe Scrum afin de fournir des informations et de faciliter le développement des incréments de produits pour le projet. |
| Scrum Guidance Body | <ul style="list-style-type: none"> • Établit les directives globales ainsi que les mesures pour le développement des rôles décrits pour les membres de l'équipe Scrum • Agit comme conseiller pour les projets de toute l'organisation à différents niveaux • Comprend et définit les niveaux de regroupement, les rôles et les réunions adéquats pour les projets Scrum |

Tableau 3-4: Résumé des responsabilités pertinentes à l'organisation

3.9 L'approche Scrum comparée à la gestion de projet traditionnelle

La structure de l'organisation et la définition des rôles et des responsabilités qui y sont associées sont quelques-uns des éléments qui permettent de différencier Scrum des méthodes traditionnelles de gestion de projet.

Dans les méthodes de gestion de projet traditionnelles, la structure d'organisation est hiérarchisée et l'autorité pour tous les aspects du projet est déléguée de haut en bas (p. ex. le sponsor du projet délègue l'autorité au chef de projet et le chef de projet délègue l'autorité aux membres de l'équipe). Les méthodes de gestion de projet traditionnelles mettent l'accent sur la responsabilité individuelle pour les responsabilités relatives au projet plutôt que sur la responsabilité et l'appropriation du projet par le groupe. Toute déviation par rapport à l'autorité déléguée est considérée comme signe de problèmes et peut être signalée au niveau hiérarchique supérieur dans l'organisation.

C'est normalement le chef de projet qui est responsable de la réussite du projet et il prend les décisions concernant différents aspects du projet, notamment le lancement, la planification, l'estimation, l'exécution, la surveillance et le contrôle, ainsi que la clôture.

Dans Scrum, l'accent est mis sur l'auto-organisation et la motivation et accorde à l'équipe une plus grande responsabilité quant à la réussite du projet. Cela permet aussi de garantir l'implication de l'équipe et le partage de la responsabilité. Par conséquent, l'équipe est plus motivée et son efficacité est optimisée. Le Product Owner, le Scrum Master et l'équipe Scrum collaborent étroitement avec parties prenantes de l'entreprise concernées afin d'affiner les exigences lors des processus de *développement des epics*, de *création du backlog de produit priorisé* et de *création des user stories*. Cela permet de garantir qu'il n'y a pas de marge pour une planification isolée dans la mise en œuvre de Scrum. L'expérience et l'expertise de l'équipe en matière de développement de produit sont utilisées afin d'évaluer les contributions nécessaires, ainsi que pour planifier, estimer et réaliser le travail du projet. La collaboration entre les membres de l'équipe Scrum principale permet de garantir que le projet est réalisé dans un environnement innovant et créatif favorable à la croissance.

3.10 Théories RH populaires et leur importance dans Scrum

3.10.1 Le modèle de Tuckman sur la dynamique des groupes

Au départ, l'approche et la méthode Scrum peuvent sembler assez différentes et compliquées pour une nouvelle équipe Scrum. Comme toute nouvelle équipe, une nouvelle équipe Scrum évolue généralement à travers un processus de quatre phases lors de son premier projet Scrum. Ce processus est connu sous le nom de modèle de Tuckman sur la dynamique des groupes (Tuckman, 1965). L'idée principale est que ces quatre phases — formation, tension, normalisation et exécution — sont impératives au développement de l'équipe et permettent d'atténuer les problèmes, de trouver des solutions, et planifier le travail et de livrer des résultats. Un Scrum Master doit être conscient du stade auquel se trouve l'équipe Scrum et l'aider à devenir une équipe plus performante.

Les quatre phases de ce modèle sont les suivantes :

1. **La phase de formation** — elle est souvent considérée comme une étape ludique du fait que tout est nouveau et que l'équipe n'a pas encore rencontré de difficultés avec le projet.
2. **La phase de tension** — pendant cette phase l'équipe tente d'accomplir le travail, cependant, des luttes de pouvoir peuvent avoir lieu et il y a souvent du chaos ou de la confusion au sein de l'équipe.
3. **La phase de normalisation** — c'est quand l'équipe commence à mûrir, est capable de résoudre les différents internes et de trouver des solutions pour travailler ensemble. C'est une période d'ajustement.
4. **La phase d'exécution** — pendant cette phase l'équipe devient plus cohésive et opère à son plus haut niveau de performance. Les membres ont évolué en une équipe efficace de pairs professionnels uniformément productifs.

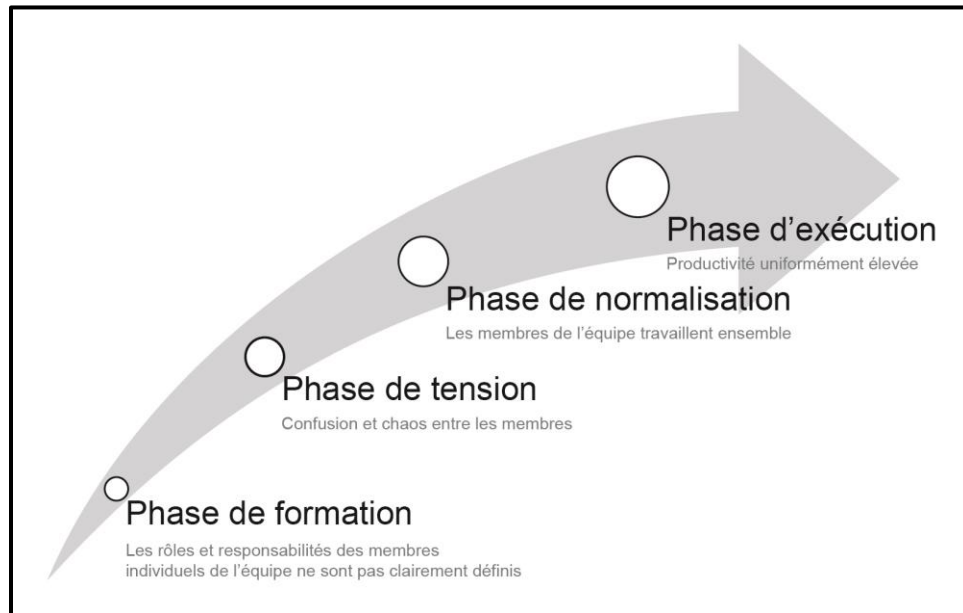


Figure 3-5: Les phases du modèle de Tuckman sur le développement des groupes

3.10.2 Gestion des conflits

Les organisations qui appliquent Scrum encouragent un environnement ouvert et le dialogue entre les employés. Les conflits entre les membres de l'équipe Scrum sont normalement résolus de façon indépendante, avec peu ou pas de participation de la part des responsables ou de personnes extérieures à l'équipe.

Les conflits sont sains lorsqu'ils encouragent les discussions et les débats au sein de l'équipe, le résultat bénéficie généralement au projet ainsi qu'aux membres de l'équipe concernés. Il est donc important d'encourager la résolution des conflits, de promouvoir un environnement ouvert où les membres de l'équipe se sentent libres d'exprimer leurs opinions, leurs inquiétudes entre eux ainsi qu'au sujet du projet, et parviennent finalement à un accord sur ce qui doit être livré ainsi que sur la façon dont le travail doit être effectué pour chaque Sprint.

Les membres de l'équipe utilisent des techniques de gestion des conflits afin de gérer les conflits qui pourraient survenir durant un projet Scrum. Le plus souvent, les sources de conflits incluent les facteurs suivant : les calendriers, les priorités, les ressources, la hiérarchie, les problèmes techniques, les procédures, les personnalités et les coûts.

3.10.3 Techniques de gestion des conflits

Pour une organisation appliquant les processus Scrum, il existe normalement quatre approches pour la gestion des conflits :

1. Gagnant-gagnant
2. Perdant-gagnant
3. Perdant-perdant
4. Gagnant-perdant

3.10.3.1 Gagnant-gagnant

Il est normalement plus avantageux pour les membres de l'équipe de résoudre les problèmes directement en adoptant une attitude coopérative et en encourageant un dialogue ouvert afin de régler les désaccords et de parvenir à un consensus. Cette approche est appelée *gagnant-gagnant*. Les organisations qui utilisent Scrum doivent promouvoir un environnement où les employés se sentent à l'aise pour discuter, confronter ouvertement les problèmes ou difficultés et les faire avancer afin de parvenir à un résultat gagnant-gagnant.

3.10.3.2 Perdant-gagnant

Certains membres de l'équipe peuvent parfois avoir l'impression que leurs contributions ne sont pas reconnues ou appréciées par les autres, ou qu'ils ne sont pas traités sur un pied d'égalité. Cela peut les empêcher de contribuer au projet de façon efficace et d'accepter tout ce qu'on leur dit de faire, même s'ils ne sont pas d'accord. Cette approche est appelée *perdant-gagnant*. Cette situation peut survenir s'il y a des membres dans l'équipe (y compris des responsables) qui utilisent un style autoritaire ou directif pour donner des ordres et/ou ne traitent pas tous les membres de l'équipe de la même façon. Cette approche n'est pas une technique de gestion des conflits désirable pour un projet Scrum puisque la contribution de chaque membre de l'équipe est essentielle à la réussite de chaque sprint. Le Scrum Master doit encourager la participation de tous les membres de l'équipe qui semblent se tenir en retrait par rapport aux situations conflictuelles. Par exemple, il est important que chaque membre de l'équipe participe et contribue à chaque mêlée quotidienne de façon à ce que tous les problèmes ou obstacles soient connus et gérés efficacement.

3.10.3.3 Perdant-perdant

Dans les situations de conflit, les membres de l'équipe peuvent tenter de négocier ou de trouver des solutions qui n'offrent qu'une mesure ou un degré de satisfaction partiel ou temporaire aux parties concernées. Cette situation peut survenir dans les équipes Scrum où les membres de l'équipe tentent de négocier des solutions suboptimales pour un problème. Plutôt que d'essayer de résoudre le vrai problème, cette approche implique généralement du donnant-donnant afin de satisfaire tous les membres de l'équipe.

Cela résulte généralement en une situation *perdant-perdant* pour les individus concernés et par conséquent pour le projet. L'équipe Scrum doit faire attention à ce que les membres de l'équipe n'adoptent pas une mentalité perdant-perdant. Les mêlées quotidiennes et autres réunions Scrum sont organisées afin de garantir que les problèmes sont résolus à travers des échanges mutuels.

3.10.3.4 Gagnant-perdant

À certains moments, un Scrum Master ou un autre membre influent de l'équipe peut penser qu'il est de fait le chef ou responsable et tenter d'imposer ses points de vue au détriment de ceux des autres. Cette technique de gestion des conflits est souvent caractérisée par la compétitivité et aboutit généralement à un résultat *gagnant-perdant*. Cette approche n'est pas recommandée lors des projets Scrum car les équipes Scrum sont par nature autoorganisées et responsables et aucune personne n'a de réelle autorité sur aucun autre membre de l'équipe. Même si l'équipe Scrum peut inclure des personnes possédant différents degrés d'expérience et d'expertise, chaque membre est traité de la même façon et personne n'a l'autorité pour être le décideur principal.

3.10.4 Styles de leadership

Les types de leadership diffèrent en fonction de l'organisation, de la situation et même des individus spécifiques et des objectifs du projet Scrum. La liste ci-dessous illustre quelques styles de leadership courants :

- **Leadership de soutien** — les leaders de soutien utilisent l'écoute, l'empathie, l'engagement et la perspicacité tout en partageant le pouvoir et l'autorité avec les membres de l'équipe. Les leaders de soutien sont des coordonnateurs qui obtiennent des résultats en se basant sur les besoins de leur équipe. Ce style est l'illustration parfaite du rôle de Scrum Master.
- **Leadership déléguant** — un leader déléguant est impliqué dans la majorité des processus de décision. Cependant, il délègue une partie des responsabilités d'organisation et de décision aux membres de l'équipe, en particulier si ceux-ci sont à même de gérer les tâches attribuées. Ce type de leadership convient aux situations où le leader est en phase avec les détails spécifiques du projet et où le temps est limité.
- **Leadership autocratique** — les leaders autocratiques prennent leurs décisions seuls, ne permettant peu ou pas de participation ou de discussion de la part de l'équipe avant que la décision ne soit prise. Ce style de leadership ne doit être utilisé qu'en de rares occasions.
- **Leadership directif** — les leaders directifs indiquent aux membres de l'équipe quelles tâches doivent être réalisées, quand et comment.
- **Leadership laissez-faire** — avec ce style de leadership l'équipe est généralement laissée sans supervision et le leader n'interfère pas avec les activités de travail quotidiennes. Ce style conduit souvent à un état d'anarchie.

- **Coaching/Supportive leadership** — un supportive leader donne des instructions, supporte et suit les progrès des membres de son équipe en écoutant, aidant, encourageant et présentant une attitude positive dans les moments d'incertitude.
- **Task-Oriented leadership** — un task-oriented leader s'assure que les tâches sont terminées tout en respectant les échéances.
- **Leadership assertif** — un leader assertif affronte les problèmes et fait preuve de confiance pour établir l'autorité avec respect.

3.10.4.1 Leadership de soutien

Étoffant sur la théorie de Greenleaf, Larry Spears identifie dix traits que tout leader serviteur efficace devrait posséder :

1. **L'écoute** — il est attendu des leaders de soutien qu'ils soient capables d'écouter attentivement et réceptifs à ce qui est dit, ou non-dit. Ils doivent être capables d'écouter leur voix intérieure et de réfléchir à leurs propres sensations.
2. **L'empathie** — les bons leaders de soutien acceptent et reconnaissent les individus pour leurs compétences et capacités spéciales et uniques. Ils partent du principe que les travailleurs sont bien intentionnés et les acceptent en tant qu'individus, même lorsqu'il existe des problèmes de comportement ou de performance.
3. **L'apaisement** — la motivation et le potentiel à s'apaiser eux-mêmes et à apaiser leurs relations avec les autres est une caractéristique forte chez les leaders de soutien. Les leaders de soutien reconnaissent et saisissent l'opportunité d'aider leurs collègues lorsque ceux-ci éprouvent des douleurs émotionnelles.
4. **La conscience** — la conscience et plus particulièrement la conscience de soi est un trait des leaders de soutien. Elle leur permet de mieux comprendre et intégrer les problèmes, en particulier ceux associés à l'éthique, aux pouvoirs ou aux valeurs.
5. **La persuasion** — les leaders de soutien utilisent la persuasion, plutôt que l'autorité conférée par leur position, afin d'atteindre un consensus de groupe et de prendre des décisions. Plutôt que de forcer le respect et d'utiliser la contrainte comme c'est souvent le cas avec les styles de leadership autoritaires, les leaders de soutien pratiquent la persuasion.
6. **La conceptualisation** — la capacité à percevoir et analyser les problèmes (dans une organisation) à partir d'une perspective conceptuelle et visuelle plus large, plutôt que de se concentrer sur les objectifs immédiats à long terme, est une compétence propre aux leaders de soutien.

7. **L'anticipation** — leur esprit intuitif permet aux leaders de soutien d'utiliser et d'appliquer les enseignements précédents aux réalités dans le présent afin d'anticiper les résultats des situations et décisions actuelles.
8. **L'intendance** — l'intendance requiert de s'engager à servir les autres. Les leaders de soutien préfèrent la persuasion au contrôle afin de gagner la confiance des autres membres de l'organisation.
9. **Le dévouement pour le développement des autres** — les leaders de soutien sont profondément dévoués à encourager le développement des autres personnes au sein de leur organisation. Ils prennent la responsabilité de cultiver le développement personnel, professionnel et spirituel des autres (p. ex. en offrant accès à des ressources pour le développement personnel et professionnel, en encourageant les travailleurs à participer aux processus de décisions).
10. **Le développement communautaire** — les leaders de soutien sont impliqués dans le développement de communautés au sein d'un environnement de travail, en particulier étant donné la mutation de la société des petites communautés vers les grandes institutions qui forment et contrôlent les vies humaines.

Selon Scrum, tous les leaders de projets Scrum (y compris le Scrum Master et le Product Owner) doivent être des leaders de soutien et posséder les traits ci-dessus.

3.10.5 La pyramide des besoins de Maslow

Maslow (1943) présente une hiérarchie (pyramide) des besoins qui reconnaît que différentes personnes se trouvent à différents niveaux concernant leurs besoins. Généralement, les gens cherchent à satisfaire leurs besoins physiologiques et évoluent ensuite progressivement vers le haut de la pyramide.

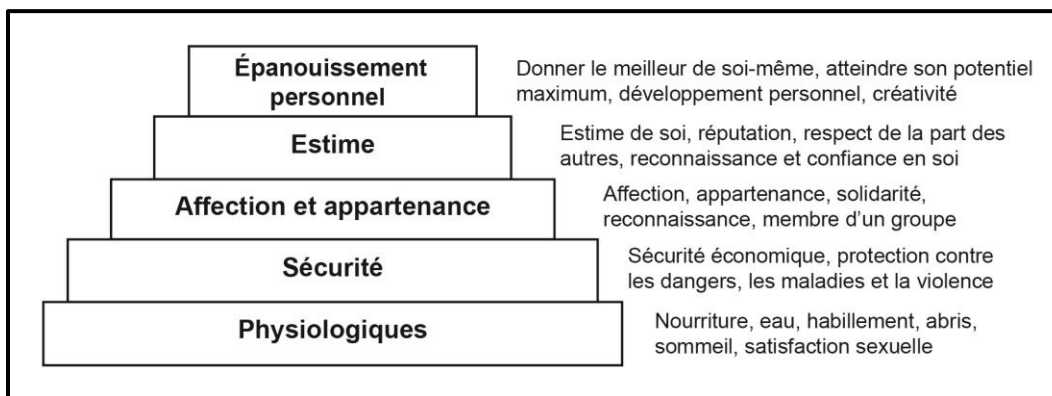


Figure 3-6: La pyramide des besoins de Maslow

Afin d'être performante l'équipe Scrum doit être composée de membres principaux et non-essentiels qui ont atteint les niveaux d'estime ou d'accomplissement de soi. Le concept d'équipes autoorganisées, qui est un principe fondamental de Scrum, nécessite des membres de l'équipe qu'ils soient motivés et participent et contribuent pleinement à la réalisation des objectifs du projet.

En tant que leader, le Scrum Master doit être en mesure de comprendre où se trouve chaque personne de l'équipe par rapport à la pyramide. Cela aidera à déterminer la meilleure approche à adopter afin de motiver chaque individu.

De plus, le niveau de chaque personne sur la pyramide varie durant sa vie, en fonction de sa propre motivation et de ses efforts pour progresser dans la pyramide ou parfois à cause de facteurs hors de son contrôle qui peuvent la tirer vers le bas. L'objectif du Scrum Master est de travailler avec les individus présents dans l'équipe afin de renforcer leurs compétences et connaissances et de les aider à progresser dans la pyramide des besoins. Cela aboutit à une équipe composée d'individus motivés et qui contribuent activement au projet et à l'organisation dans son ensemble.

3.10.6 Théorie X et théorie Y

Douglas McGregor (1960) propose deux théories de gestion :

- **La théorie X** — les leaders qui s'appuient sur la théorie X partent du principe que les employés sont naturellement peu motivés et évitent le travail lorsque c'est possible. Cette théorie induit un style de leadership autoritaire.
- **La théorie Y** — les leaders qui s'appuient sur la théorie Y partent du principe que les employés sont motivés et cherchent à accepter des responsabilités plus importantes. Cette théorie induit un style de leadership plus participatif.

Abraham H. Maslow (1960) a proposé la théorie Z et William Ouchi (1980) a fourni une autre version de la théorie Z, en développant la théorie X et la théorie Y :

- **Théorie Z**—dans la version de Maslow, les leaders de la théorie Z partent du principe que les employés sont motivés en exploitant leur désir de dépassement de soi sans ignorer leurs motivations liées à la hiérarchie des besoins. Dans la version d'Ouchi, les leaders de la théorie Z partent du principe que les employés peuvent être motivés en favorisant la stabilité par la sécurité de l'emploi, un moral élevé et la satisfaction au travail et dans la sphère privée.

Il est peu probable que les projets Scrum réussissent dans les organisations où les rôles de Scrum Master ou Product Owner sont occupés par des leaders qui s'appuient sur la théorie X. Tous les leaders de projets Scrum devraient souscrire à la théorie Y, selon laquelle les individus sont considérés comme des atouts. Ils doivent travailler au développement des compétences et à la responsabilisation des membres de leur équipe tout en exprimant leur appréciation pour le travail réalisé par l'équipe en vue de réaliser les objectifs du projet.

4. JUSTIFICATION COMMERCIALE

4.1 Introduction

L'objectif de ce chapitre est de comprendre le concept et l'utilité de la justification commerciale en relation aux projets Scrum. Il est important pour chaque organisation d'établir une justification commerciale adéquate et de créer un énoncé de vision de projet avant de commencer tout projet. Cela aide les décideurs clés à comprendre les besoins commerciaux associés à un changement de produit ou à un nouveau produit ou service, ainsi que la justification en faveur de la validation d'un projet. Cela aide également le Product Owner à créer le backlog de produit priorisé avec les attentes commerciales de la haute direction et des parties prenantes de l'entreprise .

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la *justification commerciale* s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce chapitre est divisé comme suit :

4.2 Guide des rôles — cette section offre des conseils sur les sections pertinentes à chacun des rôles principaux Scrum : Product Owner, Scrum Master et équipe Scrum.

4.3 Livraison de la valeur ajoutée — cette section décrit le concept de valeur commerciale et son importance dans chaque projet. Elle fournit également des informations concernant les responsabilités des différents individus, dont le Product Owner, qui participent à la réalisation de la valeur commerciale.

4.4 Importance de la justification commerciale — cette section détaille l'importance de la justification commerciale, les facteurs qui la détermine et la façon dont elle est maintenue et vérifiée tout au long du projet.

4.5 Techniques de justification commerciale — cette section décrit en détail la manière dont la justification commerciale est évaluée et vérifiée en utilisant certains outils.

4.6 Justification de valeur continue — cette section détaille l'importance de la continuité de la justification de la valeur et s'attarde sur la façon dont elle est réalisée.

4.7 Confirmer la réalisation des bénéfices — cette section décrit comment les bénéfices sont réalisés tout au long du projet.

4.8 Résumé des responsabilités — cette section définit les responsabilités pertinentes à la justification commerciale pour les membres de l'équipe du projet en fonction de leurs rôles.

4.9 L'approche Scrum comparée à la gestion de projet traditionnelle — cette section souligne les avantages commerciaux de Scrum comparés aux méthodes de gestions de projet traditionnelles.

4.2 Guide des rôles

1. Product Owner — la justification commerciale est essentiellement la responsabilité du Product Owner, ce chapitre dans son ensemble s'applique donc principalement à ce rôle.
2. Scrum Master — le Scrum Master doit se familiariser avec ce chapitre entier en portant une attention particulière aux sections 4.3, 4.4, 4.6, 4.7 et 4.8.
3. Équipe Scrum — l'équipe Scrum devrait surtout se concentrer sur les sections 4.3, 4.7 et 4.8.

4.3 Livraison de la valeur ajoutée

Un projet est une initiative collaborative visant soit à créer de nouveaux produits ou services soit à fournir des résultats tels qu'ils ont été définis dans l'énoncé de vision du projet. En général, les projets sont affectés par des contraintes de temps, de coûts, de portée, de qualité, de personnels et de capacité organisationnelle. Normalement, il est attendu que les résultats générés par les projets créent de la valeur commerciale ou une valeur de service sous une forme ou une autre.

Puisque la valeur ajoutée est la principale raison qui pousse une organisation à faire avancer un projet, la livraison de la valeur ajoutée doit en être l'objectif principal. L'offre de valeur ajoutée est profondément ancrée dans l'approche Scrum. Scrum facilite la livraison de la valeur ajoutée le plus tôt possible au cours d'un projet et de façon continue tout au long du cycle du projet.

L'une des caractéristiques fondamentales de tout projet est l'incertitude relative à son aboutissement et à ses résultats. Il est impossible de garantir la réussite d'un projet à la fin de celui-ci, peu importe sa taille ou son niveau de complexité. Au vu de cette incertitude quant à la réussite, il est important d'offrir des résultats aussi tôt que possible au cours du projet. La livraison anticipée des résultats, et donc de la valeur ajoutée, offre une opportunité de réinvestissement et atteste de la valeur du projet pour parties prenantes de l'entreprise intéressées.

De façon à offrir de la valeur ajoutée, il est important de :

1. comprendre ce qui a de la valeur aux yeux des clients et des utilisateurs ainsi que de hiérarchiser les exigences à forte valeur en haut du backlog de produit priorisé.
2. diminuer les incertitudes et de continuellement chercher à atténuer les risques qui pourraient réduire la valeur s'ils venaient à survenir. collaborer étroitement avec parties prenantes de l'entreprise du projet pour leur montrer les incréments de produit à la fin de chaque sprint et permettre la gestion efficace des changements.
3. *créer des incréments de produit* qui s'appuient sur les priorités établies en produisant des incréments de produit potentiellement livrables durant chaque sprint de façon à ce que les clients commencent à prendre conscience de la valeur aussi tôt que possible durant le projet.

Le concept de livraison axée sur la valeur ajoutée fait de Scrum un cadre de travail très attractif pour parties prenantes de l'entreprise des entreprises et les cadres supérieurs. Ce concept est très différent lorsque comparé aux modèles de gestion de projet traditionnels où :

1. les exigences ne sont pas priorisées en fonction de leur valeur commerciale.
2. la modification des exigences après l'initiation du projet est difficile et ne peut être réalisée qu'à travers un long processus de gestion des changements.
3. la valeur est uniquement atteinte à la fin du projet lorsque le produit ou service final est livré.

La figure 4-1 compare la livraison axée sur la valeur ajoutée dans les projets Scrum et dans les projets traditionnels.

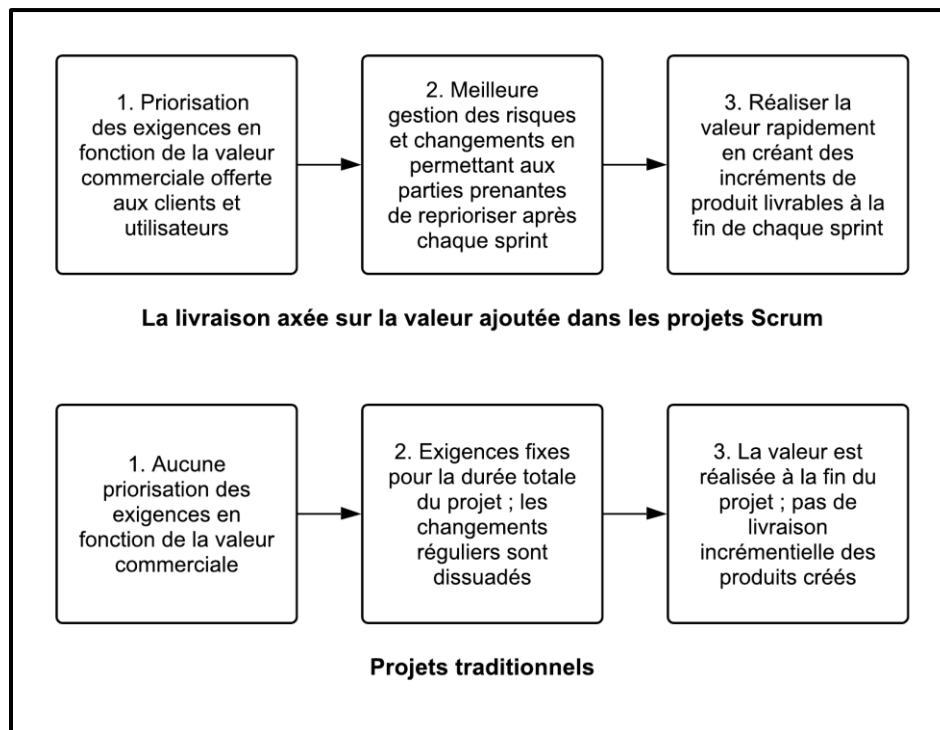


Figure 4-1: Livraison de la valeur ajoutée dans Scrum comparée aux projets traditionnels

4.3.1 Responsabilités du Product Owner concernant la justification commerciale

La responsabilité de prioriser et de fournir la valeur commerciale à une organisation pour un projet incombe principalement au Product Owner. Pour les programmes et les portefeuilles, cette responsabilité incombe respectivement au Product Owner du programme et au Product Owner du portefeuille. Leur rôle est de représenter efficacement le client et/ou le sponsor. Les directives pour évaluer et mesurer la valeur commerciale peuvent normalement être établies par le Scrum Guidance Body.

La figure 4-2 illustre les responsabilités associées à la justification commerciale dans l'ordre hiérarchique.

| | |
|-------------------------------|---|
| Product Owner du portefeuille | <ul style="list-style-type: none"> • Fournit la valeur pour les portefeuilles • Crée la justification commerciale pour les portefeuilles • Offre des conseils aux programmes au sujet de la valeur • Approuve la justification commerciale des programmes |
| Product Owner du programme | <ul style="list-style-type: none"> • Fournit la valeur pour les programmes • Crée la justification commerciale pour les programmes • Offre des conseils aux projets au sujet de la valeur • Approuve la justification commerciale des projets |
| Product Owner | <ul style="list-style-type: none"> • Fournit la valeur pour les projets • Crée la justification commerciale pour les projets • Confirme la réalisation des bénéfices aux parties prenantes de l'entreprise |

Tableau 4-1: Hiérarchie des responsabilités concernant la justification commerciale

4.3.2 Responsabilités des autres rôles Scrum concernant la justification commerciale

Il est important de noter que, même si le Product Owner est le principal responsable de la justification commerciale, d'autres personnes travaillant sur des projets Scrum y contribuent aussi de manière non négligeable :

1. Le **sponsor** finance et surveille constamment le projet pour s'assurer de la réalisation des bénéfices.

2. Les **clients** et les **utilisateurs** participent à la définition de la liste hiérarchisée des exigences et user stories dans le backlog de produit priorisé, à la revue des incréments de produit après chaque Sprint ou chaque release et confirment la réalisation des bénéfices.
3. Le **Scrum Guidance Body** peut fournir des directives et recommandations relatives aux techniques de justification commerciale, confirmer la réalisation des bénéfices etc. De telles directives et recommandations peuvent ensuite être consultées par l'équipe Scrum principale et parties prenantes de l'entreprise.
4. Le **Scrum Master** facilite la création d'incrément de produits, gère les risques, changements et obstacles durant les *mêlées quotidiennes*, la *rétrospective de sprint* et les autres processus Scrum. Le Scrum Master collabore avec l'équipe Scrum afin de créer les incréments de produit et avec le Product Owner et les autres parties prenantes de l'entreprise afin de garantir que les bénéfices du projet sont réalisés.
5. L'**équipe Scrum** travaille à la création des incréments de produit du projet et contribue à la création de la valeur commerciale pour toutes parties prenantes de l'entreprise et pour le projet. L'équipe Scrum participe également aux processus de *développement des epics*, *création du backlog de produit priorisé*, *création des user stories*, *estimation des user stories*, *validation des user stories* ainsi qu'aux autres processus associés où les exigences commerciales sont définies et hiérarchisées. L'équipe Scrum contribue également à identifier les risques et soumet des demandes de changement pour des améliorations lors des réunions de rétrospective de sprint et autres réunions.

4.4 Importance de la justification commerciale

La justification commerciale justifie les raisons d'entreprendre un projet. Elle répond à la question « Pourquoi ce projet est-il nécessaire ? » La justification commerciale motive toutes les décisions prises en lien avec un projet. Il est également important d'évaluer la viabilité et la faisabilité d'un projet avant d'engager des dépenses ou des investissements considérables lors de la phase initiale du projet mais aussi de vérifier la continuation de la justification commerciale tout au long du cycle du projet. Il faut annuler le projet si celui-ci devient infaisable, cette décision revient aux parties prenantes de l'entreprise et aux cadres concernés. La justification commerciale d'un projet doit être évaluée au début d'un projet, à des intervalles prédéfinis durant le projet et aux moments où des problèmes ou risques majeurs surviennent qui menacent la viabilité du projet

4.4.1 Paramètres utilisés pour déterminer la justification commerciale

Le Product Owner doit prendre en compte de nombreux paramètres afin de déterminer la justification commerciale d'un projet. Les paramètres les plus importants sont listés ci-dessous :

1. Justification du projet

La justification du projet inclut tous les facteurs qui justifient la nécessité du projet, qu'ils soient positifs ou négatifs, choisis ou non (p. ex. capacité inadaptée à répondre aux demandes existantes ou prévues, baisse du taux de satisfaction des clients, faibles bénéfices, exigences légales, etc.).

2. Besoins commerciaux

Les besoins commerciaux sont les besoins auxquels le projet est censé répondre, tels qu'ils sont documentés dans l'énoncé de vision du projet.

3. Bénéfices du projet

Les bénéfices du projet incluent toutes les améliorations tangibles apportées à un produit, service ou résultat qui peut être obtenu par la réalisation d'un projet.

4. Coût d'opportunité

Le coût d'opportunité fait référence à la valeur de la prochaine meilleure option ou du prochain meilleur projet commercial à avoir été rejeté en faveur du projet choisi.

5. Risques majeurs

Les risques incluent tout événement incertain ou non planifié qui pourrait affecter la viabilité et la réussite potentielle d'un projet.

6. Échéancier du projet

Les échéances reflètent la longueur et la durée d'un projet ainsi que la durée sur laquelle les bénéfices seront réalisés.

7. Coûts du projet

Les coûts du projet sont les investissements et autres coûts relatifs au développement d'un projet.

4.4.2 Justification commerciale et cycle du projet

Dans un premier temps, la justification commerciale est évaluée avant d'initier un projet, elle est ensuite continuellement vérifiée tout au long du cycle du projet. Les étapes suivantes reflètent la façon dont la justification commerciale est déterminée :

1. Évaluer et présenter une étude de cas

Normalement, la justification commerciale pour un projet est évaluée et approuvée par le Product Owner. Elle est également documentée et présentée sous la forme d'une étude de cas de projet avant la phase d'initiation et implique de considérer les différents paramètres spécifiés dans la section 4.4.1. Une fois documentée, le Product Owner doit créer un énoncé de vision du projet et obtenir son approbation de la part des décideurs principaux dans l'organisation. Généralement, ceux-ci incluent les dirigeants et/ou des membres du conseil de gestion du projet ou programme.

2. Justification de valeur continue

Une fois que les décideurs approuvent l'énoncé de vision du projet, celui-ci devient un point de référence et forme la justification commerciale. La justification commerciale est validée tout au long du projet, normalement à des intervalles prédéfinis ou lorsque des étapes majeures sont franchies, par exemple lors des réunions de revue du backlog de produit priorisé du portfolio, programme ou projet, ou lorsque des problèmes et des risques majeurs sont identifiés et menacent la viabilité du projet. Cela peut se produire au cours de différents processus Scrum, notamment lors des *mêlées quotidiennes* ou de l'*Raffinage du Product Backlog priorisé*. Le Product Owner doit mettre à jour la justification commerciale dans l'énoncé de vision du projet tout au long du projet en prenant soin d'inclure les informations pertinentes afin de permettre aux décideurs principaux de continuer à prendre des décisions éclairées.

3. Confirmer la réalisation des bénéfices

Le Product Owner confirme la réalisation des bénéfices organisationnels tout au long du projet ainsi que lors de la réalisation des user stories du backlog de produit priorisé. Les bénéfices des projets Scrum sont réalisés lors des processus de *démonstration et validation du sprint*, *rétrospective de sprint*, *livraison des produits* et *Rétrospective de release*.

La figure 4-3 résume les étapes à suivre pour déterminer la justification commerciale.

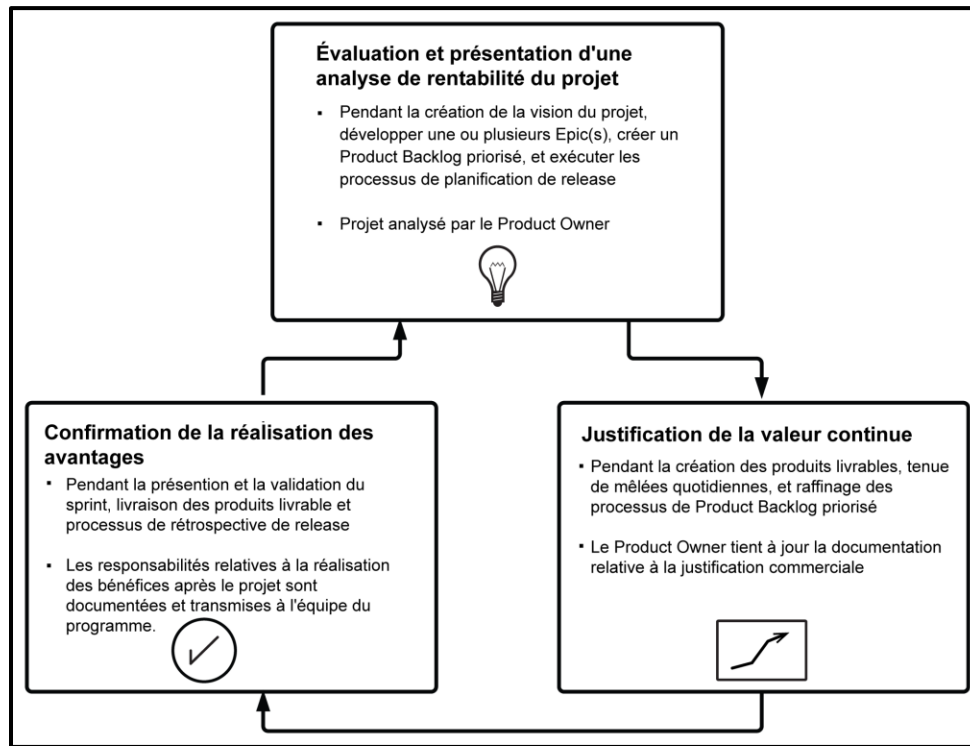


Figure 4-2: Justification commerciale et cycle du projet

4.5 Techniques de justification commerciale

Les sections suivantes présentent les outils utilisés afin d'examiner et d'évaluer la justification commerciale, ainsi que certains autres aspects associés à la justification et à la sélection d'un projet. Il n'est pas nécessaire, ni recommandé, d'utiliser toutes les techniques disponibles pour chaque projet. Certaines techniques seront appropriées à un projet spécifique, elles peuvent également être utilisées soit afin d'évaluer les projets de façon individuelle, soit afin de comparer la valeur attendue de plusieurs projets.

Le Scrum Guidance Body (SGB), qui peut être un groupe d'experts ou un ensemble de documents traitant des normes et procédures organisationnelles, définit les directives et les mesures qui seront utilisées afin d'évaluer la valeur commerciale. Cependant, chaque Product Owner est responsable de la réalisation des activités qui vérifient et effectuent le suivi de la valeur commerciale pour ses projets, programmes ou portefeuilles respectifs.

4.5.1 Estimation de la valeur du projet

La valeur qui doit être offerte par un projet commercial peut être estimée en utilisant différentes méthodes telles que le retour sur capital investi (RCI), la valeur actuelle nette (VAN) ou le taux de rendement interne (TRI).

1. Retour sur capital investi (RCI)

Le retour sur capital investi (RCI), lorsqu'il est utilisé pour justifier un projet, évalue les bénéfices nets attendus du projet. Il est calculé en déduisant les coûts ou investissements prévus pour le projet des bénéfices attendus puis en divisant le résultat (profit net) par les coûts prévus afin d'obtenir un taux de rendement. D'autres facteurs tels que l'inflation et les taux d'intérêts sur les sommes empruntées peuvent être pris en compte dans le calcul du RCI.

Formule de calcul du RCI :

$$\text{RCI} = (\text{Revenus du projet} - \text{Coûts du projet}) / \text{Coûts du projet}$$

Exemple : Pour un projet qui coûtera 125 000 € à développer, avec des bénéfices financiers estimés à 300 000 €, le RCI est calculé comme suit :

$$\text{RCI} = (300\,000\,€ - 125\,000\,€) / 125\,000 = 1,4$$

Le RCI est donc égal à 1,4 fois l'investissement (ou 140 %).

La livraison fréquente d'incrément de produit ou de service est un fondement clé de l'approche Scrum et permet de vérifier le RCI le plus tôt possible. Cela aide à évaluer la justification de la valeur continue.

2. Valeur actuelle nette (VAN)

La valeur actuelle nette (VAN) est une méthode utilisée pour déterminer la valeur actuelle nette d'un bénéfice financier à venir en tenant compte d'un taux d'inflation ou d'intérêt supposé. En d'autres mots, la VAN est la valeur totale des recettes ou bénéfices attendus pour un projet, moins le coût total attendu du projet, en prenant en compte la valeur temps de l'argent.

Exemple : Lequel des deux projets suivants est-il préférable de choisir si la VAN est utilisée comme seul critère de sélection ?

- Le projet A a une VAN de 1500 € et sera réalisé en 5 ans.
- Le projet B a une VAN de 1000 € et sera réalisé en 1 an.

Solution : Le projet A, puisque sa VAN est plus élevée. Le fait que le projet B ait une durée plus courte que le projet A n'est pas pris en considération dans ce cas car la durée est déjà prise en compte dans le calcul de la VAN (c-à-d. que c'est la valeur actuelle, et non future, qui est prise en compte dans le calcul).

3. Taux de rendement interne (TRI)

Le taux de rendement interne (TRI) est un taux de réduction sur un investissement où la valeur des flux entrants actuels est comparée à celle des flux sortants afin d'évaluer le taux de rendement d'un projet. Lorsque des projets sont comparés, un projet avec un TRI élevé est normalement meilleur.

Bien que le TRI ne soit pas utilisé pour justifier les projets aussi souvent que d'autres techniques telles que la VAN, c'est un concept important à connaître.

Exemple : En se basant sur le TRI, lequel de ces projets est le plus avantageux ?

- Le projet A, avec un TRI de 15 % et qui sera réalisé en 5 ans.
- Le projet B, avec un TRI de 10 % et qui sera réalisé en 1 an.

Solution : Le projet A, puisque son TRI est plus élevé. Le fait que le projet B ait une durée plus courte que le projet A n'est pas pris en considération dans ce cas car la durée est déjà prise en compte dans le calcul du TRI (c-à-d. que comme pour la VAN, c'est la valeur actuelle, et non future, qui est utilisée pour déterminer le TRI).

4.5.2 Planification de la valeur

Après avoir justifié et confirmé la valeur d'un projet, le Product Owner doit considérer les politiques, procédures, matrices et normes organisationnelles générales dictées par le Scrum Guidance Body (ou autre comité ou bureau de projet similaire dans une organisation) lors de la planification d'un projet, tout en maximisant la livraison de la valeur ajoutée. La responsabilité de déterminer *la façon dont* la valeur ajoutée est créée incombe aux parties prenantes de l'entreprise (sponsor, clients et/ou utilisateurs), tandis que l'équipe Scrum se concentre sur ce qui doit être développé. Certains outils courants recommandés par le Scrum Guidance Body incluent notamment :

1. La cartographie de la chaîne de valeur

La représentation de la chaîne de valeur utilise des organigrammes de processus pour illustrer le déroulement des étapes nécessaires à la réalisation d'un processus. Cette technique peut être utilisée pour rationaliser un processus en aidant à identifier et à éliminer les éléments sans valeur ajoutée et à accroître l'efficacité. La représentation de la chaîne de valeur peut être utilisée pour rationaliser les processus Scrum, par exemple, pour améliorer la vélocité des sprints.

La figure 4-4 montre comment l'identification des temps de processus et des temps d'attente peut aider à rationaliser le système en diminuant les temps d'attente et en améliorant l'efficacité des processus.

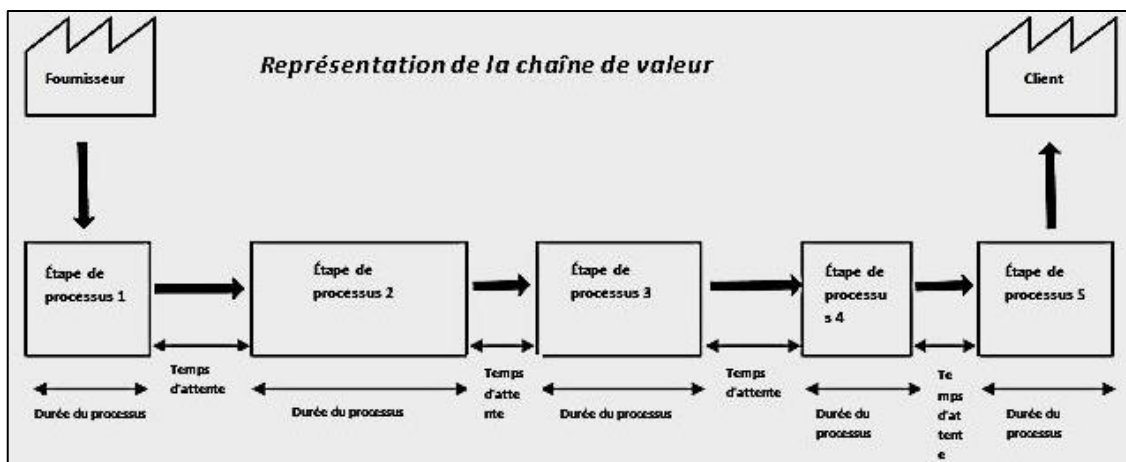


Figure 4-3: Représentation de la chaîne de valeur

2. Priorisation basée sur la valeur client

La priorisation basée sur la valeur client accorde la plus haute importance au client et cherche à réaliser les user stories ayant la plus forte valeur en premier. Ces user stories à forte valeur sont identifiées et remontées en haut du backlog de produit priorisé.

Une équipe peut utiliser différents plans de priorisation afin de déterminer les fonctionnalités à forte valeur.

a. Schémas simplifiés

Les schémas simplifiés impliquent de catégoriser les éléments en fonction de leur degré de priorité, c-à-d « 1 », « 2 », « 3 » ou « Forte », « Moyenne » et « Faible », etc. Bien que cette approche soit simple et directe, elle est souvent problématique car on observe souvent une tendance à organiser tous les éléments dans la catégorie de priorité « 1 » ou « Forte ». Même les plans de priorisation « Forte », « Moyenne » et « Faible » peuvent rencontrer des difficultés similaires.

b. Priorisation MoSCoW

La méthode de priorisation MoSCoW tient son nom des premières lettres des expressions suivantes : « Must have » (« doit être fait », vital) , « Should have » (« devrait être fait », essentiel), « Could have » (« devrait être fait, si possible », confort) et « Won't have » (« impossible », luxe). Cette méthode de priorisation est souvent plus efficace que les schémas simplifiés. Ces étiquettes sont un ordre de priorité décroissant avec « Must have » représentant les caractéristiques sans lesquelles le produit n'aura aucune valeur, et « Won't have » les caractéristiques qui, bien qu'elles seraient les bienvenues, ne sont pas nécessaires.

c. Argent de Monopoly

Cette technique implique de donner au client de « l'argent de Monopoly » ou « fausse monnaie » pour une somme égale à celle du budget du projet en lui demandant de distribuer cette somme entre les user stories considérées. De cette façon, le client priorise sur la base de ce qu'il est prêt à payer pour chaque user story.

d. Méthode des 100 points (100-Point Method)

Cette méthode a été développée par Dean Leffingwell et Don Widrig (2003). Elle consiste à donner au client 100 points qu'il peut ensuite utiliser pour voter pour les caractéristiques qui sont les plus importantes à ses yeux.

e. Analyse Kano

L'analyse Kano a été développée par Noriaki Kano (1984) et consiste en la catégorisation des caractéristiques et exigences en quatre catégories basées sur les préférences du client.

1. *Attractif/Attirant* : fonctionnalités qui sont nouvelles ou ont une forte valeur aux yeux du client
2. *Satisfaisant* : fonctionnalités qui offrent de la valeur au client
3. *Insatisfaisant* : fonctionnalités qui, si elles ne sont pas présentes, pourraient faire que le client n'apprécie pas le produit mais n'affectent pas le niveau de satisfaction si elles sont présentes
4. *Indifférent* : fonctionnalités qui n'affectent le client en aucune façon et doivent être éliminées

La figure 4-5 illustre l'analyse Kano.

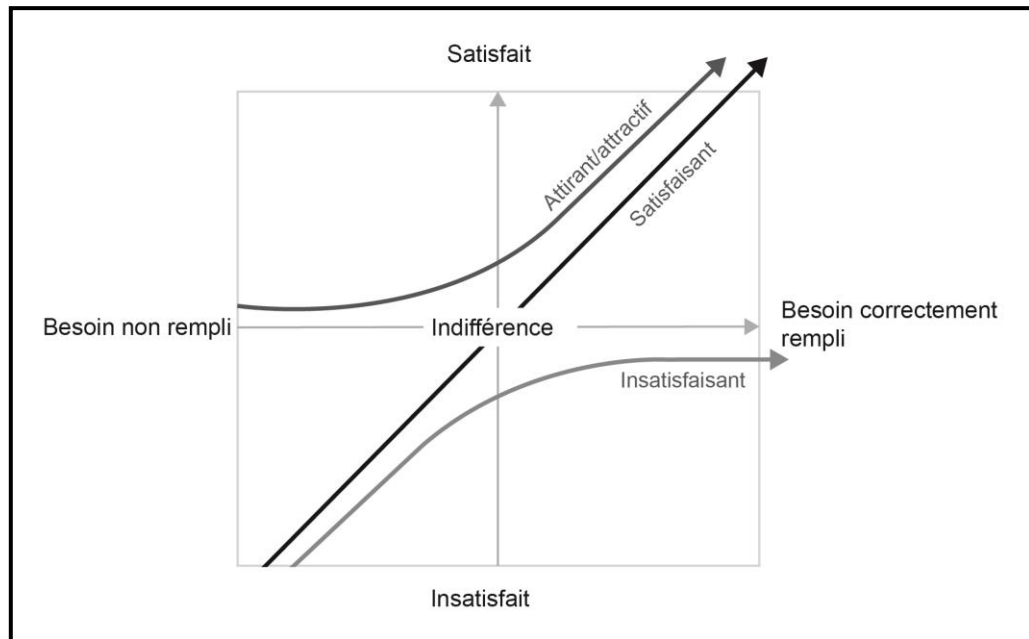


Figure 4-4: Analyse Kano

Étonnamment, le classement des fonctionnalités tend à baisser dans le temps, les clients en arriveront à attendre certaines fonctionnalités (p. ex. des appareils photo sur les téléphones) et ces fonctionnalités passeront du statut d'attrayantes celui de satisfaisantes et finalement à celui d'insatisfaisantes.

4.5.3 Classement relatif des priorités

Une simple liste des user stories en ordre de priorité est une méthode efficace pour déterminer les user stories désirées pour chaque itération ou release de produit ou de service. L'objectif est de créer une liste unique et simple dans le but de prioriser les fonctionnalités, plutôt que de se laisser distraire par plusieurs plans de priorisation.

Lorsque cela s'avère nécessaire, cette liste simple fournit également une base à l'incorporation des changements et risques identifiés. Chaque changement ou risque identifié doit être inclus à la liste en fonction de sa priorité relativement aux autres user stories de la liste. Normalement, les nouveaux changements sont inclus au détriment des fonctionnalités qui se sont vu attribuer un niveau de priorité plus faible.

Définir les MMF (sigle anglais de *minimum marketable features* c-à-d. les fonctionnalités minimales ayant de la valeur pour l'utilisateur) est extrêmement important durant ce processus. Cela permet de réaliser la première release ou itération aussi tôt que possible, conduisant à une augmentation du RCI. Normalement, ces user stories ont le niveau de priorité le plus élevé.

4.5.4 Représentation des Stories

La représentation des Stories illustre la séquence de développement des itérations de produit et établit quelles User Stories doivent être incluses dans la première, la deuxième, la troisième release et dans les releases suivantes. Lors de l'application des pratiques Scrum, il est bien entendu qu'il ne s'agit que d'une perspective actuelle et qu'elle est appelée à être révisée et modifiée fréquemment. L'exemple de représentation de Story de la figure 4-6 illustre la façon dont l'équipe Scrum planifie les différentes releases et attribue une priorité plus élevée à la release à venir. L'équipe est censée avoir une meilleure compréhension des User Stories dans la prochaine release, donc plus la release est avancée, plus les détails sont susceptibles de changer.

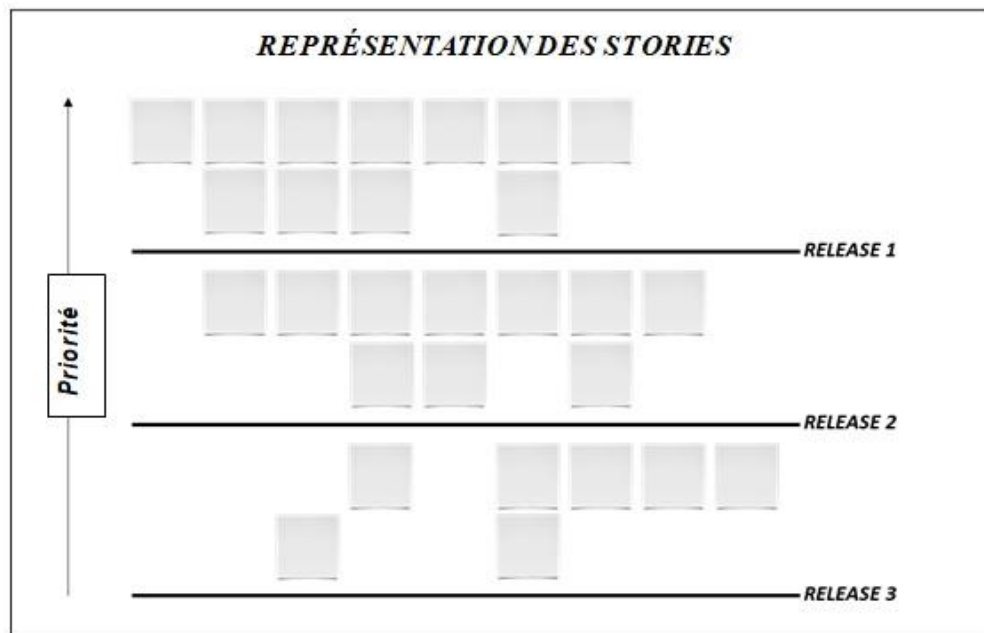


Figure 4-5: Représentation des Stories

4.6 Justification de valeur continue

La valeur commerciale doit être évaluée de façon régulière afin de déterminer si la justification ou la viabilité du projet persiste. L'évaluation fréquente des investissements sur un projet comparés à la valeur créée permet de qualifier la viabilité continue d'un projet. Les exigences attendues du projet peuvent changer fréquemment, ce qui peut à la fois affecter les investissements du projet et la création de la valeur. Un aspect clé de l'approche Scrum est sa capacité à vite s'adapter au chaos créé en changeant rapidement de modèle commercial. Dans le cas de projets ayant des exigences utilisateurs ambiguës et un fort potentiel de changements fréquents, l'approche Scrum, comparée aux autres modèles de développement, offre des avantages considérables. La surveillance du taux de livraison de la valeur est un aspect important pour les projets Scrum. Le suivi et la déclaration régulière de la valeur créée aide à évaluer le statut d'un projet et fournit des informations importantes au client ainsi qu'aux autres parties prenantes de l'entreprise.

4.6.1 Analyse de valeur acquise

Bien qu'ils soient communément utilisés, les outils tels que les histogrammes et les diagrammes de Gantt limitent le suivi et la déclaration des progrès associés à la performance du projet. L'analyse de valeur acquise (aussi désignée par son sigle anglais EVA pour *earned value analysis*) est utilisée dans ce but. L'EVA analyse la performance réelle du projet comparée à la performance prévue à un moment donné. Afin que les techniques de suivi soient efficaces, il est essentiel que le plan de base du projet soit précis. L'EVA utilise des graphiques et d'autres aides visuelles (p. ex. courbes en S), afin d'illustrer les informations relatives au statut d'un projet. L'analyse de valeur acquise mesure les variations actuelles dans le calendrier du projet, la performance des coûts et la prévision du coût final sur la base de la performance actuelle déterminée. L'EVA est généralement effectuée à la fin que chaque sprint après que les user stories du sprint backlog ont été réalisées.

Le tableau 4-1 résume les formules utilisées dans l'analyse de valeur acquise.

| Définition du terme | Acronyme | Formule |
|---|------------|---|
| Planned value (valeur planifiée) | PV | |
| Earned value (valeur acquise) | EV | |
| Actual cost (coût réel) | AC | |
| Budget at completion (budget à la clôture du projet) | BAC | |
| Schedule variance (écart entre valeur planifiée et acquise) | SV | EV - PV |
| Cost variance (écart de coûts) | CV | EV - AC |
| Schedule performance index (index de performance planning) | SPI | EV / PV |
| Cost performance index (index de performance coûts) | CPI | EV / AC |
| Percent complete (pourcentage réalisé) | % complete | EV / BAC x 100 |
| Estimate at completion (budget réévalué) 1. Hypothèses d'estimation invalides 2. Écarts actuels anormaux 3. Écarts actuels normaux | EAC | 1. AC + ETC 2. AC + BAC - EV 3. BAC / CPI |
| Estimate to complete (budget restant) | ETC | EAC - AC |
| Variance at completion (écart réévalué) | VAC | BAC - EAC |

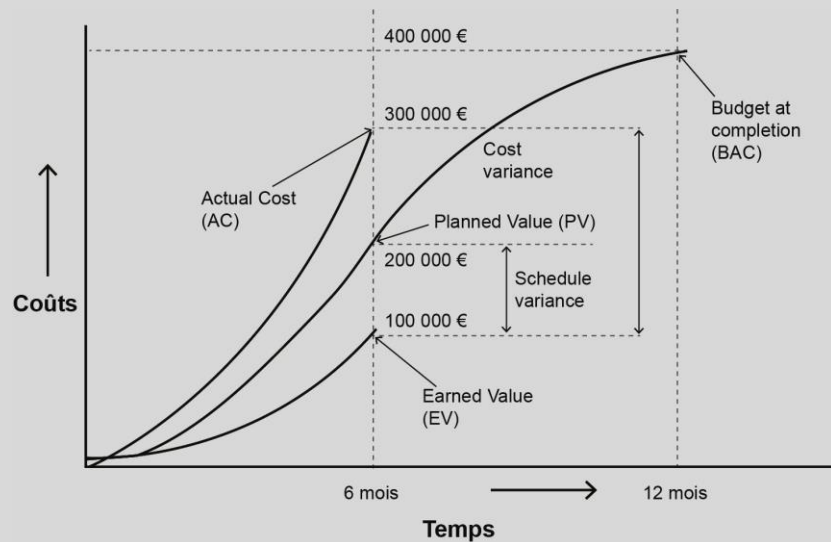
Tableau 4-2: Formules de valeur acquise

Exemple : Un site Web comptant 4000 pages Web doit être développé — nous partons du principe que la même durée est nécessaire au développement de chaque page Web et que chacune représente une user story de même niveau de priorité dans le backlog de produit priorisé. Le coût estimé pour la réalisation du projet est de 400 000 € et que le délai est de 12 mois. Après 6 mois, 300 000 € ont été dépensés et le travail réalisé représente 1000 pages Web.

De quelles données disposons-nous ?

- Budget at completion (BAC) = 400 000 € (coût de base du projet)
- Planned value (PV) = 200 000 € (puisque nous avons prévu de réaliser 2000 pages)
- Earned value (EV) = 100 000 € (valeur des 1000 réalisées)
- Actual cost (AC) = 300 000 € (ce qui a été dépensé jusqu'à présent)

Courbe en S pour les données :



Formules :

- Schedule variance (SV) = EV - PV = 100 000 € - 200 000 € = - 100 000 €
- Cost variance (CV) = EV - AV = 100 000 € - 300 000 € = - 200 000 €
 - Les écarts négatifs pour votre projet indiquent que vous êtes en retard et dépassez le budget.
- Schedule performance index (SPI) = EV / PV = 100 000 € / 200 000 € = 0,5
 - SPI < 1 indique que le travail réalisé jusqu'à présent est seulement 50 % de ce qu'il était prévu de réaliser en 6 mois.
- Cost performance index (CPI) = EV / AC = 100 000 € / 300 000 € = 0,33
 - CPI < 1 indique que pour le montant dépensé, seulement 33 % du travail a été réalisé.
- Percent complete = EV / BAC x 100 = 100 000 € / 400 000 € x 100 = 25 %
 - Donc, 25 % du travail du projet est réalisé à l'heure actuelle.

4.6.2 Diagramme de flux cumulatif (DFC)

Un diagramme de flux cumulatif (DFC) est un outil utile permettant d'effectuer le suivi des performances d'un projet. Il offre une représentation simple et visuelle des progrès du projet à un moment donné. Il est généralement utilisé pour faire état du projet global mais pas pour les mises à jour quotidiennes de chaque sprint.

La figure 4-7 est un exemple d'un DFC pour un gros projet. Il illustre combien d'user stories sont encore à réaliser, en cours de réalisation ou ont été réalisées. Au fil des changements des exigences du client, il y a un changement dans les user stories cumulatives qui doivent être livrées. Les points de changement 1 et 2 sont là où le Product Owner a supprimé des user stories dans le backlog de produit priorisé ajusté aux risques et les points de changement 3 et 4 sont là où le Product Owner a ajouté de nouvelles user stories au backlog de produit priorisé ajusté aux risques.

Ce type de diagramme peut être très utile afin d'identifier les obstacles et blocages dans les processus. Par exemple, si le diagramme affiche une bande qui devient plus étroite alors que la bande précédente devient plus large au fil du temps, il pourrait y avoir un blocage et des changements pourraient être nécessaires afin d'augmenter l'efficacité et/ou améliorer la performance du projet.

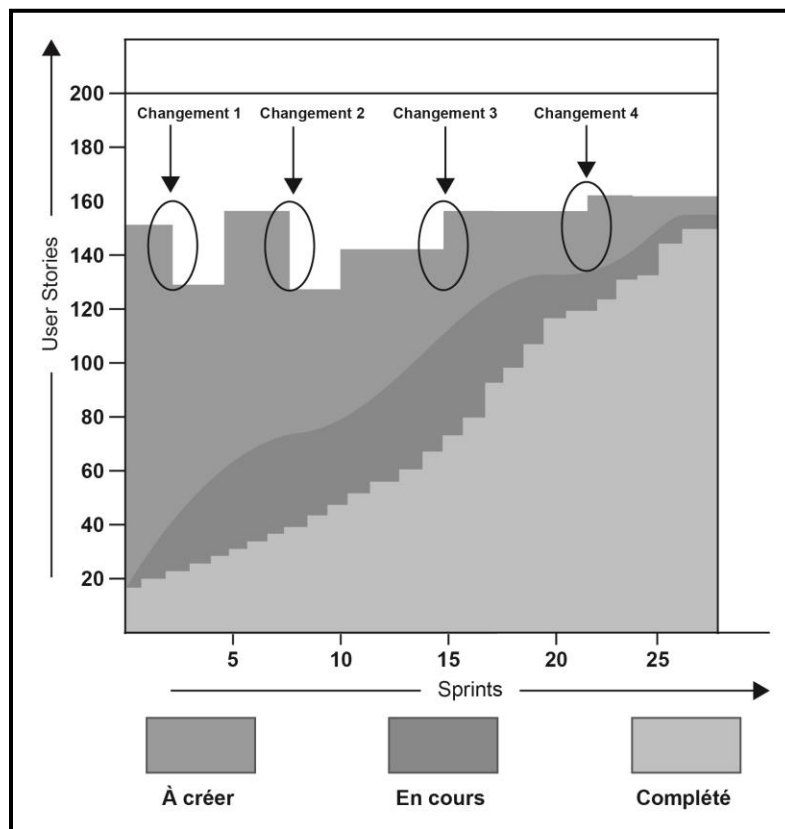


Figure 4-6: Exemple de diagramme de flux cumulatif (DFC)

4.7 Confirmer la réalisation des bénéfices

Il est important de vérifier la réalisation des bénéfices du projet tout au long de celui-ci. Que le produit du projet Scrum soit tangible ou intangible, il est essentiel d'employer des techniques de vérification appropriées afin de confirmer que l'équipe crée les incréments de produit qui réaliseront les bénéfices et la valeur définis au début du projet.

4.7.1 Prototypes, simulations et démonstrations

La démonstration des prototypes au client et la simulation de leurs fonctionnalités sont des techniques couramment utilisées afin de confirmer la valeur.

Après avoir utilisé les fonctionnalités ou en avoir fait la démonstration, les clients sont plus à même de déterminer clairement si les fonctionnalités sont adéquates ou adaptées à leurs besoins. Ils peuvent prendre conscience d'un besoin pour des fonctionnalités supplémentaires ou peuvent décider de modifier les exigences définies pour une fonctionnalité. Dans le milieu du développement de produit, cette expérience client est connue sous le nom de IKIWISI (acronyme anglais de *I'll know it when I see it*, ou « je le saurai quand je le verrai »).

Grâce aux démonstrations ou à l'accès aux premières itérations, les clients peuvent également évaluer la mesure dans laquelle l'équipe a réussi à correctement interpréter leurs exigences et à répondre à leurs attentes.

4.8 Résumé des responsabilités

| Rôle | Responsabilités |
|---------------------------------------|--|
| Équipe Scrum | <ul style="list-style-type: none"> S'assure que les incréments de produit sont réalisés conformément aux critères d'acceptation prédéterminés Effectue la justification de valeur continue pour les projets |
| Product Owner/ Chief Product Owner | <ul style="list-style-type: none"> Garantit la livraison de la valeur pour les projets Maintient la justification commerciale pour les projets Confirme et communique les bénéfices du projet aux parties prenantes de l'entreprise |
| Scrum Master/ Chief Scrum Master | <ul style="list-style-type: none"> S'assure que les résultats attendus du projet sont communiqués à et compris par l'équipe Scrum Effectue la justification de valeur continue pour les projets |

| | |
|-----------------------------------|--|
| Product Owner du programme | <ul style="list-style-type: none"> • Garantit la livraison de la valeur pour les programmes • Crée la justification commerciale pour les programmes • Offre des conseils au sujet de la valeur pour les projets du programme • Approuve la justification commerciale pour les projets d'un programme |
| Scrum Master du programme | <ul style="list-style-type: none"> • S'assure que les résultats attendus du programmes sont communiqués et compris • Effectue la justification de valeur continue pour les programmes |
| Product Owner du portefeuille | <ul style="list-style-type: none"> • Garantit la livraison de la valeur pour les portefeuilles • Crée la justification commerciale pour les portefeuilles • Offre des conseils au sujet de la valeur pour les programmes d'un portefeuille • Approuve la justification commerciale pour les programmes d'un portefeuille |
| Scrum Master du portefeuille | <ul style="list-style-type: none"> • S'assure que les résultats attendus pour le portefeuille sont réalisés • Effectue la justification de valeur continue pour les portefeuilles |
| Parties prenantes de l'entreprise | <ul style="list-style-type: none"> • Aident à prioriser les user stories et exigences dans le backlog de produit priorisé • Communiquent avec l'équipe Scrum et confirment la réalisation de la valeur à la fin de chaque sprint ou release ainsi qu'à la fin du projet |
| Scrum Guidance Body | <ul style="list-style-type: none"> • Établit les directives globales et les mesures permettant d'évaluer la valeur • Agit à titre de consultant et lorsque cela est nécessaire offre des conseils pour les projets, programmes et portefeuilles |

Tableau 4-3: Résumé des responsabilités correspondant à la justification commerciale

4.9 L'approche Scrum comparée à la gestion de projet traditionnelle

Les projets traditionnels mettent l'accent sur une planification préalable approfondie et sur le respect du plan créé par le chef de projet. Normalement les changements sont gérés via un système formel de gestion des changements et la valeur est créée à la fin du projet lorsque le produit fini est livré.

Dans les projets Scrum, la planification approfondie et à long terme n'est pas effectuée avant l'exécution du projet. La planification s'effectue de façon itérative avant chaque sprint. Cela permet de répondre rapidement et efficacement aux changements et résulte en des coûts moindres et finalement en une meilleure rentabilité et un meilleur retour sur capital investi (RCI). De plus, la livraison axée sur la valeur ajoutée (section 4.3) est un bénéfice clé de l'approche Scrum et offre un bien meilleur système de priorisation et de réalisation de la valeur commerciale. Étant donné la nature itérative du développement Scrum, il existe au moins une release du produit correspondant aux minimum marketable features (MMF). Même si un projet est annulé, des bénéfices ou de la valeur auront été créés avant l'annulation.

5. QUALITÉ

5.1 Introduction

L'objectif de ce chapitre est de définir la qualité en relation aux projets et de présenter l'approche mise en œuvre dans Scrum afin d'atteindre les niveaux de qualité requis.

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la *qualité* s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce chapitre est divisé comme suit :

5.2 Guide des rôles — cette section offre des conseils sur les sections pertinentes à chacun des rôles Scrum : Product Owner, Scrum Master et équipe Scrum.

5.3 Définir la qualité — cette section présente la définition de la qualité selon l'approche Scrum, nettement distincte de la portée, et décrit la relation qui existe entre la qualité et la valeur commerciale.

5.4 Critères d'acceptation et backlog de produit priorisé — cette section souligne l'importance des critères d'acceptation, du backlog de produit priorisé et de leur relation. Elle explique également la définition de *terminé* selon l'approche Scrum.

5.5 Gestion de la qualité dans Scrum — cette section fournit des informations sur la planification de la qualité, le contrôle qualité et l'assurance qualité dans le contexte d'un projet Scrum.

5.6 Résumé des responsabilités — cette section décrit les responsabilités correspondant à la qualité pour chaque personne ou rôle d'un projet.

5.7 L'approche Scrum comparée à la gestion de projet traditionnelle — cette section souligne les avantages de la gestion de la qualité Scrum comparée aux méthodes de gestions de projet traditionnelles.

5.2 Guide des rôles

1. Product Owner — il est important que chaque personne jouant le rôle de Product Owner lors d'un projet Scrum lise ce chapitre dans son entier.
2. Scrum Master — le Scrum Master doit également se familiariser avec ce chapitre entier en portant une attention particulière aux sections 5.3, 5.4, 5.5.3 et 5.6.
3. Équipe Scrum — l'équipe Scrum devrait surtout se concentrer sur les sections 5.3, 5.4 et 5.6.

5.3 Définir la qualité

Il existe de nombreuses façons de définir la qualité.

Dans Scrum, la qualité est définie comme la capacité à terminer un élément livrable répondant aux critères d'acceptation et produisant la valeur commerciale attendue par le client.

L'approche Scrum adopte une approche de progression continue afin de garantir qu'un projet répond aux exigences qualitatives. Les membres de l'équipe tirent les enseignements de leurs expériences et de l'implication des parties prenantes de l'entreprise afin de maintenir le backlog de produit priorisé constamment à jour en tenant compte de la modification des exigences. Le backlog de produit priorisé n'est jamais terminé jusqu'à la clôture ou l'annulation du projet. Tous les changements des exigences reflètent les changements de l'environnement commercial interne et/ou externe, ils permettent ainsi à l'équipe de travailler et de s'adapter continuellement afin de répondre à ces exigences. Puisque Scrum consiste à réaliser le travail en incréments durant les sprints, les erreurs ou défauts sont identifiés plus rapidement, grâce à la répétition des tests de qualité, plutôt que lorsque le produit ou service est presque achevé. De plus, les tâches importantes relatives à la qualité (p. ex. le développement, les tests et la documentation) sont réalisées au cours du même sprint et par la même équipe, garantissant ainsi la qualité inhérente de chaque incrément *terminé* au cours d'un sprint. Cette progression continue et les tests répétés optimisent donc la probabilité d'atteindre le niveau de qualité attendu au cours d'un projet Scrum. Les discussions constantes entre l'équipe Scrum principale et parties prenantes de l'entreprise (y compris les clients et les utilisateurs) avec de réels incréments de produits livrés à la fin de chaque sprint permettent de constamment réduire l'écart entre les attentes du client vis-à-vis du projet et les éléments livrables réalisés.

5.3.1 Qualité et portée

Les exigences relatives à la portée et à la qualité d'un projet sont déterminées en prenant en compte différents facteurs, notamment :

- le besoin commercial auquel le projet répondra
- la capacité et la volonté de l'organisation de répondre au besoin commercial identifié
- les besoins actuels et futurs du public cible

La portée d'un projet est la somme totale de tous les incréments de produit et du travail requis pour développer le produit fini. La qualité correspond à la capacité des éléments livrables à répondre aux exigences qualitatives du produit et à satisfaire les besoins du client. Dans Scrum, la portée et la qualité d'un projet sont déterminées dans le backlog de produit priorisé. La portée de chaque sprint est déterminée en affinant les larges éléments du backlog (acronyme anglais PBIs pour *product backlog items*) en un ensemble d'user stories petites mais détaillées qui peuvent être planifiées, développées et vérifiées au cours d'un sprint.

Le backlog de produit priorisé est continuellement affiné par le Product Owner. Le Product Owner s'assure que chaque user story pour laquelle l'équipe Scrum doit effectuer un sprint est affinée avant le début du sprint. Normalement, on donne la priorité aux exigences ayant le plus de valeur pour résoudre les problèmes du client ou répondre à leurs besoins tandis que les autres exigences se voient attribuer un niveau de priorité plus faible. Les user stories les moins importantes sont développées lors des sprints suivants ou entièrement laissées de côté selon les exigences du client. Tout au long du projet, le Product Owner, le client et l'équipe Scrum discutent et modifient la liste des fonctionnalités du produit pour se conformer aux besoins changeants des clients.

5.3.2 Qualité et valeur commerciale

La qualité et la valeur commerciale sont étroitement liées. Il est donc essentiel de comprendre la qualité et la portée d'un projet de façon à correctement planifier les résultats et retombées que le projet et ses produits doivent réaliser afin d'offrir une valeur commerciale au client. Afin de déterminer la valeur commerciale d'un produit il est important de comprendre le besoin commercial qui motive les exigences du produit. Ainsi, le besoin commercial détermine le produit nécessaire, et à son tour le produit définit la valeur commerciale attendue.

La qualité est une variable complexe. Augmenter la portée sans pour autant augmenter le délai ou les ressources tend à réduire le niveau de qualité. De même, la réduction du délai et des ressources sans réduire la portée, conduit généralement à une réduction du degré de qualité. L'approche Scrum estime qu'il convient de maintenir un *rythme soutenable* de travail qui permet d'améliorer la qualité au fil du temps.

Le Scrum Guidance Body peut définir les exigences qualité minimum et les normes requises de tous les projets dans l'organisation. Ces normes doivent être respectées par toutes les équipes Scrum au sein de l'entreprise.

5.4 Critères d'acceptation et backlog de produit priorisé

Le backlog de produit priorisé est un document d'exigences unique qui définit la portée du projet en offrant une liste priorisée des caractéristiques du produit ou du service à développer au cours du projet. Les fonctionnalités requises sont décrites sous la forme d'user stories. Les user stories sont des exigences spécifiques esquissées par les différentes parties prenantes de l'entreprise. Elles sont relatives au produit ou service proposé. Chaque user story possèdera des *critères d'acceptation des user stories* qui lui sont propres (aussi appelés *critères d'acceptation*) qui sont des composants objectifs utilisés pour juger la fonctionnalité d'une user story. Les critères d'acceptation sont développés par le Product Owner selon sa compréhension experte des exigences du client. Le Product Owner communique alors aux membres de l'équipe Scrum les user stories dans le backlog de produit priorisé en cherchant à obtenir leur accord. Les critères d'acceptation doivent spécifier de façon explicite les conditions que les user stories doivent satisfaire. Des critères d'acceptation clairement définis sont essentiels à la livraison efficace et dans les délais de la fonctionnalité définie par les user stories. À terme ils déterminent la réussite du projet.

À la fin de chaque sprint, le Product Owner utilise ces critères afin de vérifier les éléments livrables réalisés, et peut soit accepter soit rejeter les incréments individuels et les user stories qui leur sont associées. Si un élément livrable est accepté par le Product Owner, alors l'user story correspondante est considérée comme *terminée*. Une définition claire de *terminé* est essentielle car elle permet de clarifier les exigences et aide l'équipe à respecter les normes de qualité obligatoires. Cela permet également à l'équipe de penser du point de vue du consommateur lorsqu'elle travaille avec les user stories.

Les User Stories correspondant aux produits livrables rejetés sont ajoutés au Product Backlog priorisé afin d'être prises en compte pour être terminées dans les prochains sprints. Le rejet de quelques éléments livrables et des user stories correspondantes ne constitue pas un rejet du produit fini ou de l'incrément de produit. Le produit ou l'incrément de produit peut être potentiellement livrable même si quelques user stories sont rejetées.

La figure 5-1 illustre le concept des critères d'acceptation en parallèle au flux des incréments de produit.

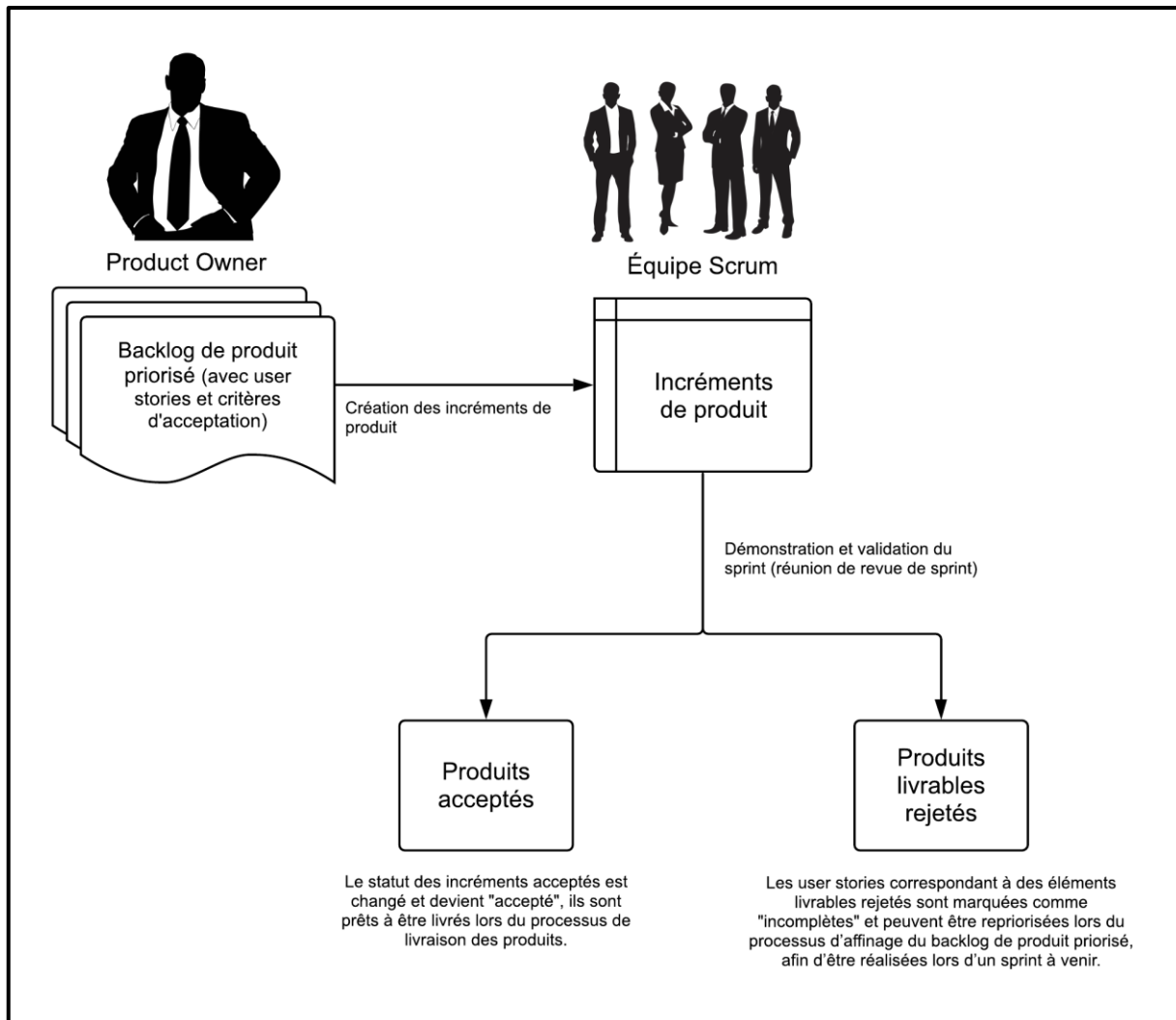


Figure 5-1: Diagramme de flux des incréments du projet

5.4.1 Écriture des critères d'acceptation

Les critères d'acceptation sont uniques à chaque user story et ne peuvent pas se substituer à une liste des exigences.

Exemple :

Personnage : Julie est une professionnelle mariée de 36 ans avec une famille de 3 enfants. C'est une femme active qui a réussi et qui arrive à équilibrer sa vie personnelle et sa vie professionnelle. Elle est à l'aise avec la technologie et est parmi les premières à adopter les produits et services les plus innovants. Elle est constamment connectée à Internet via de nombreux appareils et effectue régulièrement des achats sur des plateformes de e-commerce.

User story : « En faisant mes courses d'épicerie en ligne, je devrais être capable d'enregistrer et de consulter le brouillon de ma commande depuis n'importe lequel de mes appareils de façon à pouvoir valider ma commande à ma convenance. »

Critères d'acceptation :

- Chaque commande en cours doit être sauvegardée toutes les 5 secondes afin d'être enregistrée comme brouillon dans le compte du client
- Les nouveaux brouillons de commande doivent apparaître sous la forme de notification sur n'importe quel appareil sur lequel l'utilisateur s'identifie

Il est essentiel pour le Product Owner de comprendre que les user stories qui remplissent la majorité, mais pas tous, les critères d'acceptation ne peuvent pas être acceptées comme étant *terminées*. Les projets Scrum se déroulent sous forme de sprints à la durée limitée dans le temps, avec un backlog de sprint dédié pour chaque sprint. Souvent, la dernière partie du travail est aussi la partie la plus compliquée d'une user story et peut prendre plus de temps que prévu à réaliser. Si les user stories incomplètes sont considérées comme étant partiellement *terminées* et reportées sur le sprint suivant cela pourrait perturber les progrès du prochain sprint. Le statut *terminé* est donc tout ou rien. Une user story peut uniquement être *terminée* ou *pas terminée*.

5.4.2 Définition de « Ready » (Prêt)

La définition de « Ready » (Prêt) est un ensemble de règles ou de critères applicables à chaque User Story dans le Product Backlog priorisé. Une User Story doit satisfaire à la définition de « Ready » (Prêt) avant d'être éventuellement estimée et incluse dans un sprint. La définition de « Ready » (Prêt) met la responsabilité sur le Product Owner de définir correctement les exigences pour chaque User Story. Sans exigences correctement définies, il sera impossible d'obtenir des estimations fiables et l'équipe Scrum pourrait ne pas être en mesure de réaliser efficacement le travail requis pour le projet.

Les critères de définition de « Ready » (Prêt) doivent de préférence être établis par le Scrum Guidance Body. Cependant, il peut également y avoir des critères de définition de « Ready » (Prêt) spécifiques au projet ou à l'organisation qui devront être ajoutés ou mis à jour par le Product Owner. Il peut également y avoir des ajouts ou des mises à jour de la définition de « Ready » (Prêt) de la part de l'équipe Scrum.

L'équipe Scrum s'engage à travailler sur les User Stories qui satisfont à la définition des critères « Ready » (Prêt). L'examen des éléments du Product Backlog par rapport aux critères de définition de « Ready » (Prêt) est une activité continue qui se déroule dans le cadre du processus de *raffinage du Product Backlog priorisé*.

Voici quelques-uns des critères de définition de « Ready » (Prêt) :

- Les User Stories sont rédigées avec suffisamment de détails pour qu'elles soient comprises par l'équipe Scrum et puissent être utilisées pour l'estimation
- Les User Stories ont des critères d'acceptation bien définis
- Toute la documentation connexe qui fournit des éclaircissements sur les User Stories est incluse
- Les User Stories sont décomposées pour être suffisamment petites pour être réalisées en un seul Sprint.

5.4.3 Définition de *terminé*

Il existe une différence fondamentale entre la « définition de terminé » et les « critères d'acceptation ». Tandis que les critères d'acceptation sont uniques à chaque user story, la définition de *terminé* est un ensemble de règles qui s'applique à toutes les user stories d'un sprint donné. La définition de *terminé* générale peut notamment inclure :

- l'évaluation par les autres membres de l'équipe
- le test complet de l'user story
- la réalisation de tests d'assurance qualité
- la réalisation complète de tous les documents associés à l'user story
- la correction de tous les problèmes
- la démonstration réussie aux parties prenantes de l'entreprise et/ou autres représentants de l'entreprise

Comme pour les critères d'acceptation, toutes les conditions de la définition de *terminé* doivent être satisfaites afin de considérer l'user story comme étant *terminée*. L'équipe Scrum doit utiliser une liste des critères de la définition de *terminé* afin de garantir que la tâche est achevée et que le résultat correspond à la définition de *terminé*. La définition de « Done » (Terminé) (ou les critères « Done ») est généralement déterminée et documentée par le Scrum Guidance Body. Cependant, il peut être nécessaire d'ajouter ou de mettre à jour des critères « Done » (Terminé) spécifiques à un projet ou à une organisation. Il peut également y avoir des ajouts ou des mises à jour des critères « Done » (Terminé) par l'équipe Scrum.

Les registres et données nécessaires pour que le projet réponde aux exigences de documentation peuvent être générés pendant que l'équipe progresse dans les sprints et les releases. L'inclusion d'activités telles que les réunions de revue et l'écriture des documents de conception peut aider à garantir le respect des normes de qualité internes et externes. Les principes fondamentaux de Scrum, tels que les itérations courtes, la construction incrémentielle, l'implication du client, l'adaptation aux exigences changeantes et l'ajustement continu de la portée, des délais et des coûts pour le projet s'appliquent toujours.

5.4.4 Définition minimum de *terminé*

Une unité commerciale de plus haut niveau peut introduire des critères minimums obligatoires pour la définition de *terminé*, ceux-ci deviennent ensuite une partie de la définition de *terminé* pour toutes les user stories de cette unité commerciale. Toutes les fonctionnalités définies par l'unité commerciale doivent satisfaire à ces critères minimums de la définition de *terminé* en vue d'être acceptées par le Product Owner concerné. L'introduction de ce critère à la définition de *terminé* peut entraîner une série de critères pour la définition de *terminé* pour le portefeuille, le programme et le projet (voir figure 5-2). En conséquence, la définition de *terminé* pour une user story dans un projet inclura implicitement tous les critères minimum de la définition de *terminé* établis à un plus haut niveau, s'ils s'appliquent.

| | |
|---------------------------------------|--|
| Product Owner du portefeuille | <ul style="list-style-type: none"> • Établit les critères minimums à la définition de <i>terminé</i> pour le portefeuille dans son entier • Passe en revue les éléments livrables du portefeuille |
| Product Owner du programme | <ul style="list-style-type: none"> • Établit les critères minimum à la définition de <i>terminé</i> pour le programme dans son entier, ce qui inclut la définition de <i>terminé</i> pour le portefeuille • Passe en revue les éléments livrables du programme |
| Chief Product Owner/ Product Owner | <ul style="list-style-type: none"> • Établit les critères minimum à la définition de <i>terminé</i> pour le projet, ce qui inclut la définition de <i>terminé</i> pour le programme • Passe en revue les éléments livrables du projet |

Figure 5-2: Définition de *terminé* en cascade

Une fois que les critères minimums à la définition de *terminé* ont été définis, de tels critères peuvent être documentés dans les documents du Scrum Guidance Body pour référence par les équipes Scrum lorsque cela est nécessaire.

5.4.5 Acceptation ou rejet des éléments du backlog de produit priorisé

Vers la fin de toute itération, les unités commerciales et parties prenantes de l'entreprise concernées participent à la réunion de revue de sprint durant laquelle une démonstration de l'incrément de produit est faite pour le Product Owner, le sponsor, le client et les utilisateurs. Bien que le feedback des toutes parties prenantes de l'entreprise soit collecté, seul le Product Owner a le pouvoir d'accepter ou de rejeter une user story comme étant *terminée*, en fonction des critères d'acceptation établis. Le rôle des critères d'acceptation dans le maintien de la qualité est donc essentiel et doit être clairement compris par l'équipe. C'est la responsabilité du Scrum Master de garantir que les critères d'acceptation pour une user story ne sont pas modifiés par le Product Owner au cours d'un sprint. Les user stories partiellement réalisées sont rejetées comme étant *non terminées* et rajoutées au backlog de produit priorisé.

5.5 Gestion de la qualité dans Scrum

Le client est la partie prenante la plus importante de tout projet. Il est donc essentiel de comprendre les besoins et les exigences du client. La voix du client (VOC) fait référence aux exigences du client à la fois explicites et implicites, qui doivent être comprises avant de concevoir le produit ou le service. De manière générale, dans un environnement Scrum, le Product Owner se concentre sur les exigences et objectifs commerciaux qui, ensemble, représentent la voix du client. Le Product Owner peut grandement bénéficier des conseils offerts par le Scrum Guidance Body (soit par le biais de documents ou normes concernant la qualité, soit de la part d'experts qualité). Ces spécialistes doivent travailler avec le Product Owner et le client afin de garantir le niveau adéquat de détail et d'informations dans les user stories, étant donné que celles-ci constituent la base de la réussite de tout projet Scrum.

Il est important de noter que parties prenantes de l'entreprise externes ne sont pas directement impliquées au niveau de l'équipe Scrum, au lieu de cela, elles interagissent principalement avec le Product Owner. Pour tout projet Scrum, le client peut être :

- interne (c-à-d. qu'il provient de la même organisation),
- externe (c-à-d. qu'il n'appartient pas à l'organisation)

Dans l'approche Scrum, la gestion de la qualité permet aux clients d'être mis au courant de n'importe quel problème survenant relativement tôt dans le projet afin de les aider à décider si le projet fonctionnera ou non pour eux. Dans Scrum, la qualité garantit la satisfaction du client et la réalisation d'un produit fonctionnel, pas nécessairement le respect d'indicateurs arbitraires. Cette distinction prend beaucoup d'importance du point de vue du client car c'est lui qui investit le temps et les fonds dans le projet.

Dans l'approche Scrum, la gestion de la qualité est facilitée par trois activités complémentaires :

1. la planification de la qualité
2. le contrôle qualité
3. l'assurance Qualité

5.5.1 Planification de la qualité

Un des principes fondamentaux de Scrum est le développement en premier de la fonctionnalité ayant le plus haut niveau de priorité pour le client. Les fonctionnalités les moins importantes sont développées lors des sprints suivants ou entièrement laissées de côté selon les exigences du client. Cette approche donne à l'équipe Scrum le temps nécessaire pour se concentrer sur la qualité des fonctionnalités essentielles. Un avantage essentiel associé à la planification de la qualité est la réduction de la dette technique. La dette technique — aussi appelée dette de conception ou dette de code — fait référence au travail auquel l'équipe n'accorde pas la priorité, omet ou n'effectue pas lorsque celle-ci travaille à la création des principaux incréments relatifs au produit du projet. Une dette technique s'accumule et doit finalement être remboursée.

Certaines causes de dette technique incluent notamment :

- les solutions rapides et la conception d'éléments livrables qui ne respectent pas les normes de qualité, de sécurité, les objectifs architecturaux sur le long terme, etc.
- les tests inadaptés ou incomplets
- les documentations incorrectes ou incomplètes
- le manque de coordination entre les différents membres de l'équipe, ou si différentes équipes Scrum commencent à travailler plus isolément, en se concentrant de façon moins importante sur l'intégration des éléments requis afin de garantir la réussite du projet ou du programme
- le partage inadapté des connaissances et processus relatifs à l'entreprise entre parties prenantes de l'entreprise et les équipes du projet
- une focalisation trop importante sur les buts du projet à court terme plutôt que sur les objectifs de l'entreprise à long terme Cette négligence peut aboutir à des éléments livrables de mauvaise qualité entraînant des frais importants d'entretien et de mise à niveau.

Dans les projets Scrum, aucune dette technique n'est transférée au-delà d'un sprint car les critères d'acceptation et la définition de *terminé* devraient être clairement définis. La fonctionnalité doit respecter ces critères afin d'être considérée comme étant *terminée*. Au fur et à mesure que le backlog de produit priorisé est affiné et que les user stories sont priorisées, l'équipe crée régulièrement des éléments livrables fonctionnels afin d'éviter d'accumuler une dette technique importante. Le Scrum Guidance Body peut aussi inclure des documentations et définitions des processus afin d'aider à réduire la dette technique.

Afin de maintenir une dette technique aussi faible que possible, il est essentiel de définir le produit qui doit être créé à l'issue d'un sprint et à l'issue du projet ainsi que les critères d'acceptation, les méthodes de développement à suivre et responsabilités principales des membres de l'équipe Scrum concernant la qualité. La définition des critères d'acceptation constitue une part importante de la planification de la qualité et permet d'effectuer des contrôles qualité efficaces tout au long du projet.

La dette technique est un défi important avec les techniques de gestion de projet traditionnelles où le développement, les tests, la documentation, etc. sont réalisés de façon séquentielle et souvent par différents individus, et où personne en particulier n'est responsable d'un élément livrable fonctionnel spécifique. Par conséquent, la dette technique augmente entraînant l'augmentation des coûts de maintenance, d'intégration et de release de produit lors des dernières étapes de la release d'un projet. Les coûts associés aux changements sont également très élevés dans ces circonstances puisque des problèmes font surface lors des dernières étapes du projet. L'approche Scrum empêche l'apparition de problèmes associés à la dette technique en garantissant que les éléments livrables *terminés* et leurs critères d'acceptation sont définis dans le backlog de sprint et que les tâches fondamentales (telles que le développement, les tests et la documentation) sont réalisées lors d'un même sprint et par la même équipe Scrum.

5.5.1.1 Intégration continue et rythme soutenable

Un des principes les plus importants de Scrum est le maintien d'un rythme soutenable. Le maintien d'un rythme soutenable se traduit par une augmentation de la satisfaction des employés, de la stabilité et une meilleure précision des estimations, le tout conduisant finalement à une augmentation de la satisfaction client. Dans de telles circonstances et afin de développer un produit de très haute qualité et de maintenir un environnement de travail sain, il est important de réaliser des activités d'intégrations régulières, plutôt que de retarder le travail d'intégration. Pour fournir de la valeur à des intervalles réguliers, l'équipe doit continuellement développer, tester et intégrer les fonctionnalités de chaque éléments du backlog de produit priorisé dans chaque sprint grâce à des techniques, telles que l'intégration continue et le test de produit automatisé. Il est également important, du point de vue de l'équipe, de s'assurer que les efforts investis lors du sprint en cours sont similaires aux efforts investis lors des sprint précédents de manière à maintenir un rythme régulier tout au long des sprints du projet. Cela aide l'équipe à éviter les phases de travail intenses, garantissant qu'elle est toujours capable de fournir les efforts requis afin d'accomplir le travail qui doit être effectué. Maintenir un rythme durable est l'un des principes les plus importants de Scrum et d'autres pratiques agiles telles que DevOps.

5.5.2 Assurance qualité et contrôle de la qualité

Le contrôle qualité fait référence à l'exécution des activités de qualité prévues par l'équipe Scrum au cours du processus de création de produits potentiellement livrables. Le contrôle qualité implique également de tirer les enseignements provenant de chaque ensemble d'activités achevées afin de garantir la progression continue. Il est important de disposer des compétences nécessaires au sein de l'équipe polyvalente afin d'effectuer les activités de contrôle qualité. Les membres de l'équipe discutent des enseignements tirés durant la réunion de Rétrospective de release. Ces enseignements contribuent à la progression continue ainsi qu'à la progression du contrôle qualité. La qualité est requise non seulement pour les produits mais aussi pour les processus. L'assurance qualité réfère au processus d'évaluation et aux normes qui gouvernent la gestion de la qualité d'un projet afin de s'assurer qu'ils demeurent adéquats. Les activités d'assurance qualité sont effectuées avec le reste du travail. En fait, l'assurance qualité est un facteur de grande importance dans la définition de terminé. L'élément livrable n'est pas complet si l'assurance qualité adéquate n'a pas été effectuée. Souvent, l'assurance qualité est démontrée durant la réunion de revue de sprint.

Les Product Owners des projets, programmes et portefeuilles respectifs peuvent surveiller et évaluer les activités d'assurance qualité afin de garantir que chaque équipe continue à accepter et respecter les normes de qualité qui ont été établies. Le processus intégral d'assurance qualité peut être abordé durant la phase de test finale du produit, de la release ou du sprint. Une comparaison du nombre de problèmes rencontrés contre le nombre d'user stories réalisées peut être effectuée. Les composants de produit défectueux peuvent être intégrés aux éléments du backlog de produit priorisé et peuvent être retravaillés soit par l'équipe soit par une personne à certains moments durant le sprint, en fonction du nombre de défauts.

À certains moments, le Scrum Guidance Body peut définir les processus et documents consultables par les équipes Scrum lorsqu'elles réalisent leurs projets afin de garantir que les normes de qualité uniformes sont respectées par tous les projets au sein de l'entreprise.

5.5.3 Cycle Plan-Do-Check-Act (PDCA)

Le cycle Plan-Do-Check-Act (Préparer-Réaliser-Vérifier-Agir) ou roue de Deming ou de Shewhart a été développé par le Dr. W. Edwards Deming (considéré comme le père du contrôle qualité moderne) et le Dr. Walter A. Shewhart. Les points suivants sont essentiels à la philosophie de Deming :

- Les directives de gestion définissent la qualité. Lorsque la direction est capable de fournir un environnement propice et de motiver ses employés à améliorer la qualité de façon continue, chaque employé sera capable de contribuer à une qualité de produit supérieure. La « théorie de connaissance approfondie » de Deming recommande ce qui devrait être fait par la direction afin de créer un environnement au sein duquel chaque employé peut contribuer de façon importante à l'amélioration de la qualité.

Deming a modifié Plan-Do-Check-Act en Plan-Do-Study-Act (Préparer-Réaliser-Étudier-Agir ou PDSA) car le terme « Study » (étudier) souligne l'analyse alors que « Check » (vérifier) privilégiait simplement l'inspection. L'approche Scrum et la roue de Deming/Shewhart (ou cycle PDCA) sont deux méthodes itératives basées sur la progression continue. La figure 5-3 illustre les étapes du cycle PDCA et leur corrélation avec les différents processus Scrum.

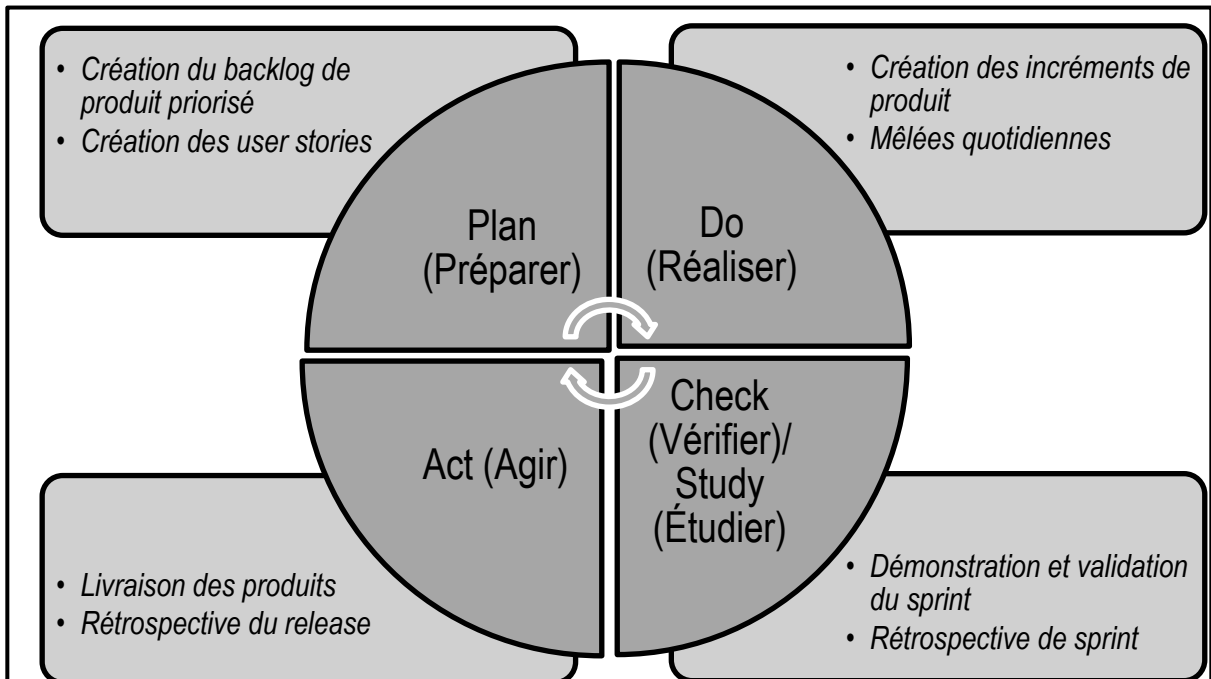


Figure 5-3: Définition de terminé en cascade

5.6 Résumé des responsabilités

| Rôle | Responsabilités |
|---------------------------------------|---|
| Équipe Scrum | <ul style="list-style-type: none"> • Développe et maintient tous les éléments livrables durant les sprints jusqu'à ce qu'ils soient transférés aux utilisateurs finaux • Pratique et encourage une communication efficace afin de clarifier et de comprendre pleinement les exigences • Partage le savoir afin de garantir que les membres de l'équipe se familiarisent avec l'ensemble des fonctionnalités et bénéficient ainsi de l'expérience des autres • Effectue rapidement les changements nécessaires aux éléments livrables • Veiller à la conformité avec les critères de la définition de « Done » (Terminé) pour chaque produit livrable |
| Product Owner/ Chief Product Owner | <ul style="list-style-type: none"> • Établit les exigences commerciales pour le produit et définit clairement les exigences dans le backlog de produit priorisé • Évalue la viabilité et assure que les éléments livrables répondent aux exigences en termes de qualité • Établit les critères minimum à la définition de <i>terminé</i> pour le projet dans son entier, ce qui inclut les critères d'acceptation pour le programme en question • Facilite la création des critères d'acceptation pour les user stories • Évalue et valide les éléments livrables lors du processus de <i>démonstration et validation du Sprint</i> |
| Scrum Master/ Chief Scrum Master | <ul style="list-style-type: none"> • Encourage la mentalité selon laquelle l'équipe passe en premier lorsqu'il est question de qualité • Élimine les obstacles environnementaux qui pourraient affecter la qualité des éléments livrables et des processus • S'assure qu'un rythme soutenable est maintenu en se concentrant sur la qualité des fonctionnalités plutôt que sur la vitesse même. • S'assure que les processus Scrum sont respectés par tous les membres de l'équipe y compris le Product Owner |
| Product Owner du programme | <ul style="list-style-type: none"> • Établit les critères minimums à la définition de <i>terminé</i> pour le programme dans son entier • Passe en revue les éléments livrables du programme |
| Scrum Master du programme | <ul style="list-style-type: none"> • S'assure qu'un rythme soutenable est maintenu en se concentrant sur la qualité des fonctionnalités plutôt que sur la vitesse même. |
| Product Owner du portefeuille | <ul style="list-style-type: none"> • Établit les critères minimums à la définition de <i>terminé</i> pour le portefeuille dans son entier • Passe en revue les éléments livrables du portefeuille |
| Scrum Master du portefeuille | <ul style="list-style-type: none"> • S'assure qu'un rythme soutenable est maintenu en se concentrant sur la qualité des fonctionnalités plutôt que sur la vitesse même. |
| Parties prenantes | <ul style="list-style-type: none"> • Examiner et fournir des commentaires sur les produits livrables |

| | |
|---------------------|---|
| de l'entreprise | <ul style="list-style-type: none"> • Travailler en collaboration avec le Product Owner et l'équipe Scrum |
| Scrum Guidance Body | <ul style="list-style-type: none"> • Fournit la définition de <i>terminé</i> • Fournit un cadre et des directives afin de définir les critères d'acceptation • Définit l'éventail d'outils pouvant être utilisés par l'équipe Scrum afin de développer et vérifier le produit. |

Tableau 5-1: Résumé des responsabilités correspondant à la qualité

5.7 L'approche Scrum comparée à la gestion de projet traditionnelle

Il existe des similitudes entre les méthodes Scrum et la gestion de projet traditionnelle concernant la définition de la « qualité » (c-à-d. la capacité d'un produit à satisfaire les critères d'acceptation et à réaliser la valeur commerciale attendue par le client). Cependant, des différences existent dans la manière dont ces approches mettent en œuvre et réalisent les niveaux de qualité exigés.

Avec les méthodes de gestion de projet traditionnelles, les utilisateurs clarifient leurs attentes, le chef de projet définit ces attentes en termes tangibles et obtient l'accord des utilisateurs. Après une planification détaillée, l'équipe du projet développe le produit sur un délai prédéfini. Si n'importe lequel des critères acceptés doit être modifié, les changements sont mis en œuvre par le biais d'un système formel de gestion des changements ou l'impact des changements est estimé et le chef de projet obtient l'accord de toutes parties prenantes de l'entreprise concernées.

Dans l'approche Scrum, cependant, le Product Owner collabore avec l'équipe Scrum et définit les critères d'acceptation pour les user stories relatives au produit à livrer. L'équipe Scrum développe alors le produit en une série d'itérations courtes appelées sprints. Le Product Owner peut effectuer des changements aux exigences afin de rester en phase avec les besoins des utilisateurs et ces changements peuvent être résolus par l'équipe Scrum soit en annulant le sprint en cours soit en incluant les exigences modifiées au prochain sprint puisque chaque sprint est d'une durée très courte (c-à-d. une à quatre semaines).

Un des avantages les plus importants de l'approche Scrum est l'accent mis sur la création d'éléments potentiellement livrables à la fin de chaque cycle de sprint plutôt qu'à la fin du projet dans son entier. Le Product Owner et les clients inspectent et approuvent donc continuellement les éléments livrables à la fin de chaque sprint. Aussi, même si un projet Scrum est annulé de façon prématurée, de la valeur aura été créée avant l'annulation au travers des éléments livrables créés durant chaque sprint.

6. CHANGEMENT

6.1 Introduction

Chaque projet, quelle que soit la méthode ou l'approche utilisée, est soumis au changement. Les membres de l'équipe du projet doivent impérativement comprendre que les processus de développement Scrum sont conçus afin d'être ouverts au changement. En appliquant les principes Scrum pour les processus de gestion des changements, les organisations essaient de maximiser les avantages qui proviennent des changements et de minimiser les impacts négatifs.

Tel qu'il est décrit dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, le *changement* s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce chapitre est divisé comme suit :

6.2 Guide des rôles — cette section offre des conseils sur les sections pertinentes à chacun des rôles principaux Scrum : Product Owner, Scrum Master et équipe Scrum.

6.3 Aperçu — cette section définit le concept de changement, plus particulièrement dans le contexte des processus Scrum. Elle traite également de la manière dont les demandes de changement sont gérées dans les processus Scrum.

6.4 Le changement dans Scrum — cette section traite de l'importance de gérer efficacement le changement lors d'un projet Scrum. Elle aborde également la façon dont la flexibilité et la stabilité peuvent être atteintes grâce à la gestion appropriée des demandes de changement qui émergent tout au long d'un projet.

6.5 Intégration des changements — cette section traite de la manière dont les demandes de changement sont approuvées (ou rejetées) lors de l'application de l'approche Scrum.

6.6 Changements apportés aux programmes et portefeuilles — cette section décrit l'impact des changements sur les programmes et les portefeuilles.

6.7 Résumé des responsabilités — cette section définit les responsabilités des membres de l'équipe du projet concernant la gestion des changements.

6.8 L'approche Scrum comparée à la gestion de projet traditionnelle — cette section traite des avantages associés à la gestion des changements selon les méthodes Scrum comparées aux méthodes de gestions de projet traditionnelles.

6.2 Guide des rôles

4. Product Owner — la responsabilité d'initier les changements au cours d'un projet tient principalement au Product Owner, ce chapitre dans son entier s'applique donc à ce rôle.
5. Scrum Master — le Scrum Master doit également se familiariser avec ce chapitre entier en portant une attention particulière aux sections 6.3, 6.4, 6.5 et 6.7.
6. Équipe Scrum — L'équipe Scrum devrait surtout se concentrer sur les sections 6.3, 6.4.2, 6.5 et 6.7.

6.3 Aperçu

Le changement est inévitable dans tous les projets. Dans un monde ultra-compétitif où la technologie, les conditions de marché et les modèles commerciaux sont en perpétuelle mutation, le changement est la seule constante.

Un principe fondamental de l'approche Scrum est le constat a) que parties prenantes de l'entreprise (p. ex. clients, utilisateurs et sponsors) changent d'avis concernant de ce qu'ils veulent et ce dont ils ont besoin au cours d'un projet (parfois appelé « modification des exigences ») et b) qu'il est très difficile, voire impossible, pour parties prenantes de l'entreprise de définir toutes les exigences lors de la phase d'initiation d'un projet.

Les projets de développement Scrum accueillent le changement en utilisant des cycles de développement courts qui intègrent le feedback du client sur les éléments livrables du projet à la fin de chaque sprint. Cela permet aux clients d'interagir régulièrement avec les membres de l'équipe Scrum, de voir les incréments de produit lorsque ceux-ci sont prêts et de modifier les exigences plus tôt durant le processus développement. Les équipes de gestion du portefeuille ou du programme peuvent également répondre aux demandes de changement applicables à leur niveau et ayant trait aux projets Scrum.

Scrum incarne un principe fondamental du Manifeste Agile (Fowler et Highsmith, 2001) : « L'adaptation au changement plus que le suivi d'un plan. » L'approche Scrum se pratique en partant du principe qu'il faut être ouvert au changement et le transformer en avantage compétitif. Il est donc plus important d'être flexible que de suivre un plan strict et prédéfini. Cela signifie qu'il est essentiel d'aborder la gestion de projet d'une manière adaptative qui favorise le changement au travers de cycles rapides de développement de produit ou de développement de service.

L'adaptation aux changements est un avantage fondamental de l'approche Scrum. Même si Scrum fonctionne bien pour tout type de projet et tout secteur d'activité, cette approche est très efficace lorsque les exigences relatives au produit ou à d'autres aspects du projet ne sont pas complètement comprises ou ne peuvent pas être complètement définies dès le début du projet, lorsque le marché du produit est volatile, et/ou lorsque l'accent est mis sur la flexibilité de l'équipe afin d'intégrer des exigences changeantes. L'approche Scrum est particulièrement utile pour les projets complexes avec un haut niveau d'incertitude. La planification et les prévisions à long terme sont généralement inefficaces pour de tels projets et impliquent un haut niveau de risque. Scrum permet de guider l'équipe vers les résultats commerciaux ayant une plus forte valeur grâce à la *transparence*, à l'*inspection* et à l'*adaptation*.

6.3.1 Demandes de changement rejetées et approuvées

Les requêtes de modification sont normalement présentées sous la forme de demandes de changement. Les demandes de changement restent non-ratifiées jusqu'à ce qu'elles soient officiellement approuvées. Normalement, le Scrum Guidance Body définit un processus afin d'approuver et de gérer les changements au sein de l'organisation. En l'absence d'un processus formel, il est recommandé que les petits changements qui n'ont pas d'impact significatif sur le projet soient directement approuvés par le Product Owner. Le degré de tolérance pour de tels changements peut être défini au niveau organisationnel ou par le sponsor d'un projet en particulier. Dans la plupart des projets, 90% des demandes de changement peuvent être qualifiées de petits changements et peuvent être approuvés par le Product Owner. Les changements qui dépassent le niveau d'approbation du Product Owner peuvent nécessiter l'approbation des parties prenantes de l'entreprise concernée en collaboration avec le Product Owner.

Les changements qui se trouvent au-delà du degré de tolérance du Product Owner peuvent nécessiter d'être approuvés par parties prenantes de l'entreprise concernées en collaboration avec le Product Owner.

À certains moments, si un changement demandé peut avoir un impact important sur le projet ou l'organisation, l'approbation des cadres supérieurs (p.ex. sponsor exécutif, Product Owner du portefeuille, Product Owner du programme ou Chief Product Owner) pourra être requise.

Les demandes de changement pour le projet sont abordées et approuvées durant les processus de *développement des epics*, de *création du backlog de produit priorisé* et de *Raffinage du Product Backlog priorisé*. Les *demandes de changement approuvées* sont ensuite priorisées avec les autres exigences du produit et leurs user stories respectives puis incorporées au backlog de produit priorisé.

La figure 6-1 résume le processus d'approbation des changements et la figure 6-2 illustre la manière dont le backlog de produit priorisé est mis à jour avec les changements approuvés.

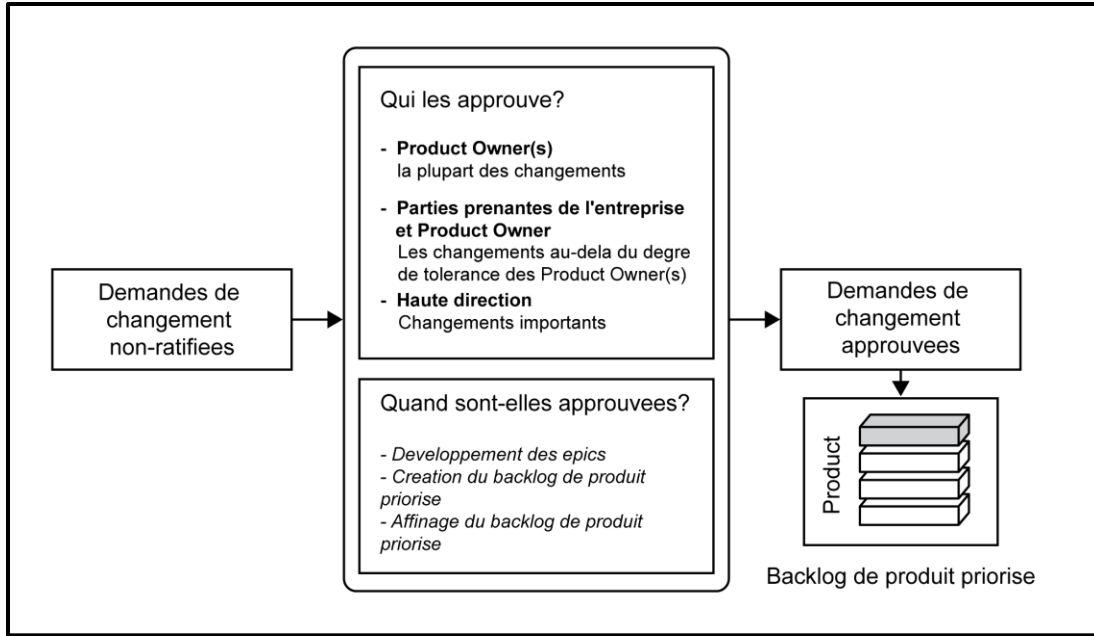


Figure 6-1: Exemple de processus d’approbation des changements

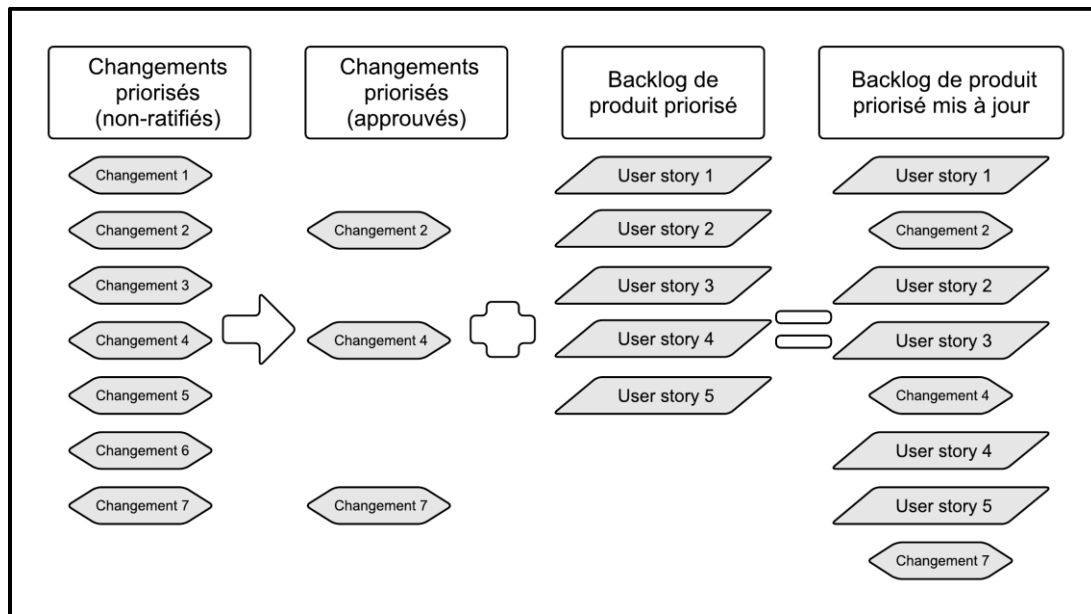


Figure 6-2: Mise à jour du backlog de produit priorisé avec les changements approuvés

6.4 Le changement dans Scrum

6.4.1 Équilibrer flexibilité et stabilité

Scrum aide les organisations à devenir plus flexibles et ouvertes au changement. Cependant, il est important de comprendre que même si l'approche Scrum encourage la flexibilité, il est également important de maintenir la stabilité tout au long du processus de changement. De la même façon que la rigidité extrême est inefficace, la flexibilité extrême est également improductive. La clé est de trouver le bon équilibre entre flexibilité et stabilité car la stabilité est nécessaire à la réalisation du travail. Scrum utilise donc le développement itératif et ses autres caractéristiques et principes afin d'atteindre le bon équilibre. Scrum maintient la flexibilité dans le sens où les demandes de changement peuvent être créées et approuvées à n'importe quel moment au cours du projet, cependant elles sont priorisées lors de la création ou de la mise à jour du backlog de produit priorisé. En même temps, Scrum garantit la stabilité en maintenant le backlog de sprint et en empêchant les interférences avec l'équipe Scrum au cours d'un sprint.

Dans Scrum, toutes les exigences associées à un sprint en cours sont gelées durant le sprint. Aucun changement n'est introduit avant la fin du sprint, sauf si un changement est suffisamment important pour justifier l'annulation du sprint. Dans le cas d'un changement urgent, le sprint est annulé et l'équipe se regroupe pour planifier un nouveau sprint. C'est de cette manière que Scrum accepte les changements sans créer le problème d'avoir à modifier les dates de release.

6.4.2 Atteindre la flexibilité

En raison de sa nature itérative et des concepts de contrôle empirique des processus que sont la transparence, l'inspection et l'adaptation, l'implémentation de Scrum doit intégrer le concept de flexibilité. Scrum fournit des mécanismes d'adaptation pour la gestion de projet dans lesquels le changement des exigences peut être géré sans impacter la progression globale du projet de façon trop importante. Il est nécessaire de s'adapter aux réalités commerciales qui surviennent durant le cycle de développement. Dans Scrum, la flexibilité est atteinte au travers de cinq caractéristiques clés (voir figure 6-3) : le développement de produit itératif, le time-boxing, les équipes polyvalentes, la priorisation basée sur la valeur client et l'intégration continue.

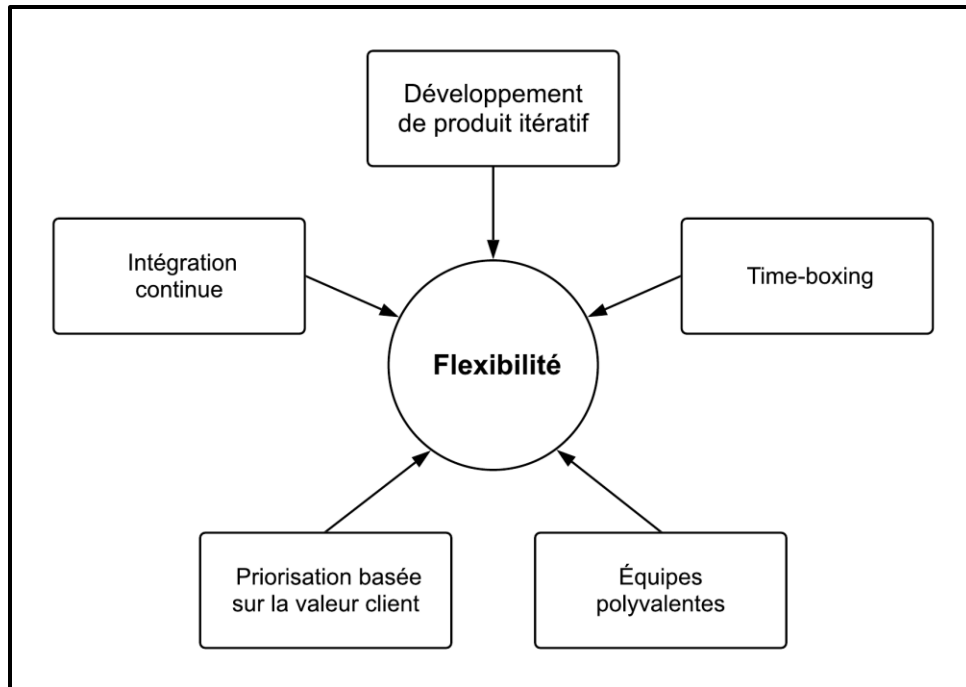


Figure 6-3: Caractéristiques Scrum pour atteindre la flexibilité

6.4.2.1 La flexibilité via le développement de produit en itération

Scrum suit une approche de développement de produit ou de service itérative et incrémentielle qui permet d'intégrer les changements à n'importe quelle étape du processus de développement. Les demandes de changement pour le projet peuvent provenir de différentes sources durant le développement du produit :

1. Parties prenantes de l'entreprise

Parties prenantes de l'entreprise du projet (en particulier les sponsors, clients et utilisateurs) peuvent soumettre des demandes de changement à n'importe quel moment tout au long du projet. Les demandes de changement peuvent être dues à des changements dans les conditions de marché, dans la direction de l'organisation, à des problèmes légaux ou en termes de réglementation, ou à d'autres raisons. De plus, parties prenantes de l'entreprise peuvent effectuer des demandes de changement lorsqu'elles passent en revue les éléments livrables lors des processus de *démonstration et validation du sprint*, de *rétrospective de sprint* ou de *Rétrospective de release*. Une fois approuvées, toutes les demandes de changement sont ajoutées au backlog de produit priorisé du projet (aussi appelé backlog de produit priorisé ou backlog de produit).

La figure 6-4 illustre certaines des raisons pour lesquelles les parties prenantes de l'entreprise lancent le processus de demande de changement.

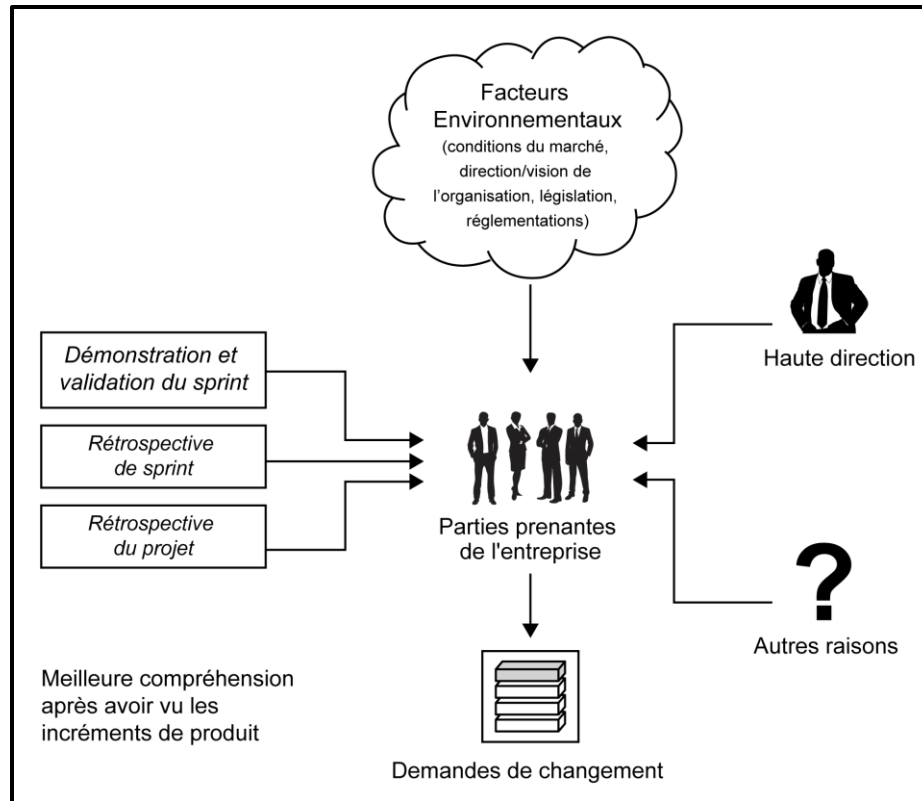


Figure 6-4: Motivations des parties prenantes de l'entreprise pour demander des changements

2. Équipe Scrum principale

L'équipe Scrum principale (c-à-d. le Product Owner, le Scrum Master et l'équipe Scrum) est impliquée dans la création des éléments livrables du produit. Les interactions continues entre les membres de l'équipe Scrum principale d'une équipe Scrum et d'autres individus, tels que les autres équipes Scrum du projet et parties prenantes de l'entreprise internes et externes impliquées dans le projet, peuvent conduire les membres de l'équipe Scrum principale à suggérer des changements ou améliorations pour le produit, service ou pour une autre partie du projet. Normalement, de tels changements (comme tous les autres) sont enregistrés sous la forme de demandes de changement. Le Product Owner prend la décision finale concernant les changements suggérés par l'équipe Scrum et le Scrum Master à prendre en considération sous la forme de demandes de changement formelles. À certains moments, il peut y avoir des défis à la création de certains éléments livrables qui peuvent résulter en la création de demandes de changement. Par exemple, l'équipe pourrait décider de modifier une fonctionnalité ou d'en ajouter une nouvelle afin d'améliorer la performance d'un produit. Dans la plupart des projets Scrum, les recommandations pour des changements de la part de l'équipe Scrum principale sont émises lorsque l'équipe Scrum travaille à la *création des éléments livrables*, ou lorsqu'elle participe aux *mêlées quotidiennes* ou à la *réunion de rétrospective du sprint*. La figure 6-5 illustre certaines des raisons pour lesquelles l'équipe Scrum principale peut initier des demandes de changement.

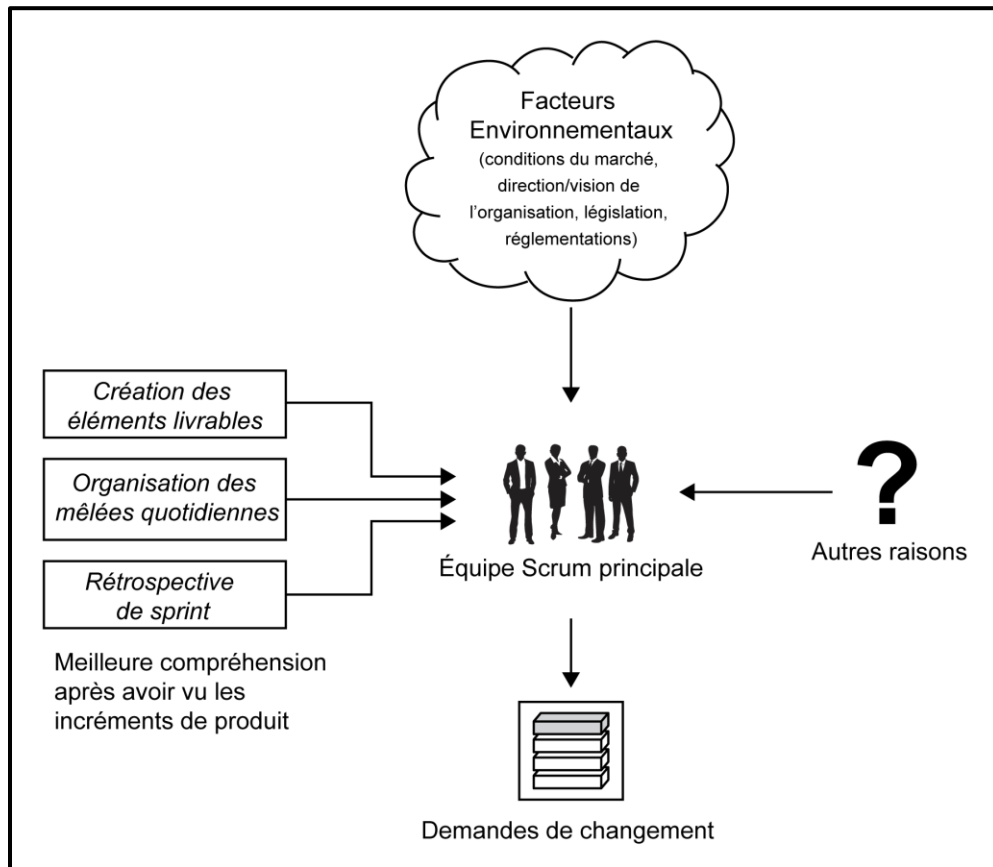


Figure 6-5: Motivations de l'équipe Scrum principale pour effectuer des demandes de changement

3. Haute direction

La haute direction (y compris les gestionnaires des portefeuilles et programmes) peut recommander des changements qui affectent le projet. Ces demandes peuvent être dues à des changements dans la direction stratégique de l'entreprise, dans l'environnement concurrentiel, à des problèmes de financement, et cætera. De tels changements sont ajoutés au backlog de produit priorisé et doivent suivre le processus habituel de gestion des changements. Si un de ces changements est urgent, tout sprint impacté devrait être annulé (voir section 6-6 pour plus d'informations).

4. Scrum Guidance Body

Le Scrum Guidance Body peut effectuer des demandes de changement qui affectent tous les projets dans n'importe lequel des cas suivants :

- changement dans les règlements gouvernementaux (p. ex. confidentialité, normes de sécurité ou nouvelle législation)

- directives propres à l'entreprise concernant la qualité, la sécurité ou tout autre initiative organisationnelle devant être mise en œuvre à l'échelle de l'entreprise
- références et meilleures pratiques afin de satisfaire à certaines normes
- enseignements tirés des projets précédents qui pourraient être mis en œuvre par d'autres équipes Scrum

La marque distinctive de l'approche Scrum est qu'elle est tolérante aux changements et adaptative. L'approche Scrum n'encourage pas la détermination ni l'établissement de plans fixes longtemps à l'avance car elle fonctionne sur le principe que le développement d'un projet est sujet à des changements et à des risques importants. Cela résulte en un degré élevé de flexibilité et de tolérance aux changements. Le projet est planifié, exécuté et géré de façon incrémentielle, il est donc normalement facile d'intégrer les changements tout au long de celui-ci.

6.4.2.2 La flexibilité à travers le time-boxing

Le time-boxing fait référence à la mise en place de courtes durées de temps pour effectuer le travail. Si le travail effectué reste incomplet à la fin de la time-box, il est transféré à la time-box suivante. Des exemples de time-boxing incluent de limiter les mêlées quotidiennes à 15 minutes et d'établir la durée des sprints entre une et quatre semaines. Les time-boxes offrent la structure nécessaire aux projets Scrum qui possèdent un élément d'incertitude, sont de nature dynamiques et sujets à des changements fréquents. Les erreurs ou les problèmes peuvent être identifiés à un stade précoce et corrigés rapidement, ce qui permet une certaine flexibilité dans les projets Scrum. Les time-boxes aident à estimer les progrès du projet et permettent à l'équipe d'identifier facilement les cas où elle a besoin de modifier un processus ou une approche.

Les sprints limités dans le temps contribuent grandement au respect des échéances et à l'atteinte d'un haut niveau de productivité. Les sprints favorisent l'ordre et la régularité dans un environnement de travail instable. Ils offrent une plateforme permettant d'évaluer les résultats et d'obtenir du feedback dans un court laps de temps. Les sprints permettent également l'évaluation fréquente des progrès et des méthodes utilisées pour gérer le projet, notamment la gestion efficace des changements. Les erreurs ou problèmes peuvent être identifiés très tôt et rectifiés rapidement.

En utilisant le time-boxing pour les sprints, l'équipe revisite fréquemment le processus d'estimation du travail à effectuer, la projection en termes de temps et d'efforts requis devient donc plus précise au fil de chaque sprint et des progrès du projet. Ces cycles itératifs motivent également les membres de l'équipe à atteindre les objectifs projetés et les buts incrémentiels afin de réaliser l'objectif global du projet.

6.4.2.3 La flexibilité au travers des équipes polyvalentes et autoorganisées

L'auto-organisation garantit que les membres de l'équipe Scrum disposent de la souplesse nécessaire pour déterminer toutes les tâches sur lesquelles ils travailleront au cours d'un sprint et la manière dont ils accompliront le travail. Elle permet également aux équipes de rester motivées pour mener à bien les tâches qui leur sont assignées, de supprimer les goulets d'étranglement et d'encourager le partage des connaissances avec les autres membres de l'équipe. La structure polyvalente et autoorganisée de l'équipe Scrum permet aux membres de l'équipe d'être extrêmement concentrés sur les résultats désirés du sprint. L'équipe dispose d'une série d'objectifs définis pour chaque sprint et de la flexibilité nécessaire pour justifier un changement dans les objectifs avant le début du prochain sprint.

L'utilisation d'équipes polyvalentes garantit également que toutes les compétences et connaissances requises afin d'effectuer le travail du projet existent au sein de l'équipe en elle-même. Cela offre un modèle de travail efficace qui résulte en la création d'éléments potentiellement livrables et prêts à la démonstration pour le Product Owner et/ou les autres parties prenantes de l'entreprise .

L'auto-organisation garantit que les membres de l'équipe Scrum déterminent eux-mêmes *la manière* dont ils effectuent le travail du projet sans responsable hiérarchique pour micro-gérer leurs tâches.

Avoir des équipes polyvalentes et autoorganisées permet au groupe d'adapter et de gérer efficacement le travail continu et les problèmes ou changements mineurs sans avoir à obtenir le soutien ou l'expertise de membres extérieurs à l'équipe, et dans le même temps de créer des éléments prêts à être livrés si nécessaire.

6.4.2.4 La flexibilité au travers de la priorisation basée sur la valeur client

La priorisation des exigences et du travail dans un projet Scrum est toujours déterminée en fonction de la valeur offerte au client. Tout d'abord, au début du projet, les exigences initiales sont priorisées en fonction de la valeur que chacune fournira et documentées dans le backlog de produit priorisé. Puis, lorsqu'une demande est faite pour une nouvelle exigence ou un changement à une exigence existante, celle-ci est évaluée lors du processus d'*Raffinage du Product Backlog priorisé*. S'il est jugé que le changement offre plus de valeur qu'une autre exigence existante, celui-ci est alors ajouté et priorisé en conséquence dans le backlog de produit priorisé mis à jour. Le backlog de produit priorisé offre donc un cadre à l'intégration des changements et à l'addition de nouvelles exigences lorsque cela est nécessaire.

Il est important de noter que les nouvelles exigences et les changements ajoutés au backlog de produit priorisé peuvent abaisser le niveau de priorité d'autres user stories existant dans le backlog. De telles user stories avec un plus faible niveau de priorité pourraient donc être réalisées plus tard dans le projet en fonction de leur nouvelle priorisation. Puisque les clients sont très impliqués dans la priorisation des exigences et des user stories correspondantes dans le backlog de produit priorisé, cette pratique garantit que les exigences à « forte valeur » pour le client sont réalisées plus rapidement et que le projet commence à offrir de la valeur ajoutée le plus tôt possible.

6.4.2.5 La flexibilité au travers de l'intégration continue

En utilisant des techniques d'intégration continue, les membres de l'équipe Scrum peuvent incorporer des fonctionnalités nouvelles et modifiées aux éléments livrables à chaque fois que cela est possible. Cela atténue le risque que différentes équipes Scrum effectuent des modifications à des composants redondants (p. ex. du code obsolète dans des produits logiciels, d'anciennes conceptions pour des pièces fabriquées). Cela permet d'assurer que le travail est effectué uniquement sur la fonctionnalité ou version la plus récente afin d'éviter les problèmes de compatibilité.

6.5 Intégration des changements

En fonction du secteur et du type de projet, l'ordre de priorité des fonctionnalités et des exigences pour un projet peut rester fixe sur une durée importante ou peut changer fréquemment. Si les exigences d'un projet sont généralement stables, il n'y a normalement que des changements mineurs apportés au backlog de produit priorisé durant le développement et l'équipe Scrum peut travailler de façon séquentielle afin de réaliser les exigences qui offriront la valeur optimale au client selon l'ordre établi dans le backlog de produit priorisé. Dans des environnements aussi stables, la durée d'un sprint est généralement plus longue, entre 4 et 6 semaines.

Si les exigences du projet changent durant le projet, par exemple à cause d'un changement des exigences commerciales, la même méthode continue de fonctionner. Avant d'entamer un sprint, durant le processus de *création du backlog de produit priorisé* ou d'*Raffinage du Product Backlog priorisé*, les exigences à plus forte priorité dans le backlog sont normalement sélectionnées afin d'être réalisées pendant ce sprint. Puisque les changements ont été pris en compte dans le backlog de produit priorisé, l'équipe a uniquement besoin de déterminer le nombre de tâche qu'elle peut accomplir durant le sprint en fonction du temps et des ressources disponibles. La gestion des changements est exécutée lors des processus continus de priorisation et d'addition des tâches au backlog priorisé.

6.5.1 Changements apportés à un sprint

S'il existe une demande de changement qui pourrait avoir un impact important sur les progrès d'un sprint, le Product Owner, après avoir consulté parties prenantes de l'entreprise concernées, décide si le changement peut attendre jusqu'au prochain sprint ou s'il représente une situation urgente qui nécessite d'annuler le sprint en cours et d'en commencer un nouveau.

L'approche Scrum établit clairement que la portée d'un sprint ne peut pas être modifiée après que le sprint a débuté. Si le changement requis est tellement important que les résultats du sprint n'auraient aucune valeur sans lui, alors le sprint doit être annulé. Sinon, le changement est alors intégré lors d'un prochain sprint (comme illustré dans la figure 6-6).

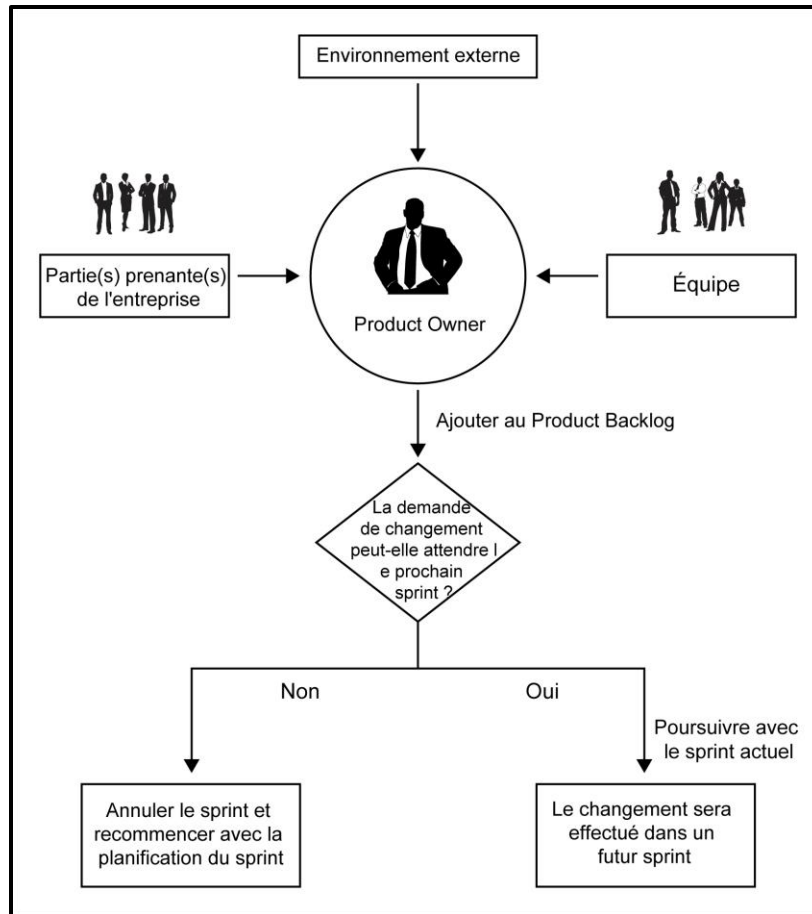


Figure 6-6: Intégration des changements dans Scrum

Il n'existe qu'une exception à cette règle interdisant de modifier la portée d'un sprint après le début de celui-ci. Si l'équipe Scrum détermine qu'elle a largement surestimé les efforts pendant le sprint et a la capacité de mettre en œuvre des user stories supplémentaires, l'équipe peut demander au Product Owner quelles user stories supplémentaires pourraient être incluses dans le sprint en cours.

En verrouillant la portée de chaque sprint, l'équipe est capable d'optimiser et de gérer son travail et ses efforts de façon efficace. Un avantage supplémentaire est que l'équipe n'a pas besoin de s'inquiéter d'avoir à gérer des changements après qu'elle a commencé à travailler sur un sprint. C'est un gros avantage de l'approche Scrum comparée à la gestion de projet traditionnelle.

Dans la gestion de projet traditionnelle, les changements peuvent être demandés et approuvés à n'importe quel moment au cours du cycle du projet. Cela crée souvent la confusion pour les membres de l'équipe du projet, diminue la motivation de l'équipe à cause du manque de continuité et résulte en un manque de concentration avec une équipe qui a le sentiment que « rien n'est jamais fait ». Au contraire, dans les projets Scrum, les changements ne sont pas autorisés après que le sprint a débuté. Cela garantit que dans chaque sprint, l'équipe réalise les éléments livrables et termine les tâches. Par ailleurs, l'entreprise reconnaît les bénéfices tangibles offerts par les incréments potentiellement livrables à la fin de chaque sprint.

De plus, puisque le Product Owner et parties prenantes de l'entreprise sont conscientes du fait que les changements ne sont pas autorisés une fois qu'un sprint a débuté et que les sprints durent entre 1 et 6 semaines, ils définissent et priorisent les exigences durant les processus adéquats de *développement des épics*, de *création du backlog de produit priorisé* et d'*Raffinage du Product Backlog priorisé*.

6.5.1.1 Impact des changements prévus sur la durée du sprint

Puisque les changements ne sont pas autorisés pendant un sprint, l'impact et la fréquence des changements prévus peut avoir un effet sur la décision associée à la durée du sprint lorsque celle-ci est déterminée lors du processus de *planification de la release*.

Si les exigences du projet sont généralement stables et que des changements importants ne sont pas attendus dans un futur proche, la durée d'un sprint peut être plus longue, entre 4 et 6 semaines. Cela offre aux membres de l'équipe Scrum la stabilité nécessaire pour travailler sur les exigences du backlog de produit priorisé pour de longues périodes de temps sans avoir à passer par les processus de *création des user stories*, d'*estimation des user stories*, de *validation des user stories*, d'*identification des tâches*, d'*estimation des tâches* et les autres processus associé qui doivent être réalisés pour chaque sprint.

Cependant, si les exigences d'un projet ne sont pas suffisamment bien définies ou que des changements importants sont attendus dans un futur proche, la durée d'un sprint doit être plus courte, entre 1 et 3 semaines. Cela offre aux membres de l'équipe Scrum la stabilité nécessaire pour travailler sur des sprint plus courts et fournir des résultats qui peuvent être évalués par le Product Owner et parties prenantes de l'entreprise à la fin du sprint. Cela leur offre également suffisamment de flexibilité pour clarifier les exigences et effectuer des changements dans le backlog de produit priorisé à la fin de chaque sprint.

Afin d'optimiser les bénéfices d'un projet Scrum, il est toujours recommandé de limiter la durée d'un sprint à 4 semaines ou moins, sauf pour les projets avec des exigences stables, auquel cas la durée des sprints peut s'étendre jusqu'à 6 semaines.

La figure 6-7 illustre l'impact de la probabilité de changement sur la durée du sprint.

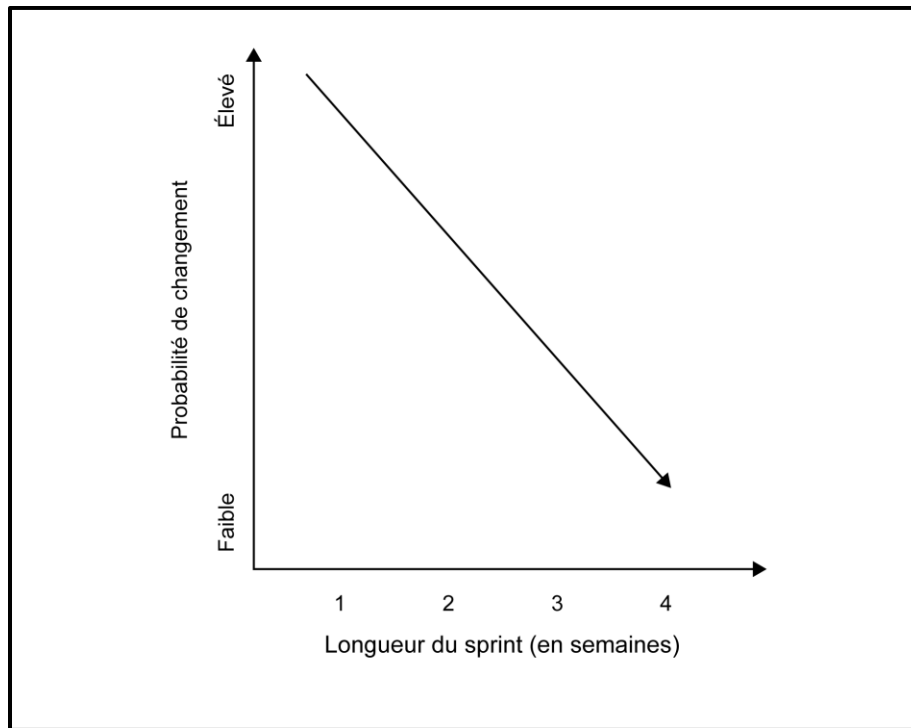


Figure 6-7: Impact des changements prévus sur la durée du sprint

Un sprint est généralement limité dans le temps et dure de une à quatre semaines. La plupart des projets Scrum ont généralement des sprints limités dans le temps d'une durée de deux ou trois semaines.

Cependant, il est important de noter que les changements prévus ne sont pas les seuls pris en compte pour déterminer la durée d'un sprint. D'autres facteurs à prendre en compte incluent notamment :

- le temps réellement disponible afin d'effectuer le travail (si l'environnement du projet ou de l'entreprise nécessite une durée nécessaire pour réaliser les tâches, cela peut déterminer la durée du sprint)
- la date de release planifiée (la durée du sprint doit prendre en compte les dates de release pour le produit ou service dans son ensemble)
- tout autre facteur établi par le Product Owner ou le Scrum Master, devant être pris en compte afin de déterminer la durée du sprint

Il est important de noter que la décision de changer la durée du sprint ne doit pas être prise à la légère ou de façon périodique (p. ex. il n'est pas conseillé d'avoir un sprint d'une durée de 3 semaines, puis de 2 semaines, puis de 4 semaines, etc.). Dans l'idéal la durée du sprint doit rester constante. L'impact le plus important relatif à la modification de la durée du sprint est que cela force à réinitialiser tous les processus de suivi à l'échelle du projet. Les vélocités précédentes deviennent alors obsolètes pour prévoir et planifier les sprints à venir.

Sans une vélocité précise (qui est un indicateur primordial dans tout projet Scrum), il est impossible de mesurer l'efficacité de l'équipe Scrum et celle-ci ne peut pas correctement choisir le nombre d'user stories à développer lors de la planification du prochain sprint. Une fois que la durée du sprint est établie, il est donc préférable de la maintenir constante pour toute la durée du projet ou pour plusieurs cycles de sprint.

6.5.1.2 Gérer les changements au travers de l’Affinage du Product Backlog priorisé

En général, le backlog de produit priorisé contient toutes les user stories, leur estimation en temps (y compris toutes les estimations révisées), et le statut des exigences à haut niveau de priorité. Toute user story nouvelle ou révisée résultant d'un changement des exigences commerciales, des demandes ou client, des conditions de marché extérieures, et/ou des enseignements tirés des sprints précédents y sont également intégrées.

Une des responsabilités principales du Product Owner consiste à affiner le backlog de produit priorisé afin de garantir que les exigences priorisées du backlog qui doivent être incluses dans les deux ou trois prochains sprints sont affinées sous la forme d'user stories adaptées. Il est recommandé que le Product Owner consacre un temps important à l'affinage du backlog de produit priorisé au cours de chaque sprint. Le Product Owner a la responsabilité d'ajouter et de réviser les éléments du backlog de produit priorisé en réponse aux changements. Il est également responsable de fournir des user stories plus détaillées qui seront utilisées pour le prochain sprint.

Cela permet de garantir que l'affinage des exigences et des user stories associées est effectué suffisamment en avance de la réunion de planification de sprint. De cette manière, l'équipe dispose d'un ensemble de stories correctement analysées et clairement définies pouvant facilement être divisées en tâches et estimées. Sur la base des enseignements du sprint en cours, il peut y avoir des changements aux exigences, ou une re-priorisation qui pourront éventuellement être intégrés aux sprints à venir. L'affinage soutient et améliore la flexibilité du modèle Scrum en intégrant les dernières perspectives commerciales et techniques dans les sprints à venir.

Une réunion de revue du backlog de produit (aussi connue sous le nom de session d'affinage des besoins du backlog de produit priorisé) est une réunion formelle tenue durant le processus d'*affinage du backlog de produit priorisé* qui aide l'équipe Scrum à passer en revue et à parvenir à un consensus au sujet du backlog. Cependant, en plus de la réunion de revue du backlog de produit priorisé, l'affinage du backlog doit être effectué tout au long du projet et peut inclure des situations dans lesquelles le Product Owner écrit de nouvelles user stories ou répertorie les user stories dans le backlog de produit priorisé existant, où les membres de l'équipe Scrum ou parties prenantes de l'entreprise donnent au Product Owner leurs suggestions pour de nouvelles user stories, et cætera.

Il est important de noter que les éléments contenus dans le backlog de produit priorisé sont toujours susceptibles d'être réestimés jusqu'à temps que le backlog de sprint soit finalisé lors du processus de *Mise à jour du Sprint Backlog*. Après cela, des changements peuvent toujours être faits, si nécessaire, jusqu'à immédiatement avant la réunion de planification du sprint.

6.5.1.2.1 Revue du backlog de produit efficace (ou session d’Affinage du Product Backlog priorisé)

C’est le Product Owner qui mène la réunion de revue du backlog de produit est tenue lors du processus d’*Affinage du Product Backlog priorisé*. Il est important que le Product Owner établisse les objectifs et idéalement développe un ordre du jour avant que la réunion de revue du backlog de produit ne débute. Sans quoi la session manquera de structure et pourrait s’avérer improductive. Il est également important de limiter le nombre de parties prenantes de l’entreprise participant à la réunion. Un trop grand nombre de participants tend à réduire l’efficacité générale de la réunion. Le Product Owner ne devrait inviter que parties prenantes de l’entreprise dont le feedback est nécessaire pour la session d’affinage. Tous les membres de l’équipe Scrum devraient être inclus car leur contribution est précieuse au travail effectué et à tout problème rencontré. Si la session d’affinage résulte en une re-priorisation ou un changement dans le backlog de produit priorisé, il est important que l’équipe donne son accord à ces changements.

Une session d’affinage efficace doit résulter en des éléments clairement définis dans le backlog de produit priorisé de façon à ce que l’équipe Scrum comprenne parfaitement les exigences du client. Cela aide aussi l’équipe à se familiariser avec toutes les user stories au cas où une ou plusieurs d’entre elles doivent rapidement être incluses dans un sprint. Les critères d’acceptation et la définition de *terminé* peuvent également être discutés lors des sessions d’affinage.

L’approche Scrum ne limite pas les activités d’affinage dans le temps. L’Affinage du Product Backlog priorisé est une activité continue pour le Product Owner.

6.5.1.3 Gestion des changement lors de la démonstration et de la validation du sprint

Le Product Owner a le dernier mot concernant les éléments du backlog de produit priorisé et le fait d’accepter ou de rejeter n’importe quelle user story (correspondant à une demande de changement approuvée) présentée durant le processus de *démonstration et de validation du sprint*. Cependant, c’est le Scrum Master qui a la responsabilité de s’assurer que les exigences et les critères d’acceptation ne sont pas modifiés durant la réunion de revue de sprint pour les user stories réalisées durant le sprint en cours. Cela permet d’éviter le rejet d’user stories réalisées basé sur le fait qu’elles ne respectent pas des exigences récemment modifiées. Si des exigences doivent être modifiées, tout élément correspondant dans le backlog devra être revu afin de prendre en compte les exigences modifiées, lors d’un futur sprint.

6.6 Changements apportés aux portefeuilles et aux programmes

Tout changement survenant soit dans un programme ou un portefeuille peut entraîner une réaction en chaîne sur tous les projets et sprints qui en dépendent. Il est donc conseillé de minimiser les changements à ces niveaux. Si un changement est nécessaire et que toutes parties prenantes de l’entreprise sont d’accord pour effectuer le changement à ces niveaux, les aspects suivants doivent être pris en compte.

6.6.1 Dans un portefeuille

1. Il n'est pas recommandé d'effectuer des changements entre les réunions portant sur le backlog du portefeuille.
2. Si le changement est minime, le Product Owner du portefeuille peut obtenir l'approbation des parties prenantes de l'entreprise concernées (p. ex. sponsor, client et utilisateur final) et ensuite ajouter les exigences au backlog du portefeuille. Les Product Owners des programmes et projets doivent prendre ces exigences en compte et les inclure dans les sprints à venir.
3. Si le changement est important, les efforts du portefeuille ainsi que des programmes, projets et sprints associés doivent être annulés et une réunion de revue du backlog du portefeuille devra être organisée afin de décider des prochaines mesures à prendre.
4. Les réunions de revue du backlog de produit priorisé du portefeuille (aussi appelées réunions de revue du backlog du portefeuille) doivent être tenues à des intervalles de 4 à 12 mois. La fréquence et l'impact des changements apportés à un portefeuille sont le principal facteur déterminant la durée entre deux réunions de revue du backlog du portefeuille. S'il existe plusieurs changements prévus dans un portefeuille, il est préférable de tenir les réunions de revue du backlog du portefeuille à des intervalles plus réguliers (p. ex. 4 à 6 mois), mais s'il y a moins de changements prévus et si les exigences sont stables, la durée entre deux réunions de revue du backlog du portefeuille peut être augmentée (p. ex. 9 à 12 mois).

6.6.2 Dans un programme

1. Il n'est pas recommandé d'effectuer des changements entre les réunions de revue du backlog du programme.
2. Si le changement est minime, le Product Owner du programme peut obtenir l'approbation des parties prenantes de l'entreprise concernées (p. ex. sponsor, client et utilisateur final) et du Product Owner du portefeuille et ensuite ajouter les exigences au backlog du programme. Les Product Owners des projets doivent prendre ces exigences en compte et les inclure dans les sprints à venir.
3. Si le changement est important, les efforts du programme ainsi que les projets et sprints associés doivent être annulés et une réunion de revue du backlog du programme devra être organisée afin de décider des prochaines mesures à prendre.
4. Les réunions de revue du backlog de produit priorisé du programme (aussi appelées réunions de revue du backlog du programme) doivent être tenues à des intervalles de 2 à 6 mois. La fréquence et l'impact des changements apportés à un programme sont le principal facteur déterminant la durée entre deux réunions de revue du backlog du programme. S'il existe plusieurs changements prévus dans un programme, il est préférable de tenir les réunions de revue du backlog du programme à des intervalles plus réguliers (p. ex. 2 à 3 mois), mais s'il y a moins de changements prévus et si les exigences sont stables, la durée entre deux réunions de revue du backlog du programme peut être augmentée (p. ex. 5 à 6 mois).

La figure 6-8 montre comment les changements peuvent être gérés dans le flux Scrum pour les programmes et les portefeuilles.

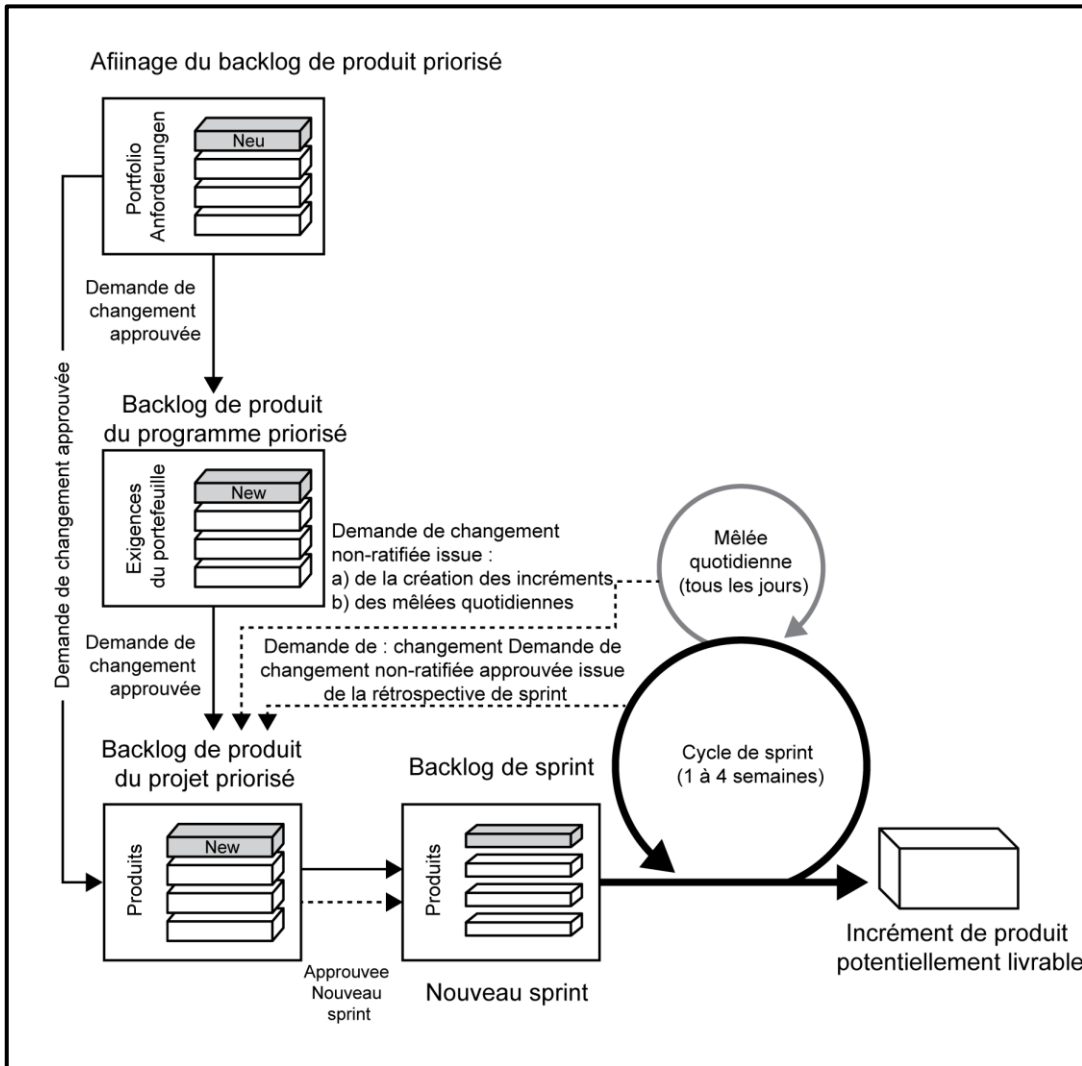


Figure 6-8: Intégrer les changements dans les portefeuilles et les programmes

6.7 Résumé des responsabilités

| Rôle | Responsabilités |
|---------------------------------------|---|
| Équipe Scrum | <ul style="list-style-type: none"> Suggère des améliorations ou des changements lors du processus de <i>création des éléments livrables</i> et lors des <i>mêlées quotidiennes</i> |
| Product Owner/ Chief Product Owner | <ul style="list-style-type: none"> Formule les demandes de changement pour un projet Évalue l'impact des demandes de changement formulées pour le portefeuille, programme ou projet Priorise les user stories dans le backlog de produit priorisé du projet Évalue l'impact des problèmes identifiés par l'équipe Scrum sur les objectifs du projet Communique clairement avec parties prenantes de l'entreprise au sujet des éléments repriorisés du backlog de produit |
| Scrum Master/ Chief Scrum Master | <ul style="list-style-type: none"> Facilite l'identification, l'évaluation et la remontée des problèmes et demandes de changement par l'équipe Scrum |
| Product Owner du programme | <ul style="list-style-type: none"> Formule les demandes de changement pour les programmes Approuve les produits qui sont modifiés, supprimés ou ajoutés en fonction des exigences du programme |
| Scrum Master du programme | <ul style="list-style-type: none"> Facilite l'identification, l'évaluation et la gestion des demandes de changement pour les programmes |
| Product Owner du portefeuille | <ul style="list-style-type: none"> Formule les demandes de changement pour les portefeuilles Approuve les produits qui sont modifiés, supprimés ou ajoutés en fonction des exigences du portefeuille |
| Scrum Master du portefeuille | <ul style="list-style-type: none"> Facilite l'identification, l'évaluation et la gestion des demandes de changement pour les portefeuilles |
| Parties prenantes de l'entreprise | <ul style="list-style-type: none"> Formulent des demandes de changement Participent à la validation et à la priorisation des demandes de changement |
| Scrum Guidance Body | <ul style="list-style-type: none"> Offre des conseils sur les procédures de gestion des changements à suivre tout au long du projet |

Tableau 6-1: Résumé des responsabilités relatives aux changements

6.8 L'approche Scrum comparée à la gestion de projet traditionnelle

Dans les projets gérés de façon traditionnelle, la gestion des changements est étroitement liée à la gestion de la configuration. Le chef de projet se voit attribuer des seuils à l'intérieur desquels il peut gérer les activités et les décisions quotidiennes du projet. Tous les changements sont évalués en fonction de la magnitude de leur variation comparée à une valeur de référence. Des niveaux de tolérance sont indiqués au chef de projet au sein desquels il peut gérer les activités et décisions quotidiennes associées au projet. Lorsqu'une demande de changement dépasse les niveaux de tolérance définis, le chef de projet doit faire remonter le changement proposé à un niveau hiérarchique supérieur et attendre qu'une décision soit prise avant d'intégrer le changement. Le chef de projet enregistre d'abord la demande de changement dans un registre des problèmes ou journal des modifications avant de faire remonter le changement aux autorités supérieures. Celles-ci peuvent notamment inclure le sponsor du projet ainsi que parties prenantes de l'entreprise et décideurs concernés. À un certain moment, une étude de l'impact sera réalisée. En fonction de l'impact estimé du changement, une décision sur l'intégration ou non du changement sera prise. Le chef de projet peut alors proposer des solutions aux possibles problèmes causés par le changement. Si les autorités supérieures décident de valider le changement, le chef de projet a la responsabilité de s'assurer que le changement est correctement intégré.

Les changements dans Scrum fonctionnent de façon très différente comparée à la gestion de projet traditionnelle. L'approche Scrum est fortement orientée vers une gestion rapide et efficace des changements. Dès que le Product Owner ou l'équipe Scrum reconnaît un problème, un défaut ou identifie un élément du backlog de produit priorisé qui doit être modifié, remplacé ou ajouté, le changement est apporté au backlog. De la même façon, la haute direction, le Product Owner ou parties prenantes de l'entreprise peuvent ajouter des demandes de changement au backlog de produit priorisé. Le Product Owner et parties prenantes de l'entreprise approuvent les demandes de changement et repriorisent le backlog en conséquence. Dès qu'il est nécessaire de faire face à un problème ou à une nouvelle exigence de façon immédiate et que celui-ci nécessite un changement affectant le sprint en cours, le Product Owner annule le sprint, avec l'accord des parties prenantes de l'entreprise concernées. Une fois annulé, le sprint sera replanifié et recommencera afin d'intégrer les nouvelles exigences.

Cependant, si le problème ou changement n'est pas très important et ne nécessite pas que le sprint en cours soit modifié, le changement sera ajouté au backlog de produit priorisé et intégré à la planification d'un prochain sprint. Cela donne aux parties prenantes de l'entreprise la capacité de répondre aux changements dans un environnement extérieur, tout en maintenant un certain degré de contrôle sur les activités en cours dans le projet. Aussi, à la fin de chaque sprint, l'équipe Scrum fait la démonstration des éléments livrables *terminés*. Ces éléments sont potentiellement livrables et peuvent être évalués par le Product Owner et les autres parties prenantes de l'entreprise.

7. RISQUES

7.1 Introduction

L'objectif de ce chapitre est de définir le risque, d'aborder la gestion des risques dans un environnement Scrum et d'envisager les outils permettant d'en faciliter la gestion. Afin de garantir la viabilité de l'entreprise, de réduire la probabilité d'échec d'un projet et de prendre des décisions commerciales plus éclairées, il est important que les risques soient gérés efficacement au travers d'une approche organisée et méthodique.

Dans un environnement Scrum, les risques sont généralement minimisés, en grande partie du fait que le travail est réalisé en sprints plutôt qu'en une série continue d'éléments livrables produits sur une courte période de temps. Les éléments livrables sont comparés aux attentes et le Product Owner participe activement au projet. Cependant, même pour les projets les plus simples, des choses peuvent mal se passer. Il est donc important d'avoir une stratégie afin d'identifier et de réagir aux risques.

Tel qu'il est décrit dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, le *risque* s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Ce chapitre est divisé comme suit :

7.2 Guide des rôles — cette section offre des conseils sur les sections pertinentes à chacun des rôles Scrum : Product Owner, Scrum Master et équipe Scrum.

7.3 Qu'est-ce que le risque ? — cette section définit le risque et explique la façon dont il peut affecter les objectifs d'un projet et contribuer à la réussite ou à l'échec de celui-ci.

7.4 Procédure de gestion des risques — cette section présente les principales techniques de gestion des risques et élabore sur le développement de stratégies permettant d'identifier, d'évaluer et de gérer les risques.

7.5 Minimiser les risques avec l'approche Scrum — cette section explique les aspects fondamentaux à Scrum qui en font l'approche idéale pour gérer les risques de façon efficace à différents niveaux : portefeuille, programme et projet.

7.6 Résumé des responsabilités — cette section décrit les responsabilités de chaque personne ou rôle d'un projet concernant la gestion des risques.

7.7 L'approche Scrum comparée à la gestion de projet traditionnelle — cette section traite des avantages relatifs à la gestion des risques selon l'approche Scrum par rapport aux méthodes utilisées dans la gestion de projet traditionnelle.

7.2 Guide des rôles

1. Product Owner — la responsabilité de gérer les risques associés à un projet tient principalement au Product Owner, ce chapitre dans son entier s'applique donc fondamentalement à ce rôle.
2. Scrum Master — le Scrum Master doit également se familiariser avec ce chapitre entier en portant une attention particulière aux sections 7.3, 7.4, et 7.7.
3. Équipe Scrum — l'équipe Scrum devrait surtout se concentrer sur les sections 7.3 et 7.7.

7.3 Qu'est-ce que le risque ?

Le risque est défini comme un évènement incertain pouvant affecter les objectifs d'un projet et contribuer à sa réussite comme à son échec. Les risques susceptibles d'impacter le projet de façon positive sont considérés comme des opportunités, tandis que les menaces sont des risques qui pourraient affecter le projet de façon négative. Les risques doivent être gérés de façon dynamique. La gestion des risques est un processus itératif qui doit débuter durant la phase d'initiation du projet et continuer tout au long du cycle du projet. Ce processus doit respecter des étapes standardisées de façon à assurer l'identification et l'évaluation des risques ainsi que la mise en place et l'exécution d'un plan d'action adapté.

Les risques sont identifiés, évalués et les actions pour y répondre sont mises en place sur la base de deux facteurs principaux : la probabilité que chaque risque se produise et le possible impact en cas d'occurrence. Les risques à fort taux de probabilité et d'impact doivent être pris en compte en priorité. Généralement, une fois le risque identifié, il est important de comprendre les aspects de base de celui-ci en termes de causes possibles, de zone d'incertitude et d'effets potentiels si le risque venait à se concrétiser.

7.3.1 Différences entre les risques et les problèmes

Les risques sont les incertitudes associées à un projet qui pourraient avoir un impact important sur les résultats du projet de façon positive ou négative. Puisque les risques sont des incertitudes futures, ils n'ont pas d'impact actuel sur le projet mais pourraient avoir un impact potentiel dans le futur. Les exemples suivants constituent des risques :

- même après une formation approfondie, les représentants du service client pourraient ne pas être prêts à recevoir les commandes à la date de lancement
- l'équipe de peintres pourrait être retardée à cause des fortes pluies ce qui pourrait avoir un impact négatif sur le calendrier du projet

Les problèmes sont généralement des certitudes spécifiques qui sont actuellement présentes dans le projet, il n'y a donc pas besoin d'effectuer une évaluation de la probabilité comme c'est le cas pour les risques. Les problèmes doivent être traités. Certains exemples de problèmes incluent notamment :

- le financement est rejeté
- les exigences sont vagues

Les risques qui ne sont pas abordés à temps peuvent devenir des problèmes. L'objectif de la gestion des risques est de se préparer, en mettant en place des stratégies, afin de répondre aux risques qui pourraient survenir.

7.3.2 Attitude face aux risques

Parties prenantes de l'entreprise incluent toutes les personnes ou organisations impactées par le projet ainsi que celles qui ont la capacité d'impacter le projet. Il est important de comprendre l'attitude des parties prenantes de l'entreprise face aux risques. L'attitude face aux risques est influencée par les trois facteurs suivants :

1. la propension au risque : fait référence au degré d'incertitude qu'une partie prenante ou organisation est prête à accepter.
2. la tolérance aux risques : indique le degré, la quantité ou le volume de risques que parties prenantes de l'entreprise sont prêtes à supporter.
3. le seuil de risque : fait référence au niveau de risque acceptable pour la partie prenante ou l'organisation. Un risque se trouve soit au-dessus soit en-dessous du seuil de risque. S'il se trouve en-dessous, alors la partie prenante ou l'organisation acceptera le risque plus facilement.

Essentiellement, l'attitude des parties prenantes de l'entreprise face aux risques détermine le niveau de risque qu'elles considèrent acceptable et donc quand elles décideront de prendre des mesures en vue d'atténuer les impacts négatifs potentiels des risques. Il est donc important de comprendre les niveaux de tolérance des parties prenantes de l'entreprise en relation à différents facteurs, notamment les coûts, la qualité, la portée et le calendrier.

Utility function est un modèle utilisé pour mesurer l'attitude ou les préférences des parties prenantes de l'entreprise face aux risques. Il définit le niveau ou la propension des parties prenantes de l'entreprise à accepter les risques. Les trois catégories d'*utility function* sont les suivantes :

1. Aversion aux risques La partie prenante est réticente à accepter un risque peu importe les bénéfices ou opportunités anticipés.

2. Neutre au risque La partie prenante n'est ni averse au risque ni risquophile et aucune décision n'est affectée par le niveau d'incertitude des résultats. Quand deux scénarios possibles comportent les mêmes niveaux d'avantages, la partie prenante neutre au risque ne se préoccupera pas du fait qu'un scénario soit plus risqué que l'autre.
3. Risquophile La partie prenante est prête à accepter le risque même s'il n'offre qu'une amélioration ou un bénéfice marginal en retour pour le projet.

7.4 Procédure de gestion des risques

La gestion des risques se compose des cinq étapes suivantes, qui doivent être réalisées de façon itérative tout au long du projet :

1. identification des risques : utilisation de différentes techniques afin d'identifier tous les risques potentiels
2. évaluation des risques : évaluation et estimation des risques identifiés
3. priorisation des risques : priorisation des risques à inclure dans le backlog de produit priorisé
4. limitation des risques : développement d'une stratégie adéquate afin de répondre aux risques
5. communication des risques : communication des résultats des quatre premières étapes de la gestion des risques aux parties prenantes de l'entreprise concernées et détermination de leur perception des événements incertains

7.4.1 Identification des risques

Les membres de l'équipe Scrum doivent tenter d'identifier tous les risques qui pourraient potentiellement impacter le projet. Ils ne peuvent réaliser cette étape consciencieusement qu'en évaluant le projet à partir de différents points de vue et en utilisant différentes techniques. L'identification des risques est effectuée tout au long du projet. Les risques identifiés contribuent à plusieurs processus Scrum dont la *création du backlog de produit priorisé*, l'*Raffinage du Product Backlog priorisé* et la *démonstration et validation du sprint*.

Les techniques suivantes sont couramment utilisées afin d'identifier les risques.

7.4.1.1 Techniques d'identification des risques

1. Évaluation des enseignements issus des processus de rétrospective de sprint et de projet

Les enseignements issus de projets similaires, ou des sprints précédents dans le même projet, et l'exploration des incertitudes qui ont affectés ces projets et sprints peut être un moyen utile d'identifier les risques.

2. Listes de contrôle des risques

Les listes de contrôle des risques peuvent inclure les points principaux à prendre en considération lors de l'identification des risques, les risques les plus courants rencontrés au cours du projet Scrum, ou même les catégories de risques que l'équipe doit prendre en compte. Les listes de contrôle sont des outils précieux afin de garantir l'identification globale des risques.

3. Listes de risques

Les listes de risques sont utilisées pour stimuler la réflexion concernant la source à l'origine des risques. Les listes de risques pour différents secteurs et types de projets sont disponibles publiquement.

4. Brainstorming

Sessions durant lesquelles parties prenantes de l'entreprise concernées et les membres de l'équipe Scrum principale partagent ouvertement leurs idées à travers des discussions et des sessions de partage des connaissances, normalement menées par un facilitateur.

5. Structure de répartition des risques

Un des principaux outils utilisés pour l'identification des risques est une structure de répartition des risques. Dans cette structure les risques sont regroupés en fonction de leurs catégories et points communs. Par exemple, les risques peuvent être catégorisés comme étant d'ordre financier, technique ou relatifs à la sécurité. Cela permet à l'équipe de mieux planifier et répondre à chaque risque.

6. Entretiens

Les entretiens et réunions informelles avec les parties prenantes de l'entreprise ou autres, telles que les cadres supérieurs, les membres de l'équipe Scrum, les experts techniques, les clients et les utilisateurs finaux peuvent aider à identifier les risques du projet.

7.4.2 Évaluation des risques

L'évaluation des risques aide à comprendre l'impact potentiel d'un risque, sa probabilité d'occurrence et quand le risque pourrait se matérialiser. L'effet global sur la valeur commerciale doit être estimé, si l'impact est suffisamment important pour l'emporter sur la justification commerciale, une décision doit être prise quant au fait de continuer ou non le projet.

L'évaluation des risques doit prendre en compte la probabilité, la proximité et l'impact. La probabilité du risque fait référence à la possibilité que le risque se produise, tandis que la proximité fait référence au moment où le risque pourrait survenir. L'impact fait référence à l'effet probable du risque sur le projet ou sur l'organisation.

Afin d'estimer la probabilité d'un risque, différentes techniques peuvent être utilisées, notamment les arbres de probabilité, l'analyse de Pareto et les grilles de probabilité et d'impact.

En plus de la probabilité, l'évaluation des risques évalue également les potentiels effets nets des risques sur le projet ou sur l'organisation. Ces effets peuvent être estimés en utilisant des techniques telles que les modèles de risque et la valeur monétaire attendue.

7.4.2.1 Techniques d'évaluation des risques

1. Réunion de risques

Les risques peuvent être facilement priorisés par le Product Owner en organisant une réunion avec l'équipe Scrum principale et éventuellement en invitant parties prenantes de l'entreprise concernées à y participer. Les membres de l'équipe peuvent se réunir afin de prioriser les différents risques en fonction de leur évaluation subjective de l'impact des risques sur les objectifs du projet.

2. Arbres de probabilité

Les évènements potentiels sont représentés sous la forme d'un arbre et chaque résultat possible d'un risque est associé à une branche. La probabilité de chaque résultat possible est indiquée sur la branche correspondante puis multipliée par l'impact estimé afin d'obtenir une valeur prévue pour chaque résultat possible. Les valeurs de résultat sont ensuite ajoutées les unes aux autres afin de calculer l'impact prévu d'un risque sur un projet (voir figure 7-1).

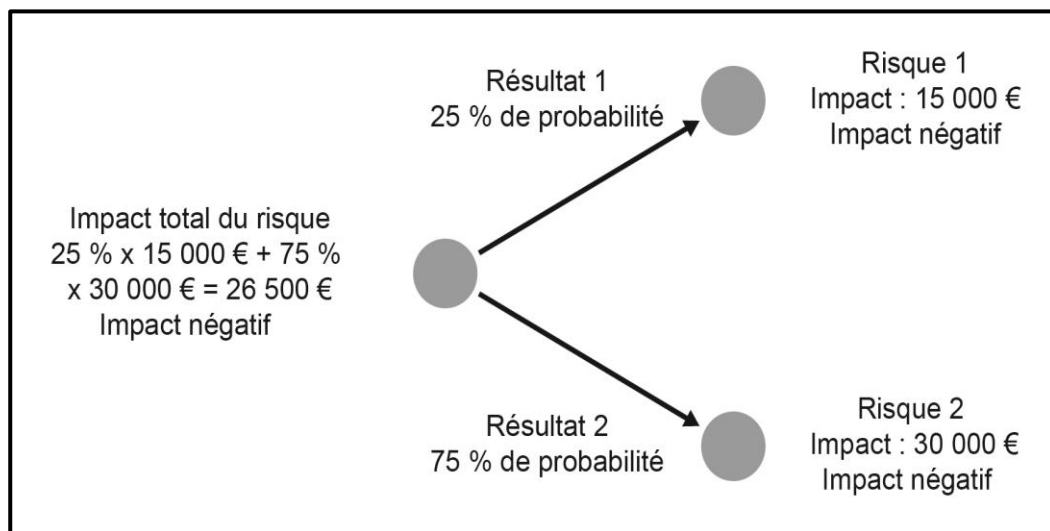


Figure 7-1: Exemple d'arbre de probabilité

3. Analyse de Pareto

Cette technique d'évaluation des risques implique de classer les risques en fonction de leur magnitude ce qui aide l'équipe Scrum à répondre aux risques selon l'ordre de leur impact potentiel sur le projet. Par exemple, dans la figure 7-2, le risque 1 a le plus fort impact et devrait préférablement être traité en premier.

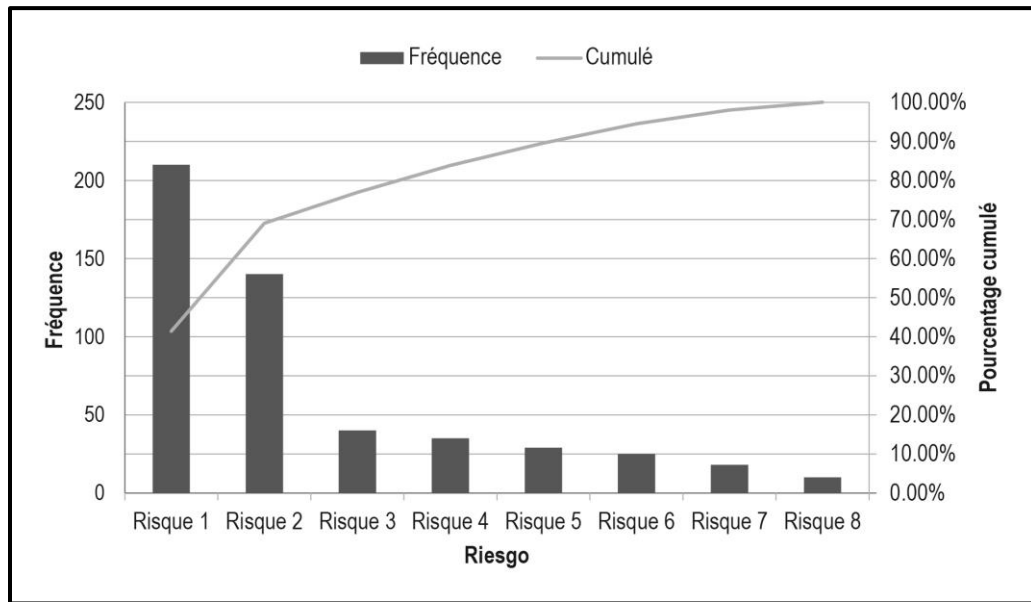


Figure 7-2: Exemple de diagramme de Pareto

4. Grille de probabilité et d'impact

Chaque risque est évalué en fonction de sa probabilité d'occurrence et de son impact potentiel sur les objectifs du projet. Généralement, un score numérique est attribué indépendamment à la probabilité et à l'impact. Les deux valeurs sont ensuite multipliées pour obtenir un score de sévérité de risque (ou valeur PI) qui peut être utilisé afin de prioriser les risques.

Par exemple, le score de gravité d'un risque dont la probabilité est de 50 % et l'impact de 0,6, où l'impact se situe sur une échelle de 0 (faible) à 1 (très élevé), serait calculé comme suit :

$$0,5 \text{ (probabilité)} \times 0,6 \text{ (impact)} = 0,3$$

Les systèmes de notation utilisés sont déterminés au sein de l'organisation ou pour le projet. Une échelle décimale allant de zéro à un est souvent utilisée, dans ce cas un taux de probabilité de 0,5 indique une possibilité de 50 %. D'autres options incluent notamment une échelle allant de un à dix, ou Fort (3), Moyen (2) et Faible (1).

La figure 7-3 illustre l'utilisation d'une échelle décimale. Chaque risque est évalué en fonction de sa probabilité d'occurrence et de son impact sur une échelle objective.

| Grille de probabilité et d'impact | | | | | | | |
|-----------------------------------|---------|---------------|--------------|--------------|-------------|--------------|---------------|
| Probabilité | Menaces | | | Opportunités | | | |
| | 0,90 | 0,09 | 0,27 | 0,72 | 0,72 | 0,27 | 0,09 |
| | 0,75 | 0,075 | 0,225 | 0,60 | 0,60 | 0,225 | 0,075 |
| | 0,50 | 0,05 | 0,15 | 0,40 | 0,40 | 0,15 | 0,05 |
| | 0,30 | 0,03 | 0,09 | 0,24 | 0,24 | 0,09 | 0,03 |
| | 0,10 | 0,01 | 0,03 | 0,08 | 0,08 | 0,03 | 0,01 |
| | | Faible 0,1 | Moyen 0,3 | Fort 0,8 | Fort 0,8 | Moyen 0,3 | Faible 0,1 |

Impact

Valeur PI faible
 Valeur PI moyenne
 Valeur PI forte

Figure 7-3: Exemple de grille de probabilité et d'impact

Les méthodes permettant d'attribuer des valeurs de probabilité et d'impact aux risques varient en fonction du projet et du nombre de risques à évaluer ainsi que des processus et procédures qui existent au sein de l'organisation. Cependant, en appliquant une simple formule $P \times I$, il est possible de calculer la sévérité d'un risque sur une échelle numérique ou nominale.

5. Valeur monétaire attendue (VMA)

La valeur monétaire d'un risque est basée sur sa valeur monétaire attendue (VMA). La VMA est calculée en multipliant l'impact monétaire par la probabilité du risque, telle qu'elle a été estimée par le client.

$$\text{Valeur monétaire attendue} = \text{Impact du risque (en monnaie)} \times \text{Probabilité du risque (en pourcentage)}$$

Par exemple, un risque ayant un impact négatif estimé à 1000 € et une probabilité d'occurrence de 50 % résulterait en une VMA comme suit :

$$\text{VMA} = 1000 \text{ €} \times 0,50 = 500 \text{ €}$$

7.4.3 Priorisation des risques

Scrum permet l'identification et l'évaluation rapide des risques. Les risques identifiés sont pris en compte lors du processus de *création du backlog de produit priorisé*, ou d'*Raffinage du Product Backlog priorisé*. Un backlog de produit priorisé peut donc être considéré comme un backlog ajusté en fonction des risques.

Il est possible d'identifier et d'évaluer les risques grâce à n'importe laquelle des méthodes d'identification et d'évaluation des risques mentionnées ci-dessus.

Lors des processus de *création du backlog de produit priorisé* ou d'*Raffinage du Product Backlog priorisé*, les user stories priorisées du backlog de produit existant et la liste de risques priorisés sont fusionnées afin de créer un backlog de produit priorisé mis à jour qui prend en compte les risques identifiés.

Étapes pour la mise à jour du backlog de produit priorisé avec les risques identifiés :

1. créer une liste des risques priorisés (p. ex. les risques peuvent être priorisés en utilisant la technique de valeur monétaire attendue)
2. sélectionner les risques identifiés pouvant être atténués et pour lesquels l'équipe décide de prendre des mesures de risque spécifiques durant le sprint afin d'atténuer de tels risques
3. Ajouter les risques identifiés qui peuvent être atténués à l'étape 2 au Product Backlog priorisé (en tant que User Stories), puis les prioriser pour obtenir le Product Backlog priorisé en fonction du risque.

La figure 7-4 illustre le processus de priorisation des risques.

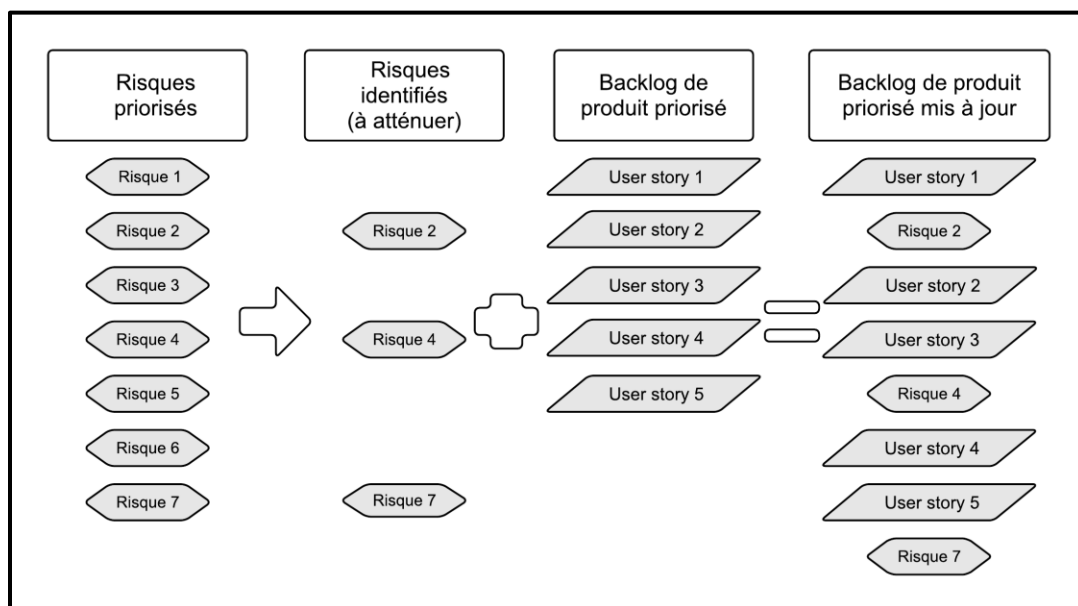


Figure 7-4: Processus de priorisation des risques

7.4.4 Atténuation des risques

La réponse apportée à chaque risque dépend de la probabilité et de l'impact du risque. Cependant, la nature itérative de Scrum avec son temps de réponse et ses cycles de feedback courts permet l'identification rapide des échecs et donc, d'un point de vue pratique, dispose d'un système d'atténuation des risques naturellement intégré.

Les risques peuvent être atténués en mettant en place un certain nombre de mesures. Dans la plupart des cas, les réponses sont proactives ou réactives. Dans le cas d'un risque, il est possible d'établir un plan B pouvant être utilisé comme solution de repli si le risque vient à survenir. Un tel plan B est une réponse réactive. Parfois des risques sont acceptés et sont un exemple de réponse au risque qui n'est ni proactive ni réactive. Les risques peuvent être acceptés pour différentes raisons, par exemple dans des situations où la probabilité ou l'impact du risque sont trop faibles pour y répondre. L'acceptation peut également être de mise dans une situation où l'appréhension d'un risque secondaire pourrait dissuader le Product Owner de toute intervention. Les efforts faits par le Product Owner afin de réduire la probabilité ou l'impact d'un risque (ou les deux) sont un exemple d'une réponse proactive à l'atténuation des risques.

Une fois que les risques identifiés sont inclus dans le backlog de produit priorisé (voir figure 7-4), plusieurs risques sont atténués durant le processus de *création des éléments livrables* au cours duquel les tâches associées aux user stories définies dans le backlog de produit priorisé sont réalisées. Dans Scrum, la responsabilité d'un risque repose clairement sur le Product Owner lorsqu'il s'agit de gérer les risques associés aux aspects commerciaux et sur l'équipe Scrum lorsqu'il s'agit de mettre en œuvre les réponses aux risques au cours d'un sprint. Il est possible de demander les conseils du Scrum Guidance Body au sujet de la façon dont les réponses aux risques sont mises en œuvre et afin de confirmer si les mesures respectent les directives de l'organisation dans son ensemble. Le Scrum Master surveille de près les risques potentiels qui pourraient affecter le projet et tient le Product Owner et l'équipe Scrum informés.

7.4.4.1 Expérimentation des risques

Le concept d'expérimentation des risques peut s'avérer utile afin d'identifier les risques. L'expérimentation des risques implique de rechercher et de créer des prototypes afin de mieux comprendre les risques potentiels. Lors d'une expérimentation, un exercice intense d'une durée de deux ou trois jours est mené (de préférence au début d'un projet, avant les processus de *développement des epics* ou de *création du backlog de produit priorisé*) afin d'aider l'équipe à déterminer les incertitudes qui pourraient affecter le projet. Les expérimentations sur les risques sont utiles lorsque l'équipe Scrum travaille avec et s'habitue à de nouveaux outils ou technologies, ou lorsque les user stories sont particulièrement longues. Elles aident également à estimer la durée et l'effort de façon plus précise.

Les pics basés sur les risques peuvent également être continus pendant toute la durée du projet et peuvent être incorporés pendant n'importe quel sprint. De tels pics devraient être ajoutés au Product Backlog priorisé. L'exploration des pics en fonction des risques est utilisée pour atténuer toute menace potentielle future.

7.4.5 Communication des risques

Étant donné que parties prenantes de l'entreprise ont un intérêt pour le projet, il est important de communiquer avec elles au sujet des risques. Les informations fournies aux parties prenantes de l'entreprise concernant les risques doivent inclure l'impact potentiel et les plans prévus pour répondre à chaque risque. Cette communication est continue et devrait avoir lieu en parallèle aux quatre étapes séquentielles discutées jusqu'à présent : l'identification, l'évaluation, la priorisation et l'atténuation des risques. Il est également possible pour l'équipe Scrum de discuter des risques spécifiques relatifs à leurs tâches avec le Scrum Master durant les mêlées quotidiennes. Le Product Owner est responsable de la priorisation des risques et de communiquer la liste priorisée à l'équipe Scrum.

Un outil important pouvant être utilisé afin de communiquer les informations associées aux risques est le risk burndown chart.

7.4.5.1 Risk Burndown Chart

La gestion des risques est essentielle afin de garantir la création de la valeur, les activités de gestion des risques sont donc réalisées tout au long de la vie du projet et non pas uniquement lors de l'initiation du projet.

Il est possible d'évaluer chaque risque en utilisant différents outils d'évaluation des risques. Cependant, l'outil privilégié afin d'évaluer les risques et de créer un risk burndown chart est la valeur monétaire attendue (VMA) telle qu'elle est décrite dans la section 7.4.2.1.

Les informations collectées durant l'évaluation des risques peuvent être utilisées afin de créer un risk burndown chart. Ce diagramme représente la sévérité cumulée des risques d'un projet dans le temps. Les probabilités des différents risques sont représentées les unes au-dessus des autres afin de montrer les risques cumulés sur l'axe y. L'identification et l'évaluation initiale des risques d'un projet et la création du risk burndown chart sont réalisés au début du projet. De nouveaux risques peuvent ensuite être identifiés à des intervalles de temps prédéterminés et les risques restant doivent être ré-évalués et mis à jour en conséquence sur le diagramme. La réunion de planification de sprint se prête particulièrement à ce procédé. Le suivi des risques de cette manière permet à l'équipe de reconnaître les tendances de l'exposition aux risques et de prendre les actions adaptées si nécessaire.

La figure 7-5 est un exemple de risk burndown chart

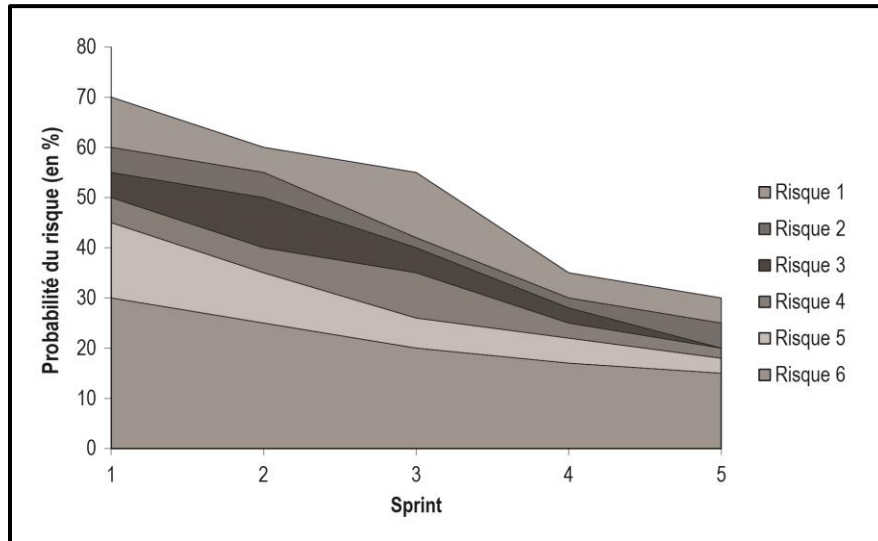


Figure 7-5: Exemple de risk burndown chart

7.5 Minimiser les risques avec l'approche Scrum

En tant que processus Agile itératif, l'approche Scrum minimise naturellement les risques. Les pratiques Scrum suivantes facilitent la gestion efficace des risques.

1. La flexibilité réduit les risques relatifs à l'environnement commercial

La flexibilité de Scrum contribue largement à minimiser les risques en permettant l'ajout ou la modification des exigences à tout moment au cours du cycle du projet. Cela permet à l'organisation de répondre aux menaces ou opportunités dérivées de l'environnement commercial et des exigences inattendues quand elles se présentent et ce avec un coût généralement bas quant à la gestion de tels risques.

2. Le feedback régulier réduit les risques relatifs aux attentes

L'approche Scrum étant itérative, elle offre suffisamment d'opportunités afin d'obtenir du feedback et d'établir les attentes tout au long du cycle du projet. Cela garantit que parties prenantes de l'entreprise du projet, ainsi que l'équipe, ne sont pas pris au dépourvu par des exigences mal communiquées.

3. La responsabilité de l'équipe réduit les risques relatifs à l'estimation

L'équipe Scrum estime et est responsable des éléments du backlog de sprint ce qui permet une estimation plus précise ainsi que la livraison des incréments de produit dans les délais.

4. La transparence réduit les risques de non-détection

Le principe de transparence sur lequel se fonde l'approche Scrum garantit que les risques sont détectés et communiqués le plus tôt possible, entraînant ainsi une meilleure gestion et atténuation des risques. De plus, lors des réunions Scrum of Scrums, il est possible que les obstacles auxquels une équipe fait actuellement face deviennent des risques pour une autre équipe Scrum dans le futur. Cela doit être identifié dans la version mise à jour du registre des obstacles.

5. La livraison en itération réduit les risques relatifs aux investissements

La livraison continue de la valeur ajoutée tout au long du cycle du projet Scrum, lorsque les éléments potentiellement livrables sont créés à la fin de chaque sprint, permet de réduire pour le client les risques relatifs aux investissements.

7.6 Les risques dans les portefeuilles et les programmes

Alors que certains risques sont spécifiquement associés à des projets individuels, d'autres peuvent survenir dans des programmes ou portefeuilles et seront généralement gérés à ces niveaux. Cependant, les risques associés à un portefeuille ou programme auront également un impact sur les projets faisant partie du portefeuille ou du programme en question. Lors de l'évaluation des risques d'un portefeuille ou d'un programme, s'il est établi qu'un risque pourrait affecter un projet individuel, les informations pertinentes concernant le risque doivent être communiquées au Product Owner et à l'équipe Scrum.

En fonction du niveau de sévérité ou de priorité, lorsque l'équipe du programme ou du portefeuille communique le risque qui impactera le projet individuel, il se peut que l'équipe Scrum doive annuler et replanifier le sprint en cours afin de répondre au risque. Dans le cas d'un risque moins urgent, l'équipe peut continuer le sprint en cours et répondre au risque lors d'un prochain sprint.

7.6.1 Dans un portefeuille

1. Lorsque des risques sont identifiés dans un portefeuille, le Product Owner du portefeuille devra les enregistrer et évaluer leur proximité, leur probabilité et leur impact de façon à prioriser et déterminer la réponse appropriée pour le portefeuille.
2. Le Product Owner du portefeuille devra également communiquer les risques aux parties prenantes de l'entreprise, équipes du programme et équipes du projet concernées. Dans certains cas, l'équipe du portefeuille peut avoir à assurer la responsabilité de risques spécifiques.

7.6.2 Dans un programme

1. Lorsque des risques sont identifiés dans un programme, le Product Owner du programme devra les enregistrer dans le backlog de produit priorisé du programme ajusté aux risques et en évaluer la proximité, la probabilité et l'impact de façon à prioriser et déterminer la réponse appropriée pour le programme.
2. Le Product Owner du programme devra également communiquer les risques aux parties prenantes de l'entreprise et aux équipes du projet concernées. Dans certains cas, l'équipe du programme pourrait avoir à assurer la responsabilité de risques spécifiques.

La figure 7-6 montre comment les risques peuvent être gérés dans le cadre du processus Scrum pour les portefeuilles et les programmes.

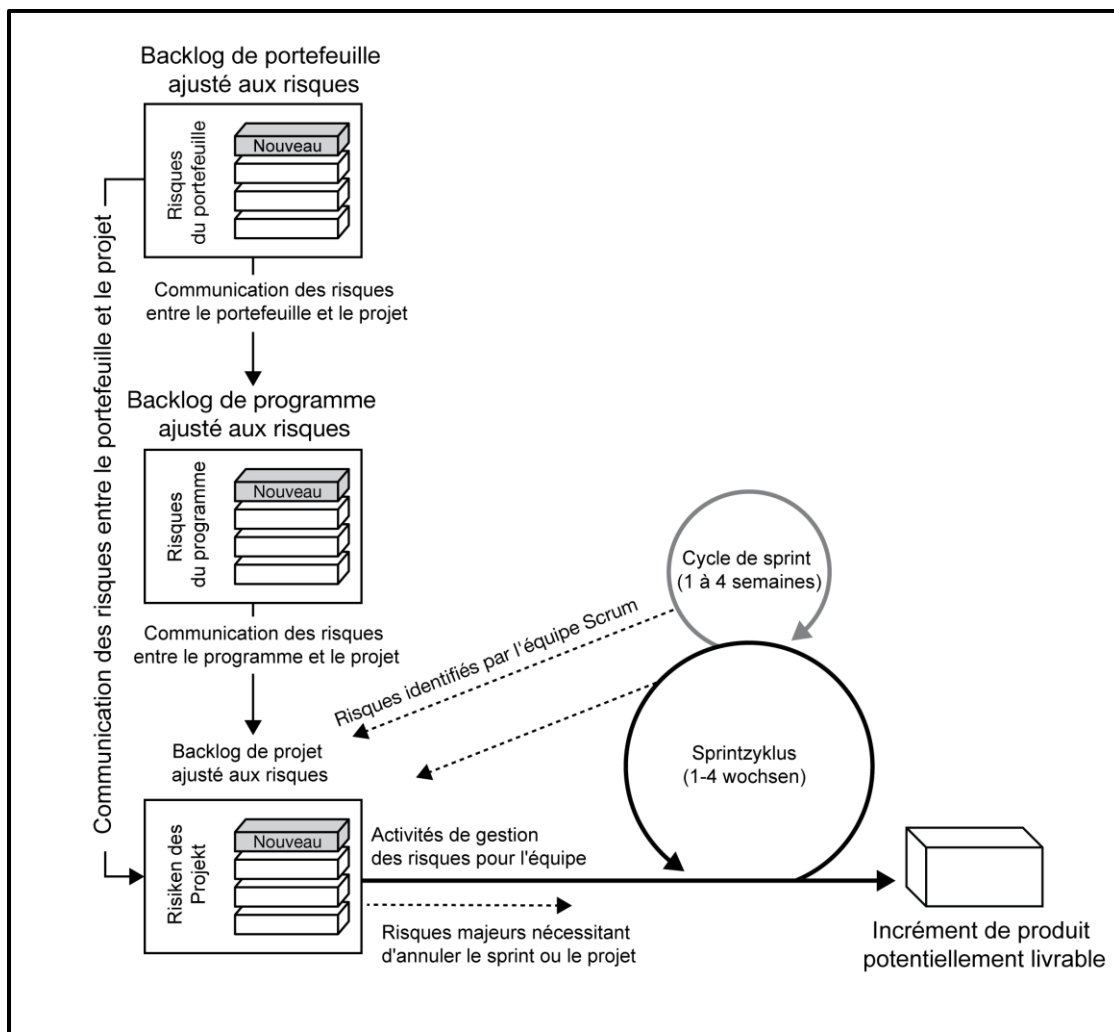


Figure 7-6: Gestion des risques dans les portefeuilles et les programmes

7.7 Résumé des responsabilités

Dans Scrum, les activités relatives à la gestion des risques sont réparties entre les différents rôles et chacun au sein de l'équipe Scrum dispose d'un certain degré de responsabilité, tout comme le Scrum Master facilitant le processus.

| Rôle | Responsabilités |
|---------------------------------------|--|
| Équipe Scrum | <ul style="list-style-type: none"> Identifie les risques lors du développement du produit au cours du processus de <i>création des éléments livrables</i> Met en œuvre les mesures de gestion des risques telles qu'elles sont recommandées par le Product Owner |
| Product Owner/ Chief Product Owner | <ul style="list-style-type: none"> Enregistre et évalue les risques du projet Priorise et communique les risques aux parties prenantes de l'entreprise et équipes de programme et de portefeuille concernées Garantit que les niveaux de risque d'un projet sont dans des limites acceptables |
| Scrum Master/ Chief Scrum Master | <ul style="list-style-type: none"> Facilite l'identification et la remontée des risques par l'équipe Scrum |
| Product Owner du programme | <ul style="list-style-type: none"> Enregistre et évalue les risques du programme Priorise et communique les risques aux parties prenantes de l'entreprise et équipes du projet concernées |
| Scrum Master du programme | <ul style="list-style-type: none"> Facilite l'identification, l'évaluation et la remontée des risques pour les programmes |
| Product Owner du portefeuille | <ul style="list-style-type: none"> Enregistre et évalue les risques du portefeuille Priorise et communique les risques aux parties prenantes de l'entreprise et équipes de programme et de projet concernées |
| Scrum Master du portefeuille | <ul style="list-style-type: none"> Facilite l'identification, l'évaluation et la remontée des risques pour les portefeuilles |
| Parties prenantes de l'entreprise | <ul style="list-style-type: none"> Interagissent avec l'équipe Scrum principale afin de contribuer à la gestion des risques qui affectent la réalisation des résultats et bénéfices attendus pour le projet |
| Scrum Guidance Body | <ul style="list-style-type: none"> Offre des conseils sur les procédures de gestion des risques à suivre tout au long du projet |

Tableau 7-1: Résumé des responsabilités correspondant aux risques

7.8 L'approche Scrum comparée à la gestion de projet traditionnelle

Scrum et la plupart des méthodes de gestion de projet traditionnelles définissent un risque comme un « un ou des événement(s) incertain(s) susceptible(s) d'affecter la réalisation des objectifs du projet de façon positive ou négative ». Les risques sont également identifiés, évalués, planifiés et communiqués de façon continue.

Dans les modèles de gestion de projet traditionnels, l'accent est mis sur la planification préalable détaillée afin d'identifier, d'évaluer et de déterminer les réponses aux risques pour tous les risques associés à un projet. Durant l'exécution d'un projet, tout membre de l'équipe du projet peut identifier les risques. Le chef de projet, le bureau de gestion du projet ou encore l'équipe de support du projet peuvent chacun mettre à jour le registre des risques. Le chef de projet surveille et contrôle régulièrement tous les risques et identifie généralement des personnes spécifiques au sein de l'équipe pour prendre la responsabilité de certains aspects du risque.

Dans l'approche Scrum, n'importe quel membre de l'équipe Scrum peut identifier les risques et le Product Owner peut mettre à jour les risques identifiés dans le backlog de produit priorisé ajusté aux risques. Les principes Scrum de contrôle empirique des processus et de développement itératif permettent à l'équipe Scrum de continuer à constamment identifier les risques et de les ajouter au backlog de produit priorisé, où de tels risques sont priorisés avec les user stories existantes dans le backlog, afin de les atténuer lors des prochains sprints. L'équipe Scrum est collectivement responsable de la gestion des risques du sprint.

8. INITIATION

Ce chapitre détaille les processus relatifs à l'initiation d'un projet : *Création d'une vision de projet, identification du Scrum Master et des parties prenantes de l'entreprise, formation de l'équipe Scrum, développement des epics, création d'un backlog de produit priorisé et planification de la release.*

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la phase d'initiation s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Afin de faciliter la meilleure application possible de l'approche Scrum, ce chapitre identifie les contributions, outils et résultats de chaque processus comme étant soit « obligatoires » ou « facultatifs ». Les contributions, outils et résultats accompagnés d'un astérisque (*) sont obligatoires ou essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont facultatifs.

Il est recommandé que l'équipe Scrum et les personnes qui découvrent l'approche et les processus Scrum se concentrent principalement sur les contributions, outils et résultats obligatoires. En revanche, les Product Owners, Scrum Masters et autres praticiens Scrum expérimentés pourront aspirer à améliorer leurs connaissances grâce à l'ensemble des informations contenues dans ce chapitre. Il est également important de noter que bien que tous les processus ne soient définis qu'une seule fois dans le *Guide SBOK®*, ils ne sont pas nécessairement réalisés de façon séquentielle ou séparément. À certains moments, il peut s'avérer utile de combiner certains processus, en fonction des exigences spécifiques de chaque projet.

Ce chapitre est écrit depuis la perspective de l'équipe Scrum travaillant sur un sprint afin de produire des éléments potentiellement livrables, pouvant faire partie d'un plus gros projet, programme ou portefeuille. Des informations supplémentaires concernant le passage à l'échelle de Scrum pour les gros projets sont disponibles dans le chapitre 13 et celles sur le passage à l'échelle pour l'entreprise dans le chapitre 14.

La phase de lancement a lieu au début d'un projet Scrum. Au cours de cette phase, l'équipe centrale Scrum et les parties prenantes de l'entreprise sont identifiées, en commençant par le Product Owner, qui crée une vision du projet qui servira de guide tout au long du projet.

Sur la base de la vision du projet, un ensemble initial d'exigences est identifié et documenté sous la forme d'Epics.

Ces exigences initiales sont priorisées et utilisées pour créer un Product Backlog priorisé initial (c'est-à-dire le document listant les exigences dans un projet Scrum). Au cours de la dernière étape de la phase de lancement, un calendrier de planification de release est créé pour l'ensemble du projet. La phase de lancement ne produit pas un plan complet et détaillé pour l'ensemble du projet. Il n'est pas nécessaire de disposer d'un plan complet car des changements sont probables. Ils pourront facilement être intégrés dans un projet Scrum grâce au principe itératif incorporé dans les processus Scrum. L'objectif de la phase de lancement d'un projet Scrum est plutôt d'établir un bon plan initial pour le projet qui s'aligne sur les besoins de l'entreprise et/ou sur les règlements de priorité élevée. Cette phase doit être courte afin que la création de valeur puisse commencer le plus rapidement possible dans le projet.

La figure 8-1 offre un aperçu des processus de la phase d'initiation qui sont les suivants :

8.1 Création de la vision de projet — Dans ce processus, Le Product Owner est également identifié au cours de ce processus. Sur la base de l'analyse de rentabilité du projet, le Product Owner crée ensuite un énoncé de la vision du projet. Cet énoncé de la vision du projet indique l'orientation générale, l'inspiration et l'objectif du projet.

8.2 Identification du Scrum Master et des parties prenantes de l'entreprise — Dans ce processus, le Scrum Master sont identifiés grâce à des critères de sélection spécifiques qui peuvent permettre d'évaluer efficacement les compétences non techniques et les connaissances de Scrum nécessaires pour ce rôle important. En outre, les parties prenantes de l'entreprise sont également identifiés au cours de ce processus.

8.3 Formation de l'équipe Scrum — Dans ce processus, les membres de l'équipe Scrum sont identifiés sur la base des compétences requises pour terminer les produits livrables du projet, ainsi que des considérations relatives à la disponibilité, aux coûts et aux compétences générales importantes pour les membres d'une équipe Scrum. Généralement, c'est le Product Owner qui a la responsabilité principale de la sélection des membres de l'équipe mais il le fait souvent en collaboration avec le Scrum Master.

8.4 Développement des epics — Dans ce processus, l'énoncé de vision du projet sert de base au développement des epics, qui définissent les exigences de haut niveau pour le projet. Le Product Owner peut utiliser les réunions du groupe d'utilisateurs et d'autres outils pour collecter les exigences auprès des parties prenantes de l'entreprise.

8.5 Création d'un backlog de produit priorisé — Dans ce processus, les epics sont redéfinis, élaborés, et surtout priorisés en fonction de leur valeur commerciale respective afin de créer un backlog de produit priorisé pour le projet. La définition de terminé est égale à ces considérations ce moment-là. En outre, sur la base des recommandations du Scrum Guidance Body, le Product Owner et l'équipe Scrum établissent les critères « Done » (Terminé) pour le projet.

8.6 Planification de la release — Dans ce processus, le Product Owner, avec l'aide de l'équipe Scrum, élabore le calendrier initial de planification de release, qui est communiqué et partagé avec les parties prenantes de l'entreprise. Il est entendu que la nature itérative de Scrum peut nécessiter des ajustements futurs du calendrier de release. La durée des sprints est également déterminée au cours de ce processus.

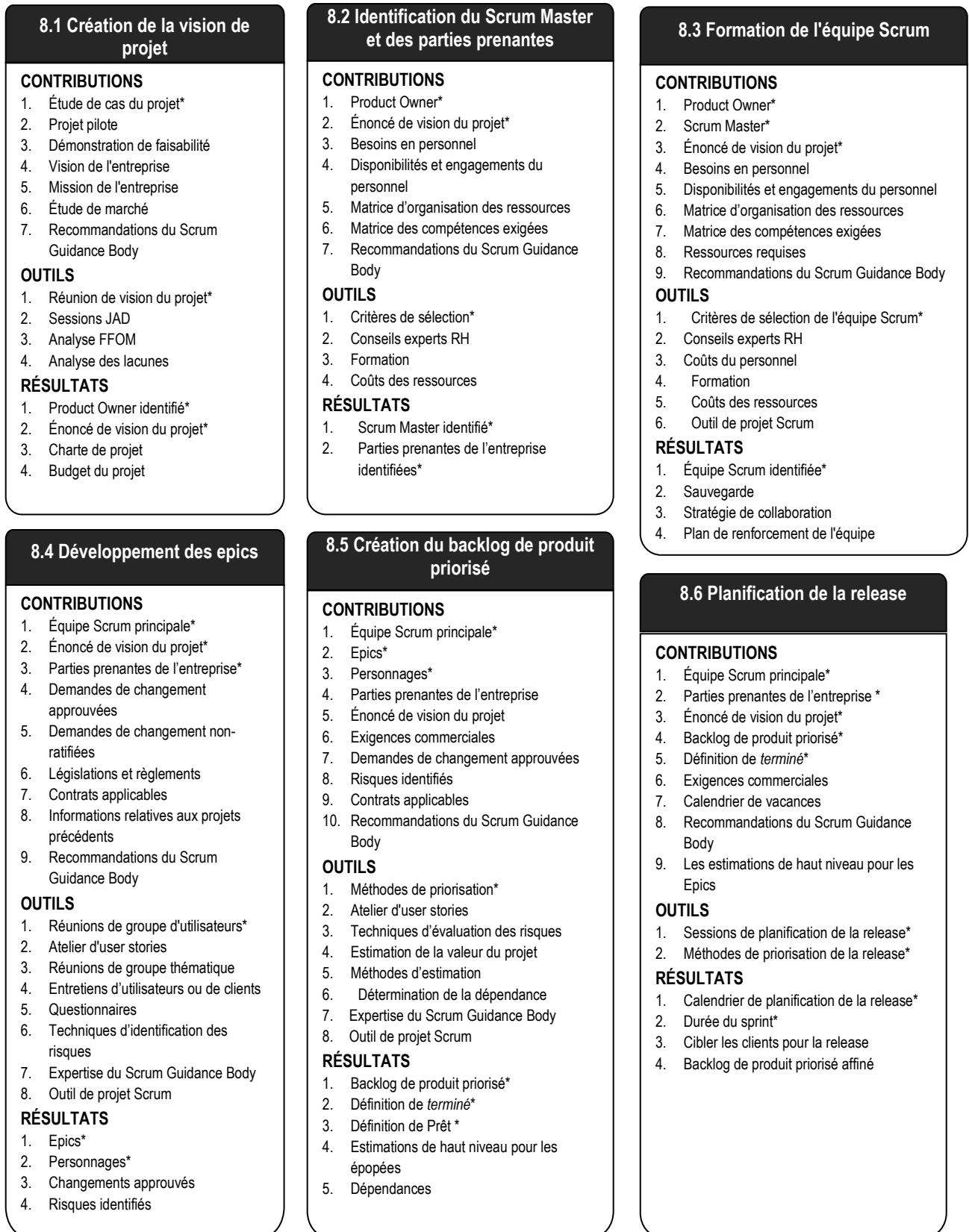


Figure 8-1 : Aperçu de la phase d'initiation

La figure 8-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de la phase d'initiation.

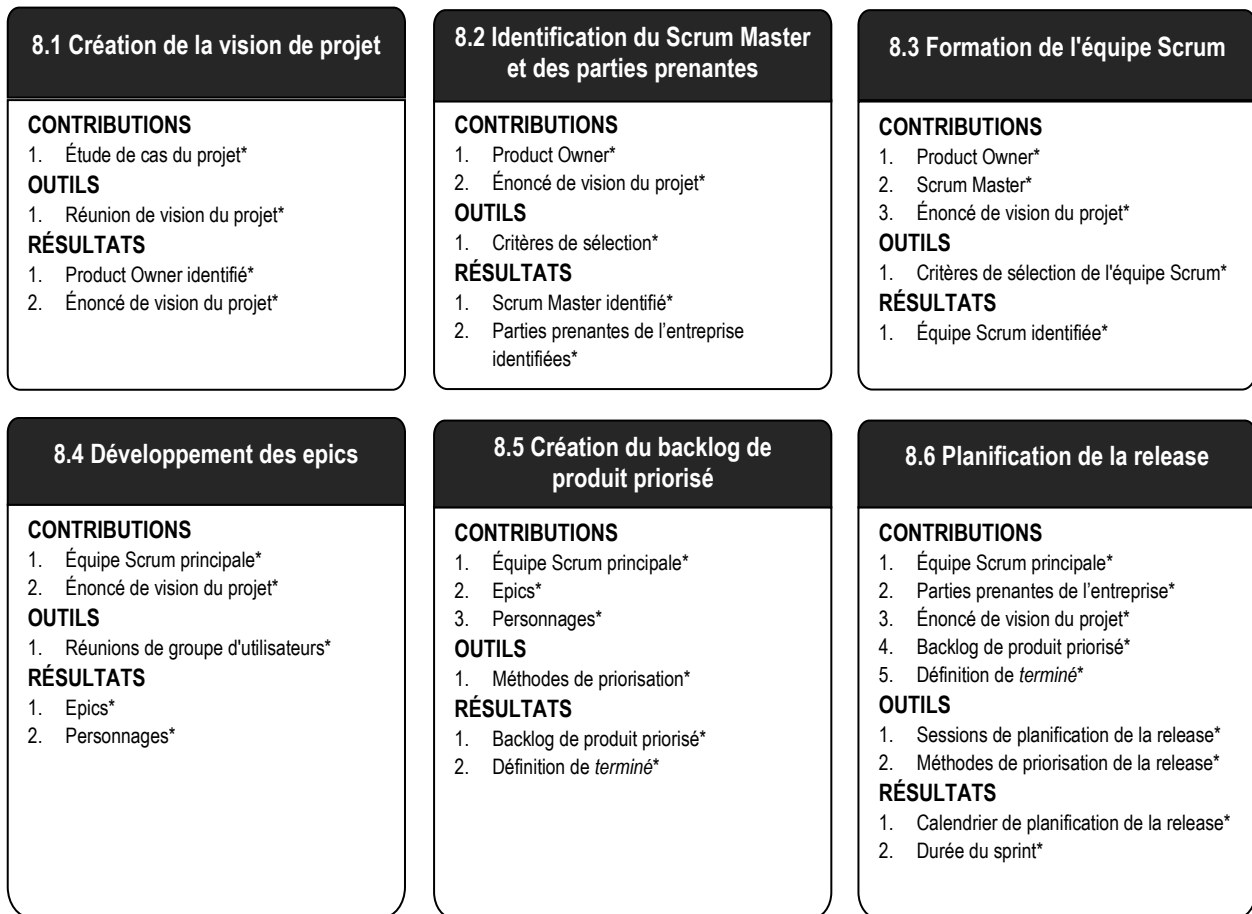


Figure 8-2 : Aperçu de la phase d'initiation (éléments essentiels)

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

8.1 Création de la vision de projet

Dans ce processus, le Product Owner est identifié. Sur la base de l'analyse de rentabilité du projet, le Product Owner crée ensuite un énoncé de la vision du projet. Cet énoncé de la vision du projet indique l'orientation générale, l'inspiration et l'objectif du projet.

La figure 8-3 illustre les contributions, outils et résultats du processus de *création de la vision de projet*.

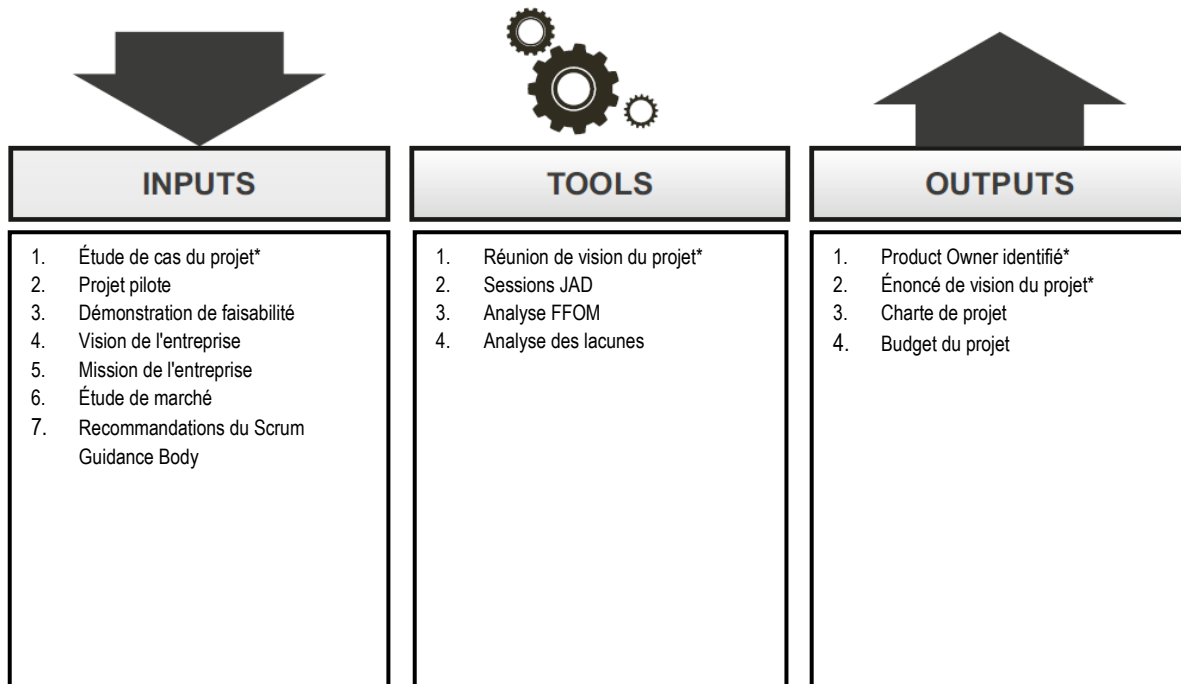


Figure 8-3 : Création de la vision de projet — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

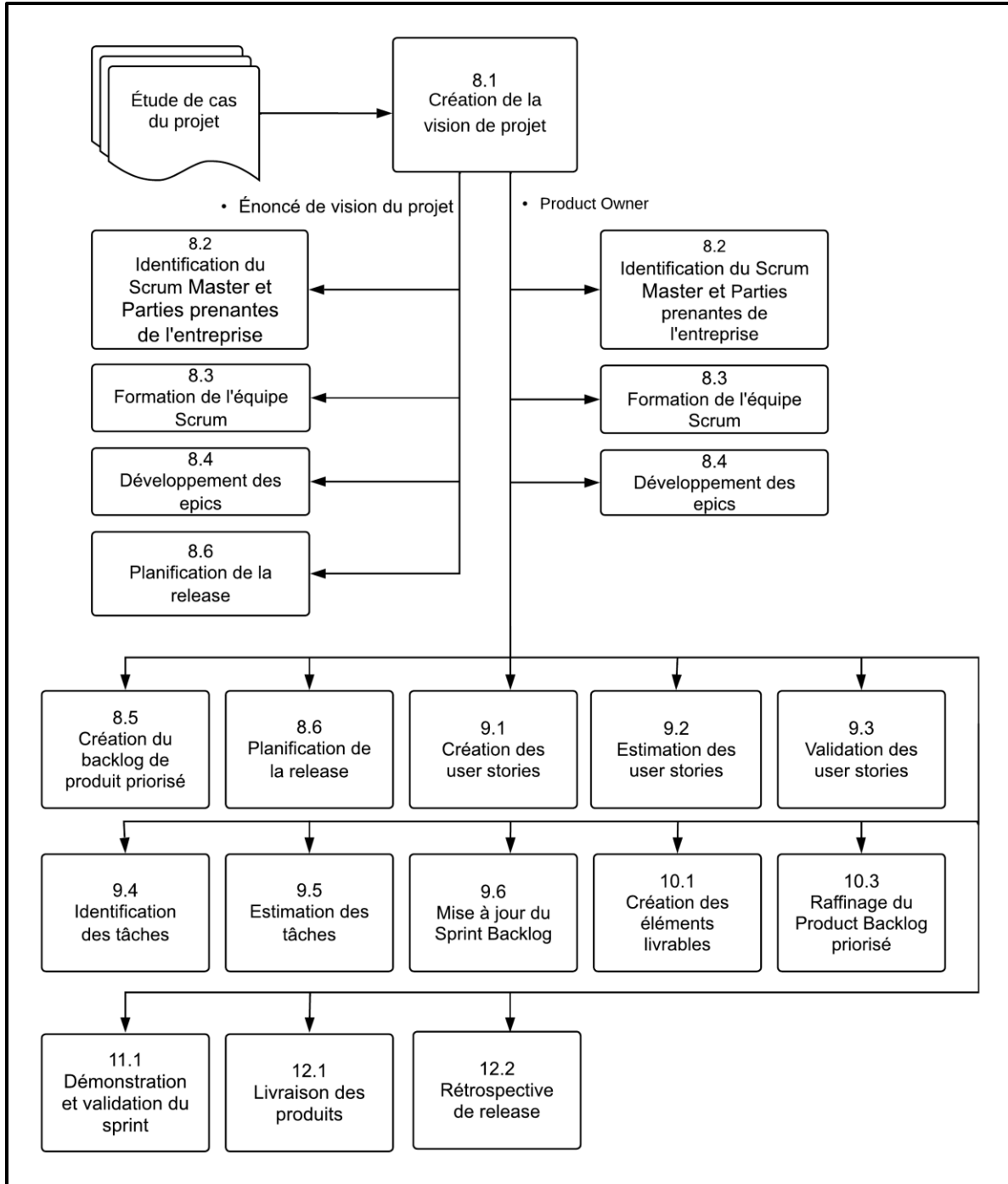


Figure 8-4 : Création de la vision de projet — diagramme de flux de données

8.1.1 Contributions

8.1.1.1 Étude de cas du projet*

Une étude de cas peut être un document structuré ou simplement une déclaration verbale qui justifie l'initiation d'un projet. Elle peut être formelle et complète ou informelle et brève. Quel que soit le format, elle inclut souvent des informations importantes concernant le contexte du projet, les objectifs commerciaux prévus et les résultats désirés, un rapport sur l'analyse FFOM et sur l'analyse des lacunes, une liste des risques identifiés ainsi qu'une estimation de la durée, des efforts et des coûts du projet. Analyse de rentabilité du projet doit également tenir compte de tous les facteurs externes pertinents, tels que les réglementations légales, les exigences gouvernementales, les préoccupations relatives à la confidentialité des données, etc.

8.1.1.2 Projet pilote

Si cela est faisable, une démonstration à petite échelle ou un projet pilote peuvent être réalisés afin de prévoir et d'évaluer la viabilité, la durée, les coûts, les risques et les effets potentiels du projet en lui-même. Cela aide à évaluer l'environnement pratique et oriente la conception du projet réel avant son initiation complète.

8.1.1.3 Démonstration de faisabilité

Une démonstration de faisabilité démontre et vérifie que l'idée du projet actuel est potentiellement viable dans le monde réel. Prenant souvent la forme d'un prototype, elle est destinée à déterminer la viabilité financière et technique, à aider à comprendre les exigences et à contribuer à l'évaluation des décisions de conception le plus tôt possible au cours du processus. Cependant, la démonstration de faisabilité n'a pas forcément besoin de représenter les éléments livrables réels du projet.

8.1.1.4 Vision de l'entreprise

Comprendre la vision de l'entreprise aide à maintenir l'axe du projet sur les objectifs de l'organisation et le futur potentiel de l'entreprise. Le Product Owner peut s'inspirer des conseils et directives de la vision de l'entreprise afin de créer l'énoncé de vision du projet.

8.1.1.5 Mission de l'entreprise

La mission de l'entreprise offre un cadre pour la formulation des stratégies d'une entreprise et guide le processus de décision global au sein de l'entreprise. La vision du projet doit être formulée de telle façon à ce que sa réalisation aide l'organisation à réaliser sa mission.

8.1.1.6 Étude de marché

Une étude de marché consiste en la recherche, la collecte, le rassemblement et l'analyse des données relatives aux préférences du client pour des produits. Elle inclut souvent des données importantes sur les tendances et la segmentation du marché ainsi que sur les procédés de marketing. Une étude de marché peut également inclure une analyse des concurrents. Elle offre ainsi une meilleure compréhension de leurs forces et faiblesses et peut aider les décideurs à formuler des produits mieux positionnés.

8.1.1.7 Recommandations du Scrum Guidance Body

Le Scrum Guidance Body (SGB) est un rôle facultatif. Il se compose généralement d'un groupe de documents et/ou d'un groupe d'experts qui sont normalement impliqués dans la définition des objectifs relatifs à la qualité, aux législations gouvernementales, à la sécurité et à d'autres paramètres organisationnels clés. Ces objectifs guident le travail effectué par le Product Owner, le Scrum Master et l'équipe Scrum. Le Scrum Guidance Body aide également à capter les meilleures pratiques qui peuvent être appliquées à tous les projets Scrum de l'organisation.

Le Scrum Guidance Body ne prend aucune décision concernant le projet en lui-même. Au contraire, il agit comme une structure de soutien ou de conseil à tous les niveaux hiérarchiques de l'organisation du projet — portefeuille, programme et projet. Lorsqu'elles en ont besoin, les équipes Scrum peuvent demander des conseils au Scrum Guidance Body.

Il est important de vérifier que la vision du projet est conforme aux recommandations fournies par le Scrum Guidance Body et que les processus respectent les normes et directives établies par celui-ci.

8.1.2 Outils

8.1.2.1 Réunion de vision du projet*

C'est une réunion entre les parties prenantes de l'entreprise, le Product Owner et le Scrum Master d'un programme. Elle aide à identifier le contexte commercial, les exigences commerciales et les attentes des parties prenantes de l'entreprise de façon à développer un énoncé de vision du projet efficace. L'approche Scrum prône la participation et la collaboration étroite avec tous les représentants commerciaux afin d'obtenir leur assentiment pour le projet et d'offrir une meilleure valeur ajoutée.

8.1.2.2 Sessions JAD

Une session Joint Application Design (JAD) est une technique permettant de déterminer des besoins. C'est un atelier facilité et très structuré qui accélère le processus de création de la vision de projet et permet aux parties prenantes de l'entreprise et aux autres décideurs de parvenir à un consensus sur la portée, les objectifs et autres spécifications du projet.

Elle consiste en des méthodes qui permettent d'encourager la participation des utilisateurs, d'accélérer le développement et d'améliorer les spécifications. Au sein d'un programme, les parties prenantes de l'entreprise concernées, le Product Owner et le Scrum Master peuvent se réunir afin d'esquisser et d'analyser les résultats commerciaux désirés et de visualiser leur vision pour le projet Scrum.

8.1.2.3 Analyse FFOM

L'analyse FFOM est une approche structurée de la planification de projet qui permet d'évaluer les **F**orces, **F**aiblesses, **O**pportunités et **M**enaces relatives à un projet. Ce type d'analyse permet d'identifier à la fois les facteurs internes et externes qui pourraient affecter un projet. Les forces et faiblesses sont des facteurs internes, tandis que les opportunités et les menaces sont des facteurs externes. L'identification de ces facteurs aide les parties prenantes de l'entreprise et décideurs à finaliser les processus, outils et techniques à utiliser afin de réaliser les objectifs du projet. Effectuer une analyse FFOM permet d'identifier les priorités, changements potentiels et risques le plus tôt possible.

8.1.2.4 Analyse des lacunes

L'analyse des lacunes est une technique utilisée afin de comparer l'état réel actuel avec l'état désiré. Dans une organisation, cela implique de déterminer et de documenter les différences entre les capacités commerciales actuelles et les capacités finales désirées. Normalement, un projet est initié afin d'offrir à l'organisation l'état désiré. Mener une analyse des lacunes peut donc aider les décideurs à déterminer le besoin pour un projet.

Les principales étapes associées à l'analyse des lacunes sont présentées dans la figure 8-5.

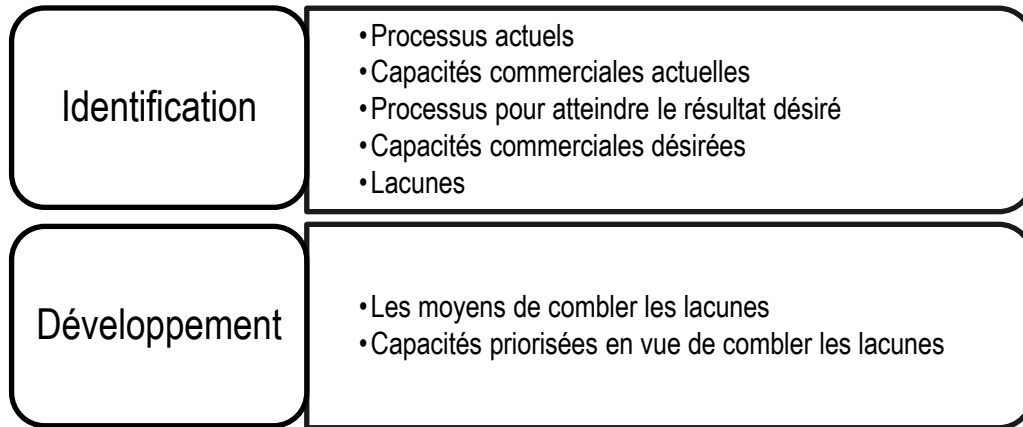


Figure 8-5 : Processus de l'analyse des lacunes

8.1.3 Résultats

8.1.3.1 Product Owner identifié*

Un des résultats de ce processus est l'identification du Product Owner. Le Product Owner est la personne responsable d'atteindre la valeur commerciale maximum du projet. Il est aussi chargé d'exprimer les exigences du client et de soutenir la justification commerciale du projet. Le Product Owner représente la voix du client.

Le rôle du Product Owner est décrit de façon plus détaillée dans la section 3.4.

8.1.3.2 Énoncé de vision du projet*

Le principal résultat du processus de *création de la vision de projet* est un énoncé de vision du projet structuré. Une bonne vision de projet explique les besoins commerciaux et les besoins auxquels le projet doit répondre mais pas comment ceux-ci seront remplis.

L'énoncé de vision du projet ne doit pas être trop spécifique et devrait laisser un certain degré de flexibilité. Il est possible que la compréhension actuelle du projet soit basée sur des suppositions qui changeront alors que le projet progresse, il est donc important que la vision du projet soit suffisamment flexible afin de tenir compte de ces changements. La vision du projet doit se concentrer sur le problème plutôt que sur la solution.

8.1.3.3 Charte de projet

La charte de projet est une déclaration officielle des objectifs et résultats désirés pour le projet. Dans beaucoup d'organisations, la charte de projet est le document qui autorise officiellement et formellement le projet et donne à l'équipe l'autorité nécessaire pour commencer le travail.

8.1.3.4 Budget du projet

Le budget du projet est un document financier qui inclut les coûts du personnel, des matériaux et autres dépenses associées à un projet. Normalement, le budget du projet est approuvé par le sponsor afin de garantir la disponibilité de fonds suffisants. Une fois approuvé, le Product Owner et le Scrum Master peuvent être régulièrement impliqués dans la gestion du budget du projet et garantir que les personnels et autres ressources requises pour les activités du projet sont disponibles.

8

8.2 Identification du Scrum Master et des parties prenantes de l'entreprise

Dans ce processus, le Scrum Master est identifié à l'aide de critères de sélection spécifiques qui permettent d'évaluer efficacement les compétences générales et les connaissances de Scrum nécessaires à ce rôle important. En outre, les *parties prenantes de l'entreprise* sont également identifiés au cours de ce processus.

La figure 8-6 illustre les contributions, outils et résultats du processus d'*identification du Scrum Master et des parties prenantes de l'entreprise*.

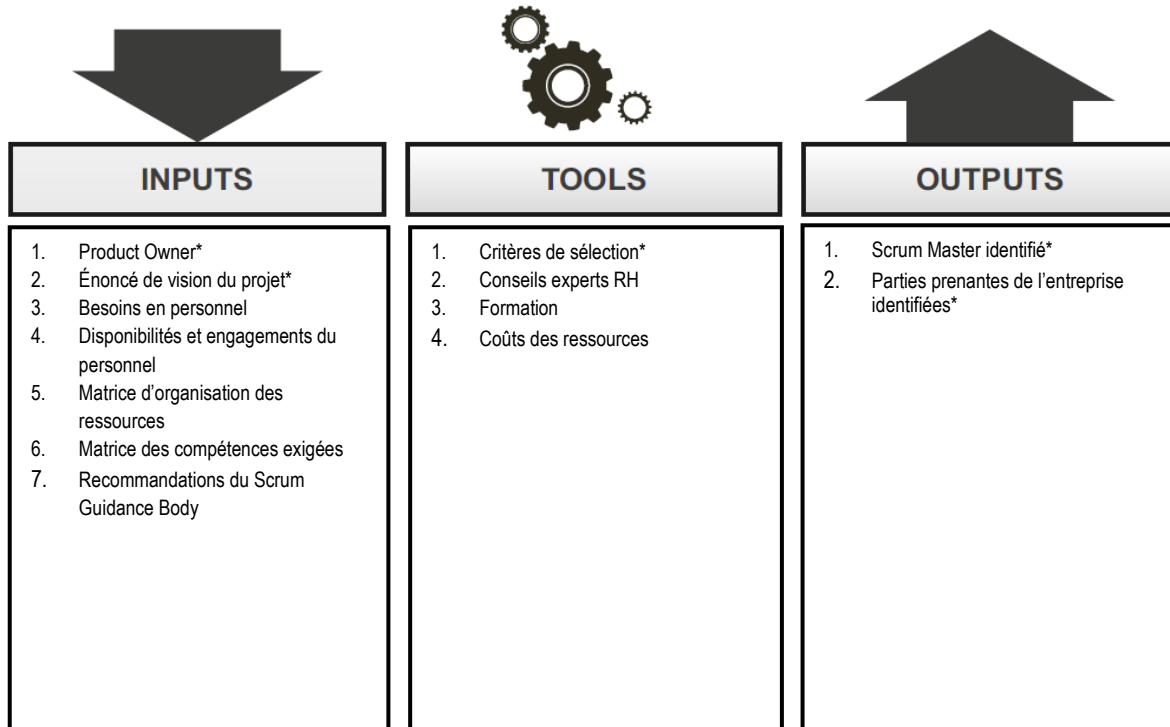


Figure 8-6 : Identification du Scrum Master et des parties prenantes de l'entreprise — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

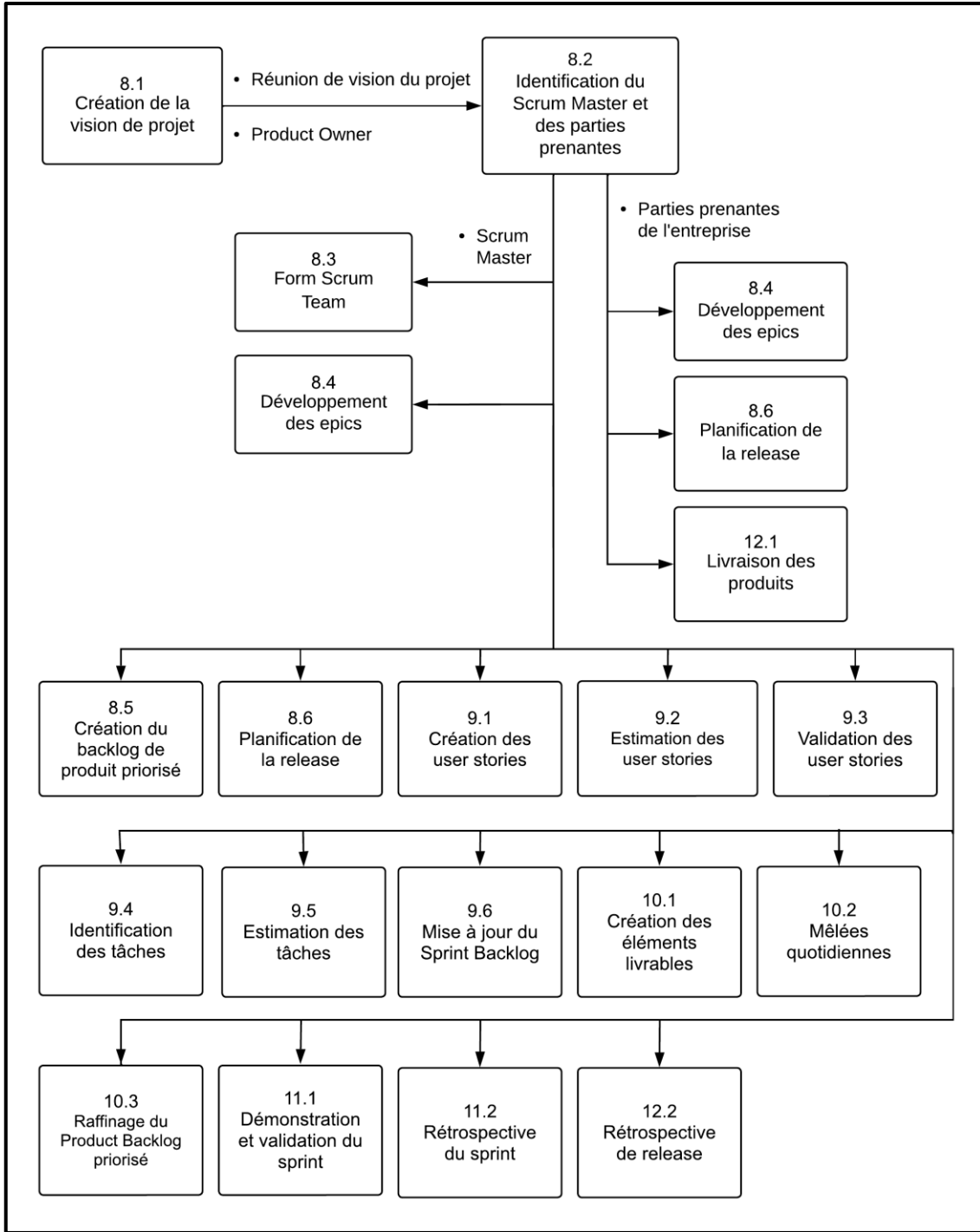


Figure 8-7 : Identification du Scrum Master et des parties prenantes de l'entreprise — Diagramme de flux de données

8.2.1 Contributions

8.2.1.1 Product Owner*

Décrit dans la section 8.1.3.1.

8.2.1.2 Énoncé de vision du projet*

Décrit dans la section 8.1.3.2.

8.2.1.3 Besoins en personnel

L'identification des besoins en termes de personnel est l'une des étapes initiales lors de la sélection du Scrum Master et des parties prenantes de l'entreprise . Il est important de documenter les rôles et responsabilités de tous ceux qui pourraient participer à la réalisation des tâches du projet. Habituellement, le Product Owner travaille avec le département des ressources humaines ou d'autres parties prenantes internes pertinentes pour déterminer et finaliser du role du Scrum Master.

Les principales parties prenantes de l'entreprise sont également identifiées et leur rôle dans le projet est déterminé.

8.2.1.4 Disponibilités et engagements du personnel

Avant de sélectionner le Scrum Master et parties prenantes de l'entreprise, leurs disponibilités doivent être confirmées. Seuls les membres de l'équipe qui seront disponibles et peuvent entièrement se consacrer au projet doivent être sélectionnés. Les disponibilités et engagements du personnel sont normalement illustrés sous la forme de calendriers montrant la disponibilité des ressources en termes de personnel pour toute la durée du projet.

En vue d'être efficaces, les équipes Scrum doivent idéalement compter entre six et dix membres. Remplacer des personnes ou changer les membres au sein de l'équipe Scrum principale n'est pas conseillé. Il est donc important d'avoir dans l'équipe Scrum principale des personnes disponibles et capables de se consacrer entièrement au projet.

8.2.1.5 **Matrice d'organisation des ressources**

La matrice d'organisation des ressources est une représentation hiérarchique combinant la structure d'organisation fonctionnelle et la structure organisationnelle projetée. La matrice d'organisation réunit les membres de l'équipe d'un projet venus de différents départements tels que l'informatique, la finance, le marketing, les ventes, la production et d'autres départements, pour créer des équipes polyvalentes.

Dans une matrice d'organisation, les membres de l'équipe remplissent deux objectifs — fonctionnel et projet. Les membres de l'équipe sont dirigés par le Product Owner concernant les activités associées au projet tandis que les gestionnaires fonctionnels réalisent les activités de gestion relatives à leur département telles que l'évaluation des performances et la validation des demandes de congés.

8

8.2.1.6 **Matrices des compétences exigées**

La matrice des compétences exigées, aussi connue sous le nom de cadre de compétences, est utilisée afin d'évaluer les lacunes de compétences et exigences de formation pour les membres de l'équipe. Une matrice des compétences établit les compétences, les capacités et le niveau d'intérêt des membres de l'équipe à utiliser ces compétences et capacités pour un projet. L'utilisation de cette matrice permet à l'organisation d'évaluer toute lacune de compétence chez les membres de l'équipe et d'identifier les employés qui auront besoin d'une formation complémentaire pour une compétence ou dans un domaine particulier.

8.2.1.7 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

8.2.2 Outils

8.2.2.1 Critères de sélection*

La sélection du Scrum Masters adéquat et l'identification des parties prenantes de l'entreprise correspondantes est essentiel à la réussite de tout projet. Dans certains projets, il peut y avoir des préconditions stipulant certains membres de l'équipe ainsi que leur rôle.

Lorsque la sélection du Scrum Masters est flexible, les éléments suivants sont d'importants critères de sélection :

1. *capacité à résoudre les problèmes* — c'est l'un des principaux critères à être pris en compte lors de la sélection du Scrum Master. Le Scrum Master doit avoir les compétences et l'expérience nécessaires afin d'aider à éliminer les obstacles pour l'équipe Scrum.
2. *disponibilité* — le Scrum Master doit être disponible afin de planifier, superviser et faciliter différentes réunions, notamment la réunion de planification de la release, les mêlées quotidiennes et les autres réunions relatives au sprint.
3. *responsabilité* — le Scrum Master doit être très responsable en vue de s'assurer que l'équipe Scrum dispose d'un environnement de travail favorable et de garantir la livraison réussie des projets Scrum.
4. *leader serviteur* — pour plus d'informations voir la section 3.10.4.1

Lors de l'identification des parties prenantes il est important de se rappeler que parties prenantes de l'entreprise incluent tous les clients, utilisateurs et sponsors qui interagissent régulièrement avec le Product Owner, le Scrum Master et l'équipe Scrum afin de contribuer et de faciliter la création des produits du projet. Parties prenantes de l'entreprise influencent le projet tout au long de son développement.

8.2.2.2 Conseils experts RH

Les conseils experts offerts par les responsables des ressources humaines peuvent être utiles afin d'identifier le Scrum Master et les parties prenantes de l'entreprise . Les RH possèdent des connaissances spécialisées sur les employés d'une organisation et sur les différentes techniques qui pourraient permettre d'identifier le Scrum Master et les parties prenantes de l'entreprise .

8.2.2.3 Formation

Scrum est une approche radicalement différente des autres méthodes traditionnelles de gestion de projet. Les membres de l'équipe peuvent ne pas toujours disposer des connaissances ou compétences nécessaires afin de travailler dans un environnement Scrum.

Le Product Owner doit évaluer les besoins en termes de formation des potentiels membres de l'équipe et faciliter les formations afin de combler les lacunes de connaissances au sein de l'équipe. Le Product Owner est normalement responsable de l'évaluation et de la sélection des membres de l'équipe, mais effectue souvent cette opération en consultant le Scrum Master qui pourrait disposer de connaissances supplémentaires au sujet du personnel pour avoir travaillé avec eux sur d'autres projets.

Les formations nécessaires devraient être offertes aux membres de l'équipe Scrum avant le début du projet et pendant qu'ils travaillent sur leur projet. Les membres de l'équipe Scrum doivent également être prêts à apprendre les uns des autres et des membres de l'équipe plus expérimentés.

8.2.2.4 Coûts des ressources

Un des principaux éléments à prendre en compte lors de la sélection du personnel concerne les concessions en termes d'expérience par rapport au salaire. Il existe également d'autres facteurs associés au personnel capables d'avoir un impact sur les coûts et qui doivent être pris en compte. Dans l'idéal, le Scrum Master, les membres de l'équipe et les parties prenantes de l'entreprise doivent être regroupés de façon à pouvoir communiquer fréquemment et facilement. Si le regroupement est impossible et que les équipes sont dispersées, des ressources additionnelles seront consacrées à la facilitation des communications, à la compréhension des différences culturelles, à la synchronisation du travail et à l'encouragement du partage des connaissances.

8.2.3 Résultats

8.2.3.1 Scrum Master identifié*

Un Scrum Master, c'est un facilitateur et leader serviteur qui s'assure que l'équipe Scrum dispose d'un environnement favorable à la réussite du projet. Le Scrum Master guide, facilite et enseigne les pratiques Scrum à toutes les personnes impliquées dans le projet. Il élimine les obstacles pour l'équipe et garantit le respect des processus Scrum. C'est le Product Owner qui est responsable d'identifier le Scrum Master d'un projet Scrum.

Le rôle du Scrum Master est décrit de façon plus détaillée dans la section 3.5.

8.2.3.2 Parties prenantes de l'entreprise identifiées*

Parties prenantes de l'entreprise (un terme collectif qui inclut les clients, les utilisateurs et les sponsors) interagissent fréquemment avec l'équipe Scrum principale et influencent le projet au cours du processus de développement du produit. Le projet produit des bénéfices collaboratifs pour parties prenantes de l'entreprise.

Le rôle des parties prenantes de l'entreprise est décrit dans la section 3.3.2.

8.3 Formation de l'équipe Scrum

Dans ce processus, les membres de l'équipe Scrum sont identifiés sur la base des compétences requises pour mener à bien les produits livrables du projet, ainsi que des considérations relatives à la disponibilité, aux coûts et aux compétences générales importantes pour les membres d'une équipe Scrum. Généralement, c'est le Product Owner qui a la responsabilité principale de la sélection des membres de l'équipe mais il le fait souvent en collaboration avec le Scrum Master.

La figure 8-8 illustre les contributions, outils et résultats du processus de *formation de l'équipe Scrum*.

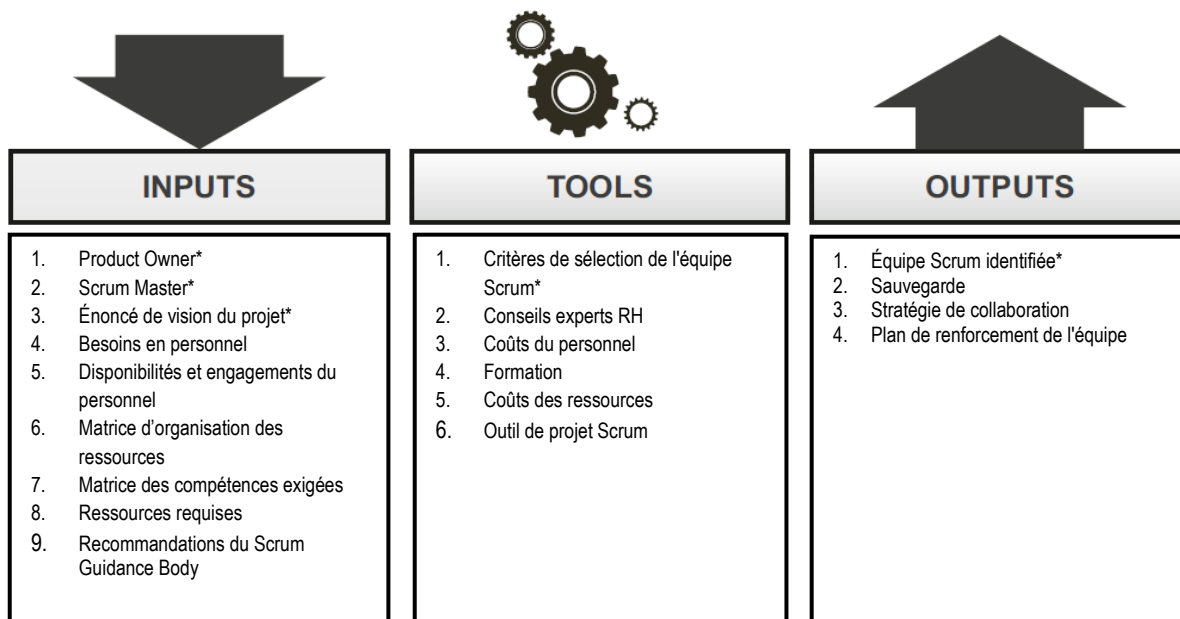


Figure 8-8 : Formation de l'équipe Scrum — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

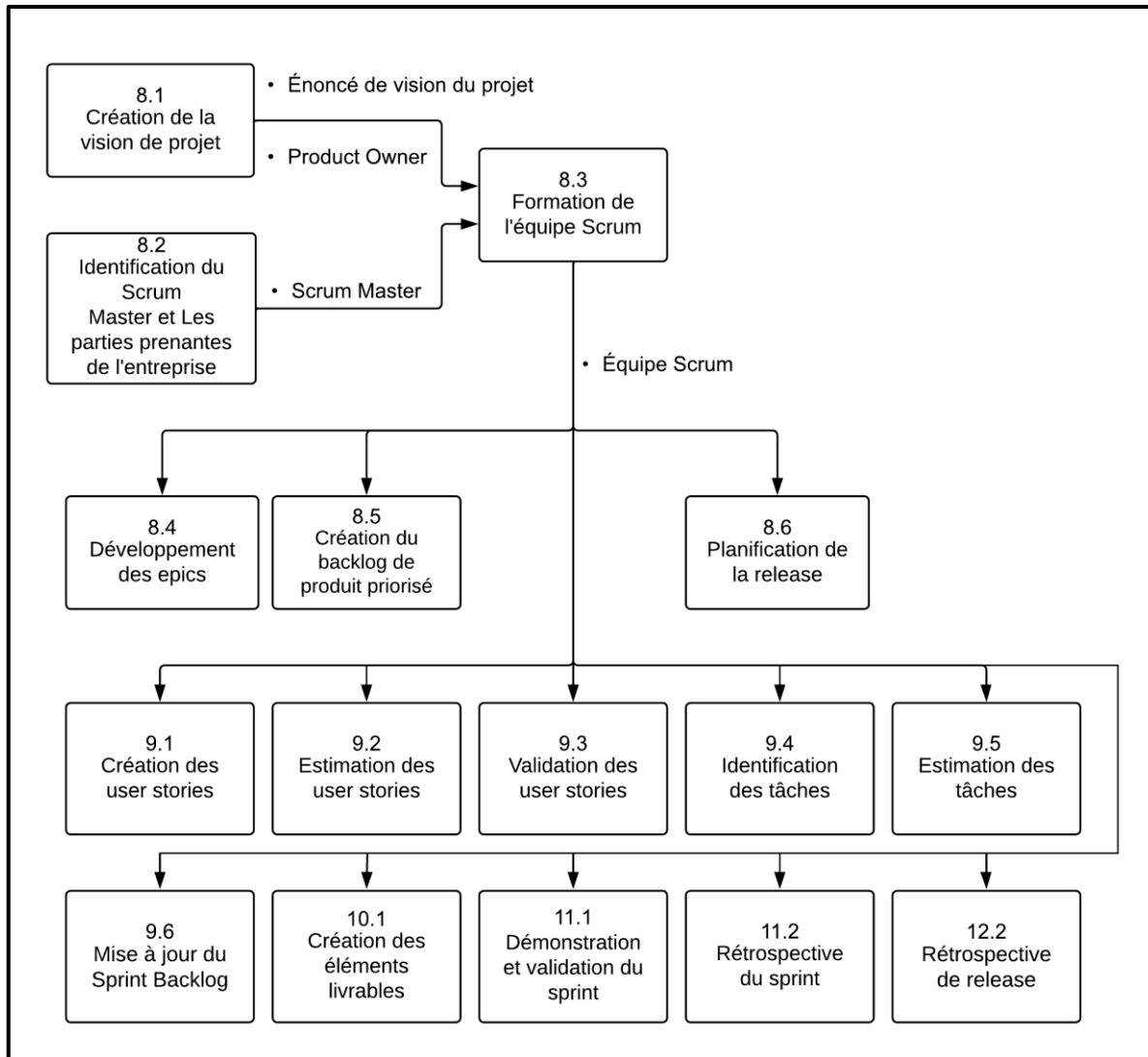


Figure 8-9 : Formation de l'équipe Scrum — Diagramme de flux de données

8.3.1 Contributions

8.3.1.1 Product Owner*

Décrit dans la section 8.1.3.1.

8.3.1.2 Scrum Master*

Décrit dans la section 8.2.3.1.

8.3.1.3 **Énoncé de vision du projet***

Décrit dans la section 8.1.3.2.

8.3.1.4 **Besoins en personnel**

Décrits dans la section 8.2.1.8.

8.3.1.5 **Disponibilités et engagements du personnel**

Décrits dans la section 8.2.1.9.

8.3.1.6 **Matrice d'organisation des ressources**

Décrite dans la section 8.2.1.10.

8.3.1.7 **Matrice des compétences exigées**

Décrite dans la section 8.2.1.11.

8.3.1.8 **Ressources requises**

Ces exigences incluent toutes les ressources - autres qu'en termes de personnels - requises afin que l'équipe Scrum fonctionne efficacement. Des exemples de ces ressources incluent notamment le matériel de bureau, les espaces de réunions, les équipements de travail et les scrumboards. Dans le cas d'équipes virtuelles, des ressources additionnelles telles que des outils de collaboration, la vidéoconférence, des répertoires de documents partagés et des services de traduction doivent être pris en compte.

8.3.1.9 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

8.3.2 Outils

8.3.2.1 Critères de sélection de l'équipe Scrum*

L'équipe Scrum est le noyau de tout projet Scrum et choisir les bons membres pour l'équipe est essentiel à la réussite des projets Scrum. Les membres de l'équipe Scrum sont des généralistes/spécialistes dans le sens où ils disposent de connaissances dans différents domaines et sont experts pour au moins l'un d'entre eux. En plus de leur expertise du sujet, c'est le savoir-être des membres de l'équipe qui détermine la réussite des équipes autoorganisées.

Les membres idéaux d'une équipe Scrum sont indépendants, motivés, axés sur le client, responsables et collaboratifs. L'équipe devrait être capable d'entretenir un environnement de réflexion indépendante et de décisions de groupe de façon à tirer les meilleurs avantages de sa structure.

8.3.2.2 Conseils experts RH

Les conseils experts des responsables des ressources humaines (RH) peuvent être précieux lors de la formation d'une équipe Scrum. Les RH disposent de connaissances spécialisées au sujet des employés d'une organisation et des différentes techniques pouvant être utiles aux Product Owners, Scrum Masters et sponsors afin d'identifier les bons membres pour l'équipe.

8.3.2.3 Coûts du personnel

Tous les coûts relatifs aux besoins en termes de personnel doivent être évalués, analysés, approuvés et budgétisés.

8.3.2.4 Formation

Les membres de l'équipe peuvent ne pas disposer des compétences et connaissances requises afin de réaliser des tâches spécialisées. Le Product Owner doit évaluer les besoins en termes de formation des potentiels membres de l'équipe et fournir les formations nécessaires lorsque des lacunes en termes de compétences ou de connaissances sont identifiées.

Pour une mise en œuvre réellement efficace de l'approche Scrum, l'organisation doit disposer de connaissances importantes en termes de principes et valeurs Scrum. Ces connaissances contribueront à la mise en œuvre réussie de Scrum. L'équipe Scrum doit être sensibilisée et formée aux pratiques Scrum et le Scrum Master doit jouer le rôle de coach pour l'équipe. Étant donné que la planification des sprints est un facteur de réussite important, la formation aidera l'équipe à comprendre comment discuter et identifier des objectifs de sprint réalisables.

Le Scrum Master doit amener l'équipe Scrum à faire de son mieux en motivant les membres de l'équipe et en facilitant le processus de développement. En formant et en coachant les membres de l'équipe, le Scrum Master peut les aider à exprimer les problèmes et défis auxquels ils font face. Normalement, tous les problèmes ou conflits rencontrés au sein de l'équipe sont résolus par l'équipe avec l'aide du Scrum Master lorsque cela est nécessaire. Le Scrum Master doit résoudre les problèmes tels que la baisse du moral ou le manque de coordination au sein de l'équipe. Il est responsable de l'élimination des obstacles rencontrés par l'équipe. Lorsque cela est nécessaire, le Scrum Master peut faire remonter aux responsables les problèmes et obstacles externes en vue de leur résolution ou de leur élimination.

La formation et les coûts de formation sont également abordés dans le processus d'*identification du Scrum Master et des parties prenantes de l'entreprise* dans la section 8.2.2.3

8.3.2.5 Coûts des ressources

Tous les coûts relatifs aux exigences qui ne sont pas associées au personnel doivent être évalués, analysés, approuvés et budgétisés. Dans l'environnement d'un projet, une ressource est tout ce qui est utilisé afin de réaliser une tâche ou activité, y compris notamment l'équipement, les matériaux, les services extérieurs et l'espace physique.

8.3.2.6 Outil de projet Scrum

Décrites dans la section 2.5.3.1

8.3.3 Résultats

8.3.3.1 Équipe Scrum identifiée*

L'équipe Scrum, parfois également appelée équipe de développement, est le groupe ou l'équipe responsable de l'interprétation des exigences commerciales spécifiées par le Product Owner, de l'estimation des user stories et de la création finale des produits du projet. Les équipes Scrum sont polyvalentes et autoorganisées. L'équipe décide de la quantité de travail sur lequel elle s'engage pour un sprint et détermine la meilleure façon d'effectuer le travail. L'équipe Scrum est formée de membres polyvalents qui effectuent tout le travail impliqué par la création d'éléments potentiellement livrables y compris le développement, le test, l'assurance qualité, etc.

C'est le Product Owner, souvent avec l'aide du Scrum Master, qui est responsable d'identifier l'équipe Scrum.

Le rôle de l'équipe Scrum est décrit de façon plus détaillée dans la section 3.6.

8.3.3.2 Sauvegardes

Lors de la sélection des membres de l'équipe, il est important d'identifier des remplaçants pour les compétences critiques, de préférence au sein de la même équipe Scrum.

8.3.3.3 Stratégie de collaboration

La collaboration est un élément très important dans l'approche Scrum. Il est essentiel de planifier la façon dont les différents décideurs, parties prenantes de l'entreprise et membres de l'équipe participent et collaborent entre eux. La stratégie de collaboration est un résultat facultatif qui peut être formel ou informel. Puisque Scrum tend à éviter toute documentation superflue, la stratégie de collaboration peut parfois être un simple accord oral entre les différentes parties prenantes de l'entreprise. Cependant, pour les projets plus larges et complexes, particulièrement ceux avec des équipes dispersées, un accord plus formel pourra s'avérer nécessaire. Cette stratégie peut aborder la façon dont les membres de l'équipe Scrum principale, parties prenantes de l'entreprise et autres personnes impliquées dans le projet Scrum communiquent et collaborent tout au long du projet. Elle peut également définir des outils ou techniques spécifiques à utiliser dans ce but. Par exemple, dans le cas d'équipes dispersées, il peut y avoir besoin d'un accord sur la date et la façon dont les réunions seront conduites, le type d'outils de communication à utiliser, et les personnes devant participer à chaque type de réunion.

8.3.3.4 Plan de renforcement de l'équipe

Puisqu'une équipe Scrum est polyvalente, chaque membre doit participer activement à tous les aspects du projet. Le Scrum Master doit pouvoir identifier les problèmes auxquels les membres de l'équipe font face et doit les résoudre consciencieusement afin de maintenir une équipe efficace.

Afin de renforcer la cohésion de l'équipe, le Scrum Master doit s'assurer que les relations entre les membres de l'équipe sont positives et que les membres de l'équipe sont unifiés afin d'atteindre les objectifs globaux du projet et les objectifs de l'organisation, menant ainsi à une efficacité et à une productivité améliorée.

Dans ce contexte, il est essentiel d'étudier la section 3.10 qui traite des théories RH populaires et de leur importance dans Scrum.

8.4 Développement des epics

Dans ce processus, l'énoncé de la vision du projet sert de base au développement des Epics, qui définissent les exigences de haut niveau du projet. Le Product Owner peut utiliser les réunions du groupe d'utilisateurs et d'autres outils pour collecter les exigences auprès des parties prenantes de l'entreprise.

La figure 8-10 illustre les contributions, outils et résultats du processus de *développement des epics*.

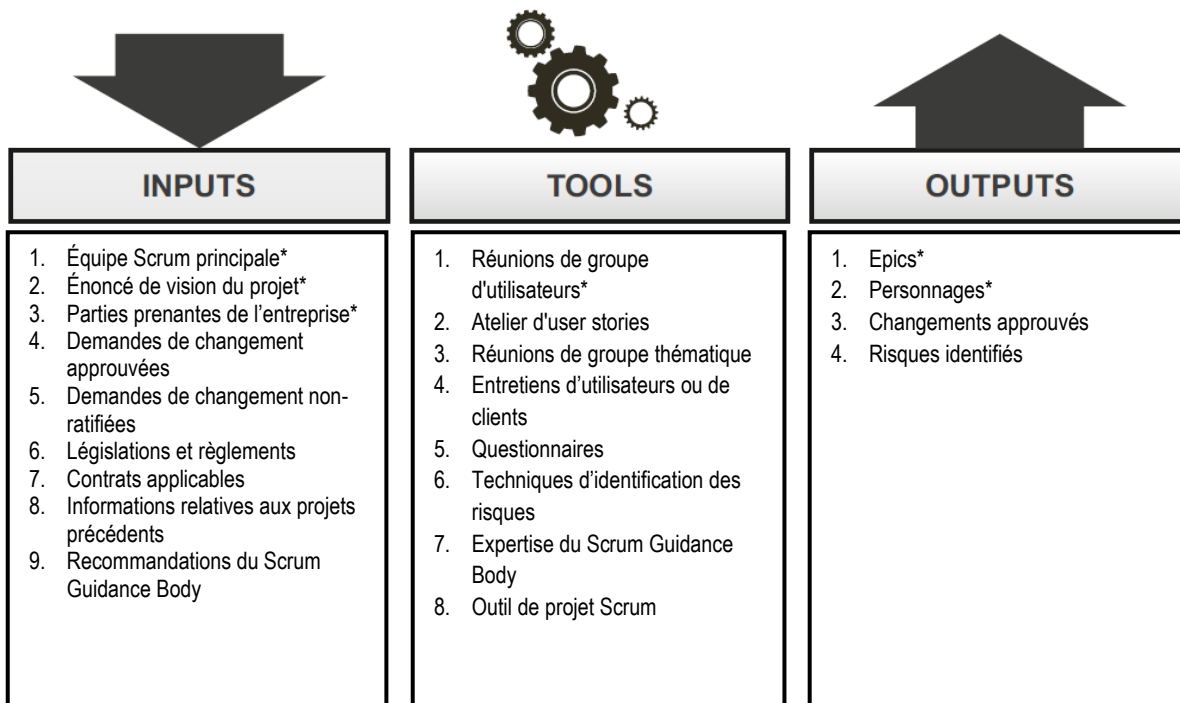


Figure 8-10 : Développement des epics — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

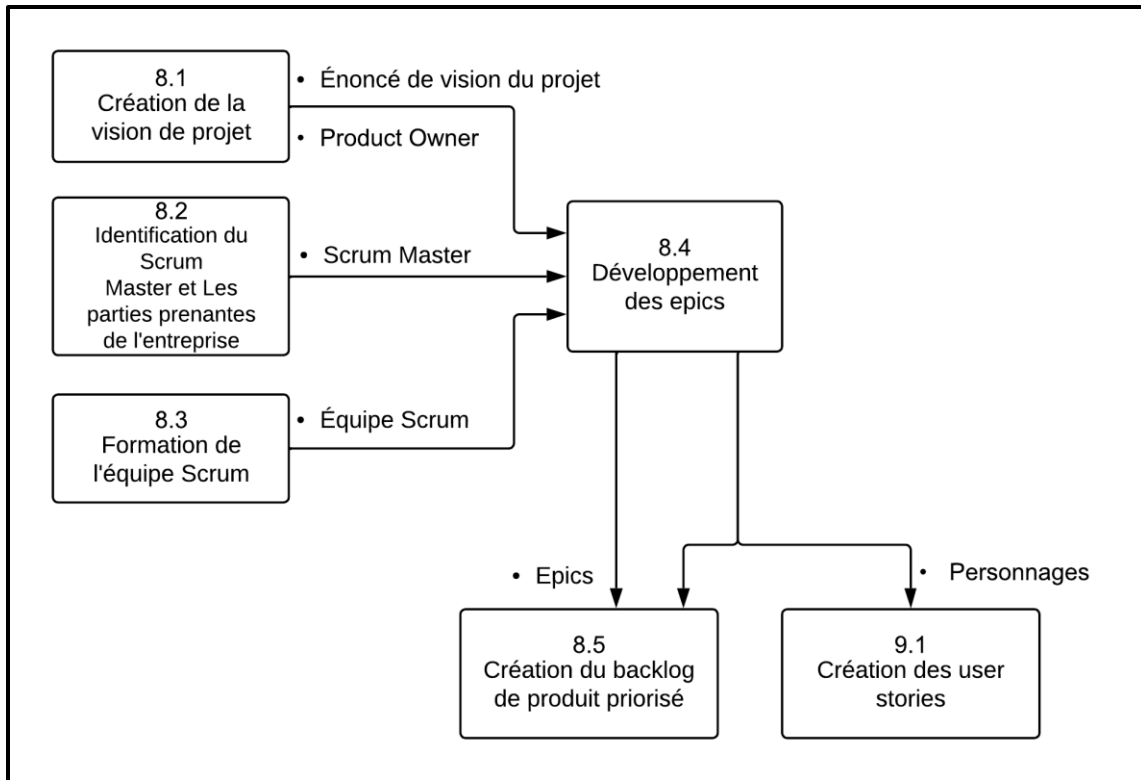


Figure 8-11 : Développement des epics — Diagramme de flux de données

8.4.1 Contributions

8.4.1.1 Équipe Scrum principale*

L'équipe Scrum principale telle qu'elle est décrite dans la section 3.3.1. se compose de l'équipe Scrum, du Scrum Master et du Product Owner.

8.4.1.2 Énoncé de vision du projet*

Décrit dans la section 8.1.3.2.

8.4.1.3 Parties prenantes de l'entreprise

Décrites dans la section 8.2.3.2.

8.4.1.4 **Backlog de produit du programme**

Décrit dans la section 8.1.1.5.

8.4.1.5 **Demandes de changement approuvées**

Les demandes de changement approuvées provenant du programme ou du portefeuille constituent des contributions qui doivent être intégrées à la liste des changements approuvés du projet afin d'être mises en œuvre lors des sprints à venir. Chaque changement peut nécessiter son propre epic ou sa propre user story et peut devenir une contribution dans le processus de *développement des epics*. Les demandes de changement approuvées dans ce processus peuvent également être le résultat d'autres processus Scrum.

Les demandes de changement et demandes de changement approuvées sont abordées dans les sections 6.3.1, 6.4.2.1 et 6.6.

8.4.1.6 **Demandes de changement non-ratifiées**

Les requêtes de modification sont normalement présentées sous la forme de demandes de changement et restent non-ratifiées jusqu'à ce qu'elles soient officiellement approuvées. Les demandes de changement non-ratifiées dans le processus de *développement des epics* peuvent provenir de la *création des éléments livrables*, des *mêlées quotidiennes* ou d'autres processus.

Les demandes de changement et demandes de changement non-ratifiées sont abordées dans les sections 6.3.1, 6.4.2.1 et 6.6.

8.4.1.7 **Risques du programme et du portefeuille**

Les risques relatifs à un portefeuille ou à un programme auront également un impact sur les projets faisant partie du portefeuille ou du programme en question. Lors de l'évaluation des risques d'un portefeuille ou d'un programme, s'il est établi qu'un risque pourrait affecter un projet individuel, les informations pertinentes concernant le risque doivent être communiquées au Product Owner et à l'équipe Scrum. Les risques relatifs aux programmes et portefeuilles peuvent contribuer au processus de *développement des epics* et avoir un impact global sur la façon dont ce processus est mené.

Les risques associés aux programmes et portefeuilles sont décrits dans la section 7.5.1.

8.4.1.8 Législations et règlements

Selon le projet, il peut exister des législations et règlements, imposés par les instances dirigeantes, susceptibles d'avoir un impact sur la planification et l'exécution du projet. Les législations sont externes à l'organisation et imposées par un organisme gouvernemental. Les règlements peuvent être soit internes soit externes. Les règlements internes sont ceux qui s'appliquent au sein de l'organisation, normalement sur la base de politiques d'entreprise. Ces règlements peuvent être associés aux systèmes de gestion de la qualité, aux règlements du personnel, etc. Les règlements externes sont ceux qui ont trait à des standards, normes ou exigences gouvernementaux.

Les législations et règlements doivent être pris en compte lors du développement des epics. Les epics sont basés sur les exigences commerciales. Afin de répondre à ces exigences, l'équipe du projet doit respecter les législations et règlements internes et externes.

Parfois, certains des règlements et législations impactant plusieurs projets Scrum peuvent être inclus dans les recommandations du Scrum Guidance Body (voir section 8.1.1.11).

8.4.1.9 Contrats applicables

Si le projet dans son entier ou des parties de celui-ci sont réalisés sous contrat, le contrat définit la portée du travail et les termes spécifiques du contrat. Le type de contrat utilisé influence le niveau de risque du projet.

Les types de contrats les plus utilisés dans les projets Scrum incluent notamment :

Le contrat de livraison graduelle — ce contrat inclut des points d'inspection à intervalles réguliers. Cela aide le client et parties prenantes de l'entreprise à prendre des décisions périodiques au sujet du développement du produit pendant le projet à chaque point d'inspection. Le client peut soit accepter le développement du produit, décider d'interrompre le développement du produit ou demander des modifications.

Le contrat de "joint-venture » — ce contrat est normalement utilisé lorsque deux ou plusieurs parties s'associent afin de réaliser le travail d'un projet. Les parties impliquées dans le projet réaliseront toutes deux un retour sur investissement car les revenus ou bénéfices générés seront partagés entre les parties.

Le contrat de développement par palier — ce contrat libère des fonds chaque mois ou chaque trimestre après qu'une release a été réalisée avec succès. Cela fournit une incitation à la fois au client et au fournisseur tout en limitant les risques financiers pour le client sur une période de temps définie puisque les releases infructueuses ne sont pas payées.

Le contrat d'incitation et de pénalité — ce contrat est basé sur l'acceptation du fait que le fournisseur sera récompensé avec une incitation financière si les produits du projet sont livrés à temps, mais entraînera une pénalité financière si la livraison est retardée.

D'autres types de contrats populaires incluent le contrat de paiement par fonctionnalité, les contrats de temps et de matériaux, les contrats à prix et protée fixes, et les contrats à bénéfices fixes.

Les epics doivent être développées en gardant à l'esprit les termes et conditions du contrat utilisé.

8.4.1.10 Informations relatives aux projets précédents

Les informations et idées acquises lors de projets précédents similaires au sein de l'organisation sont des contributions précieuses pour le développement des epics et l'évaluation des risques. Les informations relatives aux projets précédents peuvent inclure les notes des chefs de projet, les registres des projets et les commentaires des parties prenantes de l'entreprise.

Certaines informations et meilleures pratiques relatives à des projets précédents peuvent être disponibles via les recommandations du Scrum Guidance Body.

8.4.1.11 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Les recommandations du Scrum Guidance Body peuvent inclure des informations concernant les règles, règlements, normes et meilleures pratiques pour le développement des epics.

8.4.2 Outils

8.4.2.1 Réunions de groupe d'utilisateurs*

Les réunions de groupes d'utilisateurs impliquent parties prenantes de l'entreprise concernées (principalement les utilisateurs ou clients du produit). Les membres concernés de l'équipe centrale Scrum peuvent également être présents lors des réunions du groupe d'utilisateurs. Cela aide à formuler les critères d'acceptation du produit et fournit des informations importantes pour le développement des epics. Les réunions de groupes d'utilisateurs sont essentielles afin d'éviter des retouches coûteuses qui pourraient résulter en un manque de clarté en termes d'attentes et d'exigences. Ces réunions permettent également de promouvoir l'appropriation du projet et de créer une compréhension commune entre l'équipe Scrum principale et parties prenantes de l'entreprise concernées.

8.4.2.2 Atelier d'user stories

Des ateliers d'user stories ont lieu au cours du processus de *développement des epics*. Le Scrum Master facilite ces sessions auxquelles toute l'équipe Scrum principale participe. Il est parfois également souhaitable d'inclure d'autres parties prenantes de l'entreprise. Ces ateliers aident le Product Owner à prioriser les exigences et permettent à l'équipe Scrum principale d'avoir une vision partagée des critères d'acceptation. Ils garantissent que les epics et user stories décrivent les fonctionnalités du point de vue des utilisateurs, sont faciles à comprendre et peuvent être estimés avec fiabilité. Les ateliers d'user stories sont utiles afin de comprendre les attentes des utilisateurs concernant les éléments livrables et sont une excellente opportunité pour encourager la consolidation de l'équipe. Ils facilitent également la préparation en vue de la planification du prochain sprint. Un atelier d'user stories est une plateforme idéale pour discuter et clarifier chaque élément d'un produit et creusent souvent les plus petits détails afin d'en garantir la clarté.

8.4.2.3 Réunions de groupe thématique

Les groupes thématiques rassemblent des personnes lors d'une séance guidée pour partager leurs opinions, perceptions ou notes sur un produit, service ou un résultat désiré. Les membres du groupe thématique sont libres de se poser des questions et de demander des clarifications sur certains sujets ou concepts. Les questionnements, critiques constructives et feedbacks permettent aux groupes thématiques d'atteindre une meilleure qualité de produit et contribuent donc à répondre aux attentes des clients. Lors de ces réunions, les membres du groupe thématique peuvent parvenir à un consensus dans certains domaines, tandis que leurs opinions pourront être différentes dans d'autres. Lorsque les membres du groupe ont des opinions ou perspectives différentes, tous les efforts sont faits afin de résoudre les différends de façon à atteindre un consensus.

Les sessions de groupe thématique peuvent aider les équipes à trouver des idées innovantes, résoudre des problèmes et exprimer des suggestions d'amélioration. Ces réunions facilitent la recherche des faits et génèrent des idées et des feedbacks de la part des utilisateurs potentiels et des développeurs du produit. Ces réunions sont généralement menées afin de planifier, d'évaluer ou d'améliorer un produit ou service. Les perspectives obtenues lors de ces réunions peuvent également contribuer au développement des epics et user stories. Les réunions de groupe thématique sont parfois organisées afin de résoudre des problèmes qui pourraient survenir lors du développement des epics.

8.4.2.4 Entretiens d'utilisateurs ou de clients

Il est important de faire participer parties prenantes de l'entreprise (y compris le sponsor, les utilisateurs et les clients du produit) afin d'obtenir les informations contextuelles et perspectives nécessaires au développement des epics. Passer du temps à effectuer des entretiens de qualité avec les utilisateurs et clients permet de garantir que les exigences des epics correspondent à la vision globale du projet, offrant donc une meilleure valeur ajoutée.

Ces entretiens permettent :

- d'identifier et de comprendre les besoins et attentes des parties prenantes de l'entreprise
- de recueillir des opinions et des faits
- de comprendre la perspective des parties prenantes de l'entreprise au sujet du produit fini
- de recueillir du feedback concernant les produits itératifs ou partiellement développés
- Obtenir l'adhésion et l'engagement des utilisateurs et/ou des clients

8.4.2.5 Questionnaires

Une façon rentable d'obtenir des informations statistiques quantitatives et qualitatives d'un large panel d'utilisateurs ou de clients consiste à utiliser des sondages ou questionnaires. Un questionnaire est un instrument de recherche qui contient les questions à poser à une personne interrogée de façon à obtenir des informations relatives à un problème ou sujet spécifique. Les questionnaires peuvent être auto-administrés ou administrés par un enquêteur.

Il est important de faire très attention à la conception des questionnaires, au choix du public cible et au choix de la méthode de sondage appropriée afin d'éviter les erreurs et la partialité.

Lors du développement des epics, le Product Owner ou le Scrum Master pourra mener un sondage afin de recueillir des informations pertinentes de la part des parties prenantes de l'entreprise ou de l'équipe Scrum.

8.4.2.6 Techniques d'identification des risques

Décrites dans la section 7.4.1.1.

8.4.2.7 Expertise du Scrum Guidance Body

Décrite dans la section 3.3.2.

Lors de la création des epics, l'expertise du Scrum Guidance Body fait référence aux règles et règlements documentés ou aux standards et meilleures pratiques pour la création des epics. Il peut également y avoir une équipe d'experts sur un sujet, capable d'assister le Product Owner pour la création des epics. Cette équipe peut inclure des analystes commerciaux, des architectes principaux, des développeurs expérimentés, des experts Scrum et d'autres personnes expérimentées. Ce groupe d'experts n'est normalement pas l'équipe qui travaillera au développement du projet en particulier. Ces experts tendent à naviguer de projet en projet durant la « phase de vente » ou « phase zéro » avec les clients ou utilisateurs.

8.4.2.8 Outil de projet Scrum

Décrites dans la section 2.5.3.1.

8.4.3 Résultats

8.4.3.1 Epics*

Les epics sont écrits durant la phase initiale du projet lorsque la majorité des user stories sont des fonctionnalités de haut niveau ou que les descriptions et exigences du produit sont relativement larges. Les Epics sont de grandes User Stories brutes dans le Product Backlog priorisé. Les Epics aident le Product Owner et les parties prenantes pertinentes de l'entreprise à planifier les releases, à hiérarchiser les exigences de haut niveau et à établir une feuille de route globale pour le projet.

Au fur et à mesure que le Product Owner obtient plus de clarté sur les besoins des utilisateurs, ces Epics sont ensuite décomposés en User Stories plus petites et plus granulaires. Ainsi, les Epics contiennent une ou plusieurs User Stories. Ces petites user stories correspondent normalement à des fonctionnalités ou blocs de tâches simples, courts et faciles à mettre en œuvre qui doivent être réalisés au cours d'un sprint. Les User Stories doivent être définies par le Product Owner et doivent satisfaire à la définition de « Ready » (Prêt), et doivent être estimées, avant d'apparaître dans le Product Backlog priorisé pour être ajoutées à un prochain sprint.

8.4.3.2 Personnages*

Les personnages sont des personnages fictifs très détaillés, représentant la majorité des utilisateurs ainsi que les autres parties prenantes de l'entreprise qui ne verront pas forcément directement le produit fini. Les personnages sont créés pour identifier les besoins des principaux utilisateurs cibles. La création de personnages spécifiques peut aider l'équipe à mieux comprendre les utilisateurs ainsi que leurs besoins et objectifs. En fonction d'un personnage, le Product Owner est à même de prioriser les fonctionnalités de façon plus efficace afin de créer le backlog de produit priorisé.

Créer un personnage : Cela implique d'attribuer au personnage un nom fictif et de préférence une image (telle qu'une image d'archive). Le personnage devra inclure des attributs très spécifiques tels que l'âge, le genre, le niveau d'éducation, l'environnement, les intérêts et les objectifs. Une citation illustrant les exigences du personnage peut également être incluse. L'exemple ci-dessous correspond à un personnage pour un site de voyage.

Exemple :

Vanessa, 39 ans, habite à San Francisco. Après une brillante carrière en tant qu'avocate, elle poursuit sa passion pour les voyages. Elle aime avoir plusieurs alternatives lorsqu'elle choisit ses vols et ses hébergements de façon à pouvoir choisir l'option la meilleure et la plus abordable. Elle est frustrée par les sites Web lents et encombrés.

8.4.3.3 Changements approuvés

Les changements non ratifiés peuvent être acceptés par le Product Owner durant le processus de *développement des epics*, parfois avec des suggestions offertes par parties prenantes de l'entreprise concernées. Ces changements sont catégorisés en tant que changements approuvés et peuvent être priorisés et mis en œuvre lors d'un sprint à venir.

Les demandes de changement et demandes de changement approuvées sont abordées dans les sections 6.3.1, 6.4.2.1 et 6.6.

8.4.3.4 Risques identifiés

Lors de la création des epics, de nouveaux risques peuvent être identifiés. Ces risques identifiés forment un résultat important de cette étape. Ces risques contribuent au développement du backlog de produit priorisé (aussi appelé backlog de produit ajusté aux risques).

L'identification des risques est décrite dans la section 7.4.1.

8.5 Création du backlog de produit priorisé

Au cours de ce processus, les Epics sont raffinées, élaborés et, surtout, priorisées en fonction de leur valeur commerciale respective, afin de créer un Product Backlog priorisé pour le projet. En outre, sur la base des recommandations du Scrum Guidance Body, le Product Owner et l'équipe Scrum établissent les critères « Done » (Terminé) du projet.

La figure 8-12 illustre les contributions, outils et résultats du processus de *création du backlog de produit priorisé*.

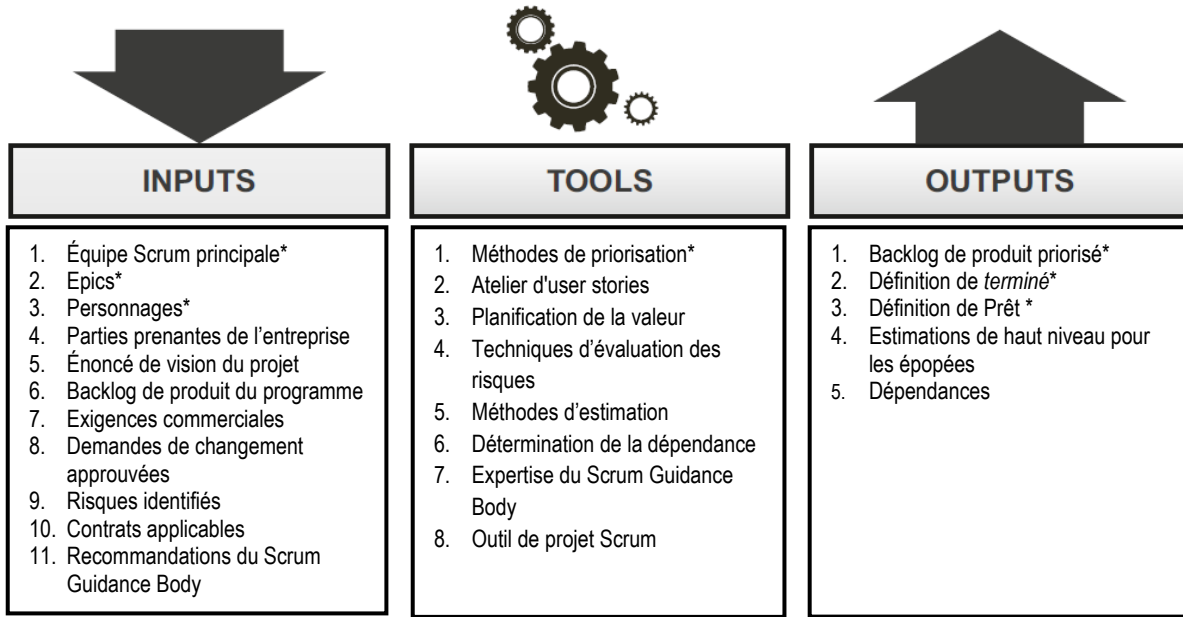


Figure 8-12 : Création du backlog de produit priorisé — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

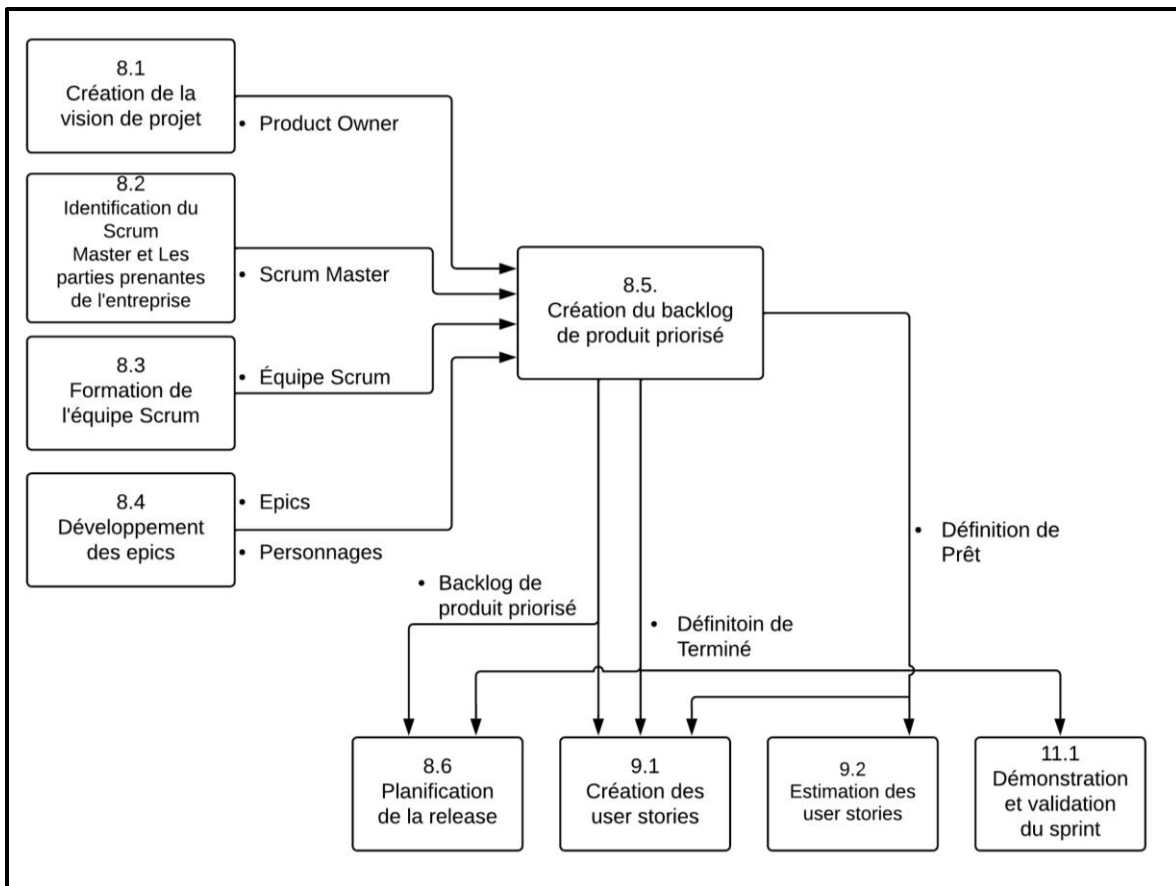


Figure 8-13 : Création du backlog de produit priorisé — Diagramme de flux de données

8.5.1 Contributions

8.5.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

8.5.1.2 Epics*

Décrits dans la section 8.4.3.1.

8.5.1.3 Personnages*

Décrits dans la section 8.4.3.2.

8.5.1.4 Parties prenantes de l'entreprise

Décrites dans la section 8.2.3.2.

8.5.1.5 Énoncé de vision du projet

Décrit dans la section 8.1.3.2.

8.5.1.6 Backlog de produit du programme

Décrit dans la section 8.1.1.5.

8.5.1.7 Exigences commerciales

La somme de toutes les informations obtenues grâce à divers outils tels que les entretiens avec les utilisateurs ou les clients, les questionnaires, les sessions JAD, l'analyse des lacunes, l'analyse FFOM et autres réunions, permet de développer une meilleure perspective des exigences commerciales et aide à créer le backlog de produit priorisé.

8.5.1.8 Demandes de changement approuvées

Décrites dans la section 8.4.3.3.

8.5.1.9 Risques identifiés

Décrits dans la section 8.4.3.4.

8.5.1.10 Contrats applicables

Décrits dans la section 8.4.1.9.

8.5.1.11 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Lors de la création du backlog de produit priorisé, les recommandations du Scrum Guidance Body peuvent inclure des informations relatives aux règles, règlements, standards et meilleures pratiques pour le développement du backlog de produit priorisé.

8.5.2 Outils

8.5.2.1 Méthodes de priorisation*

Ci-dessous sont listées certaines techniques utilisées pour la priorisation des user stories ou des exigences dans le backlog de produit en fonction de leur valeur commerciale.

- **Méthode de priorisation MoSCoW** — la méthode de priorisation MoSCoW tient son nom des premières lettres des expressions suivantes : « Must have » (« doit être fait », vital) , « Should have » (« devrait être fait », essentiel), « Could have » (« devrait être fait, si possible », confort) et « Won't have » (« impossible », luxe). Cette méthode de priorisation est souvent plus efficace que les schémas simplifiés. Ces étiquettes sont un ordre de priorité décroissant avec « Must have » représentant les user stories sans lesquelles le produit n'aura aucune valeur, et « Won't have » les user stories qui, bien qu'elles seraient les bienvenues, ne sont pas nécessaires.

- **Comparaison par paire** — c'est une technique où la liste complète des user stories dans le backlog de produit priorisé est préparée. Puis, chaque user story est considérée individuellement et comparée à chacune des autres user stories dans la liste. Chaque fois que deux user stories sont comparées, une décision est prise pour définir laquelle des deux est la plus importante. À la fin de ce processus, il est possible de générer une liste priorisée des user stories.
- **Méthode des 100 points** — la méthode des 100 points a été développée par Dean Leffingwell et Don Widrig (2003). Elle consiste à donner au client 100 points qu'il peut ensuite utiliser pour voter pour les user stories qui sont les plus importantes. L'objectif est de donner plus de poids aux user stories dont le degré de priorité est plus important comparé aux autres user stories disponibles. Chaque membre du groupe attribue des points aux différentes user stories, en donnant plus de points à celles qu'il juge plus importantes. À la fin du processus de vote, l'ordre de priorité est déterminé en calculant le nombre total de points attribués à chaque user story.
- **Analyse Kano**

Décrite dans la section 4.5.2.

8.5.2.2 **Atelier d'user stories**

Décrit dans la section 8.4.2.2.

8.5.2.3 **Planification de la valeur**

Décrite dans la section 4.5.2.

8.5.2.4 **Techniques d'évaluation des risques**

Décrites dans la section 7.4.2.1.

8.5.2.5 **Méthodes d'estimation**

Il peut être utile de formuler des estimations de temps de très haut niveau pour les Epics car ces estimations peuvent aider le Product Owner à planifier les releases de manière appropriée et peuvent également aider au processus de priorisation.

Toutefois, étant donné que les Epics planifiées à ce stade ne sont généralement pas encore décomposées en User Stories bien définies, et qu'il est également possible que certaines des Epics initialement développées ne soient pas implémentées, l'équipe ne devrait pas passer trop de temps à ce stade sur l'estimation des Epics.

Cet outil est utilisé différemment dans le processus d'*estimation des User Stories* (tel que décrit dans la section 9.2.2.1), où l'équipe Scrum est très impliquée dans l'estimation des User Stories bien définies. Le Product Owner, le Scrum Master et les parties prenantes de l'entreprise concernées doivent comprendre que toutes les estimations de haut niveau fournies dans ce processus ne sont utilisées qu'à titre indicatif, et qu'il peut y avoir des écarts importants par rapport aux estimations de l'équipe Scrum pour les User Stories, une fois que les User Stories sont correctement définies et estimées.

Certains outils spécifiques peuvent être utilisés pour fournir des estimations de haut niveau des Epics :

- **Estimations des Epics préexistantes**—si des Epic similaires ont été développées dans le passé pour le même projet ou pour d'autres projets connexes, ces estimations antérieures peuvent être utilisées pour établir des estimations détaillées dans ce processus.
- **Avis d'expert**—les experts qui ont implémenté des fonctionnalités de projet similaires dans le passé peuvent être en mesure de fournir des estimations d'effort de haut niveau pour les Epics.
- **Estimations de 5 à 10 minutes**—les membres de l'équipe concernés peuvent recevoir une vue d'ensemble des exigences générales pour chaque Epic et être invités à fournir une estimation approximative (par exemple, en heures, jours ou semaines) de cinq à dix minutes. Cette méthode d'estimation devrait être une option lorsque seul un petit nombre d'Epics doit être estimé, sinon elle pourrait devenir chronophage.
- **Estimation par affinité**—l'estimation par affinité, comme le dimensionnement des t-shirts, est une technique utilisée pour créer rapidement une estimation de temps pour un grand nombre d'Epics. Ce point est décrit en détail dans la section 9.2.2.1.

8.5.2.6 Détermination des dépendances

Une bonne documentation des « dépendances » aide l'équipe Scrum à déterminer l'ordre relatif dans lequel les Epics (et les User Stories) doivent être exécutées pour créer les produits livrables du Sprint. Les dépendances mettent en évidence la relation et l'interaction entre les Epics (et entre les User Stories), à la fois au sein de l'équipe Scrum travaillant sur un Sprint donné et avec les autres équipes Scrum travaillant sur le projet. Au fur et à mesure que le Product Backlog priorisé est créé, le Product Owner identifie toutes les dépendances et relations entre les Epics (et entre les User Stories), y compris les dépendances techniques et les dépendances liées à la disponibilité des personnes, car ces dépendances auront un impact sur l'ordre et la priorité du travail à effectuer sur le projet.

Les dépendances peuvent être obligatoire ou discrétionnaire, interne ou externe, ou une combinaison de ceux-ci. Par exemple, une dépendance peut être à la fois obligatoire et externe pour le projet.

Vous trouverez ci-dessous une description de chaque type de dépendance.

- **Dépendances obligatoires** —les dépendances obligatoires sont des dépendances qui dictent l'ordre des tâches à réaliser dans le cadre du projet, et qu'il est donc important de prendre en compte lors de la création initiale du Product Backlog priorisé. Ces dépendances peuvent être obligatoires en raison de la nature du travail (par exemple, une limitation physique de la séquence de travail), ou peuvent exister en raison d'obligations contractuelles ou d'exigences légales. Les dépendances obligatoires sont également communément décrites comme une logique dure.
- **Dépendances discrétionnaires**—les dépendances discrétionnaires sont des dépendances qui sont placées dans le workflow en fonction des expériences passées ou des meilleures pratiques dans un domaine ou un domaine particulier. Par exemple, l'équipe peut décider de terminer une Epic (ou User Story) avant une autre parce que ce flux de travail a bien fonctionné dans des projets antérieurs.
- **Dépendances externes**—Les dépendances externes sont des dépendances relatives à des activités ou des facteurs qui sortent du cadre du travail à exécuter par l'équipe Scrum mais qui sont nécessaires pour terminer une tâche de projet ou créer un livrable de projet. Les dépendances externes sont généralement hors du contrôle de l'équipe Scrum et peuvent donc générer un risque plus important pour un projet.
- **Dépendances internes**—Les dépendances internes sont les facteurs qui, au sein du projet, affectent la séquence des travaux à effectuer. Ces facteurs sont généralement sous le contrôle de l'équipe Scrum.

Il existe différentes façons d'identifier, de définir et de présenter les tâches du projet et leurs dépendances.

8.5.2.7 Expertise du Scrum Guidance Body

Décrite dans la section 8.4.2.7.

Lors de la création du backlog de produit priorisé, l'expertise du Scrum Guidance Body peut faire référence aux règles et règlements documentés ou aux standards et meilleures pratiques pour la création des epics. Il peut également y avoir une équipe d'experts sur le sujet, capable d'assister le Product Owner lors du processus de création du backlog de produit priorisé. Cette équipe peut inclure des analystes commerciaux, des architectes principaux, des développeurs expérimentés, des experts Scrum et/ou d'autres personnes expérimentées. Ce groupe d'experts n'est normalement pas l'équipe qui travaillera au développement de ce projet. Ces experts tendent à naviguer de projet en projet durant la « phase de vente » ou « phase zéro » avec les clients ou utilisateurs.

8.5.2.8 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

8.5.3 Résultats

8.5.3.1 Backlog de produit priorisé*

Le Product Owner développe un backlog de produit contenant une liste priorisée des exigences commerciales et des exigences du projet écrite sous forme d'epics qui sont des user stories de haut niveau. Le backlog de produit priorisé est basé sur trois facteurs principaux : la valeur, le risque ou l'incertitude, et les dépendances. Il est également appelé backlog de produit ajusté aux risques puisqu'il inclut les risques identifiés et évalués relatifs au projet. Il inclut également tous les changements approuvés pouvant être correctement priorisés dans le backlog de produit priorisé (tels qu'ils sont décrits dans la section 6.3.1).

- **Valeur** — le Product Owner est responsable de garantir la livraison en priorité des produits qui offrent le plus haut degré de valeur commerciale. Même un produit extrêmement précieux peut ne pas faire partie de la première release s'il y a d'autres produits de valeur encore plus élevée qui suffisent à la première release.
- **Risque et incertitude** — plus il existe d'incertitudes plus le projet est risqué. Il est donc important d'accorder un plus fort degré de priorité aux produits les plus risqués du backlog de produit priorisé. Les produits comportant un haut niveau de risque nécessiteront également des actions d'atténuation des risques. Lorsque ces actions d'atténuation des risques sont priorisées contre le backlog, le résultat est un backlog de produit ajusté aux risques. Traiter les risques le plus tôt possible durant le projet ne garantit pas que le projet réussira, mais cela permet d'améliorer la capacité de l'équipe à faire face aux risques. Cela est décrit dans la section 7.4.3.
- **Dépendances** — La plupart des projets auront intrinsèquement des dépendances entre certaines Epics ou User Stories. Ces dépendances doivent être prises en considération lors de la création du Product Backlog priorisé. Il est important de noter que la priorisation des Epics peut être différente de celle des User Stories sous-jacentes. Par exemple, même si une Epic est catégorisée comme hautement prioritaire, certaines User Stories contenues dans l'Epic peuvent être catégorisées comme faiblement prioritaires tandis que d'autres User Stories dans l'Epic peuvent être catégorisées comme hautement prioritaires.
- **Estimations** — les estimations de haut niveau pour les epics sont également disponibles dans le backlog de produit priorisé.

8.5.3.2 Définition de *terminé**

La définition de *terminé* est une série de règles applicables à toutes les user stories. Une définition claire de *terminé* est essentielle car elle permet d'éviter toute ambiguïté au sujet des exigences et d'aider l'équipe à respecter les normes de qualité obligatoires. Ces critères clairs sont utilisés afin de créer la *définition de terminé* qui est le résultat du processus de *création du backlog de produit priorisé*. Une user story est considérée comme *terminée* lorsque la démonstration en a été faite et qu'elle a été approuvée par le Product Owner qui la juge selon la définition de *terminé* et les critères d'acceptation des user stories.

8.5.3.3 Définition de « Ready » (Prêt)*

La définition de « Ready » (Prêt) définit les critères qu'une User Story doit satisfaire avant d'être éventuellement estimée et incluse dans un sprint. La définition de « Ready » (Prêt) met la responsabilité sur le Product Owner de définir correctement les exigences pour chaque User Story. Sans exigences correctement définies, il sera impossible d'obtenir des estimations fiables et l'équipe Scrum ne sera pas en mesure de réaliser efficacement le travail requis pour le projet.

8.5.3.4 Estimations de haut niveau pour les Epics

Les estimations de haut niveau pour les Epics peuvent être initiées par le Product Owner en utilisant diverses méthodes d'estimation (comme décrit dans la section 8.5.2.5). Ces estimations aideront le Product Owner à se faire une idée approximative du temps et des efforts nécessaires à la réalisation de chaque Epic, ce qui facilitera la priorisation des Epics dans le Product Backlog priorisé et à planifier les releases du projet.

8.5.3.5 Dépendances

Les dépendances décrivent la relation et l'interaction entre les différents Epics (ou User Stories) dans un projet. Les dépendances peuvent être classées comme obligatoires ou discrétionnaires, internes ou externes, ou une combinaison de celles-ci (comme indiqué dans la section 8.5.2.6). Les dépendances affecteront l'ordre relatif dans lequel les Epics (et les User Stories) seront exécutées pour créer les produits livrables du sprint, et affecteront donc leur priorité, comme le documente le Product Backlog priorisé.

8.6 Planification de la release

Dans ce processus, le Product Owner, avec les inputs des parties prenantes de l'entreprise et des membres de l'équipe Scrum, élabore le calendrier initial de planification de release, qui est communiqué et partagé avec toutes les parties prenantes et les membres de l'équipe Scrum. Il est entendu que la nature itérative de Scrum peut nécessiter des ajustements futurs du calendrier de release. La durée des sprints est également déterminée au cours de ce processus.

La figure 8-14 illustre les contributions, outils et résultats du processus de *planification de la release*.

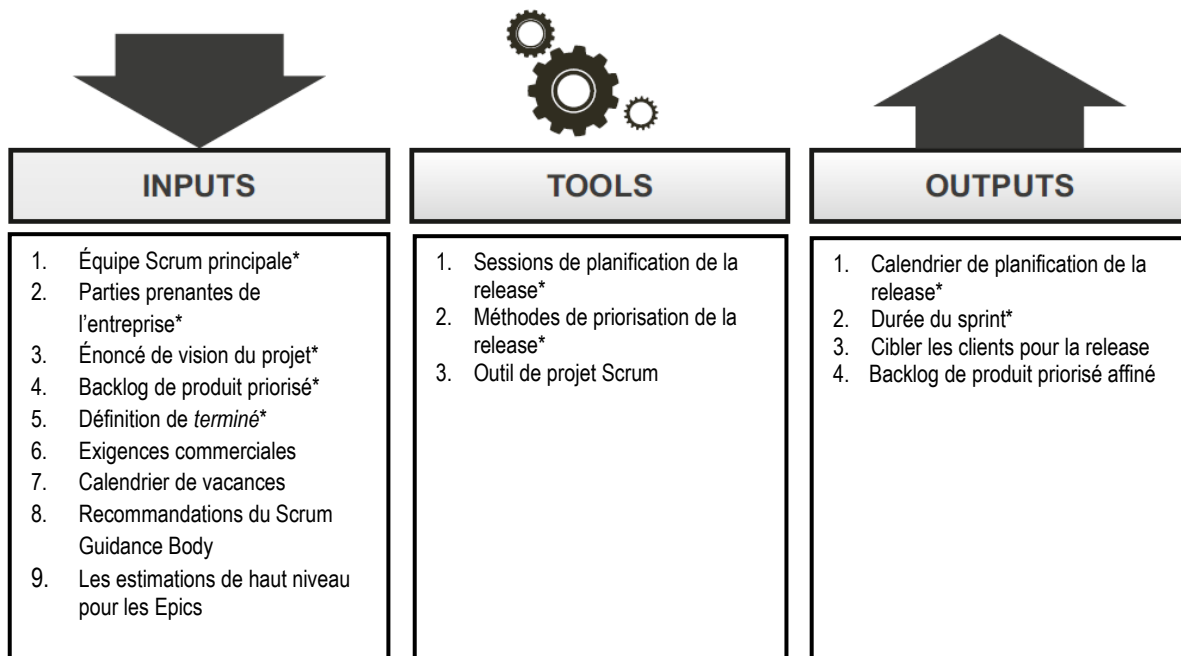


Figure 8-14 : Planification de la release — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

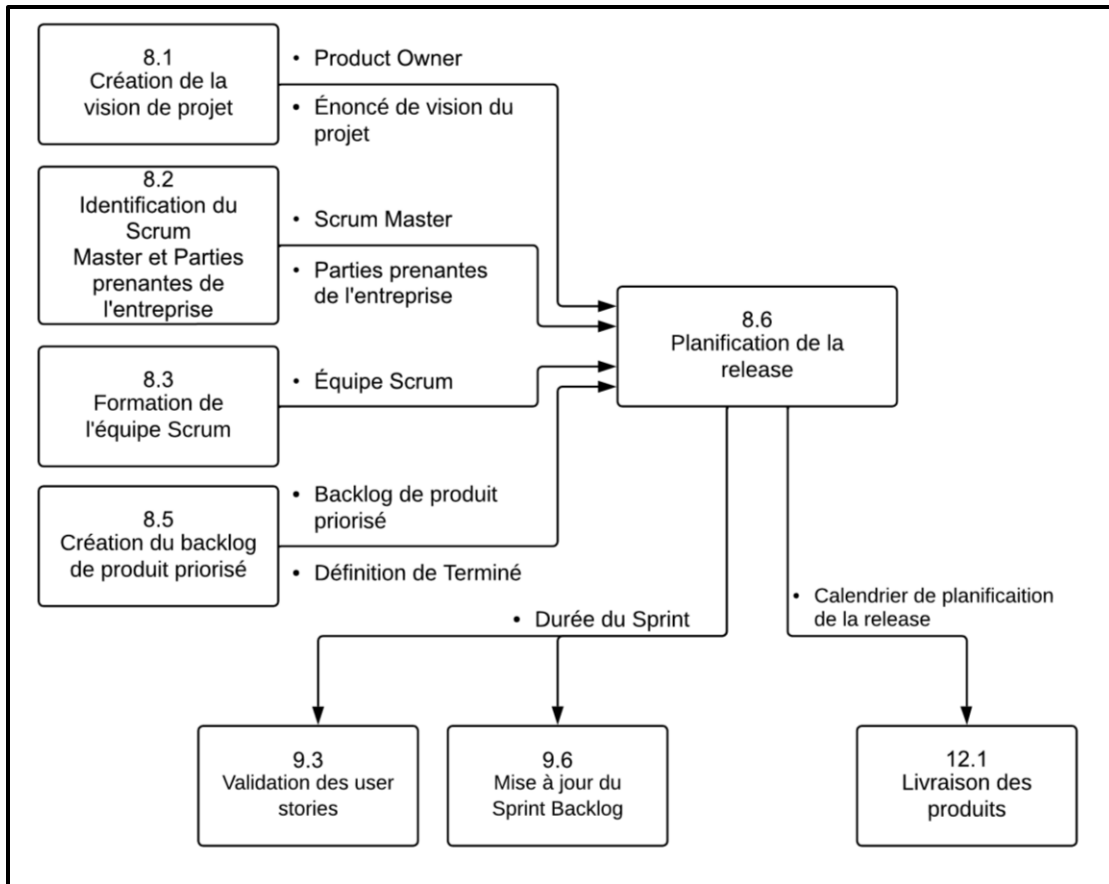


Figure 8-15 : Planification de la release — Diagramme de flux de données

8.6.1 Contributions

8.6.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

8.6.1.2 Parties prenantes de l'entreprise *

Décrites dans la section 8.2.3.2.

8.6.1.3 Énoncé de vision du projet*

Décrit dans la section 8.1.3.2.

8.6.1.4 **Backlog de produit priorisé***

Décrit dans la section 8.5.3.1.

8.6.1.5 **Définition de *terminé****

Décrite dans la section 8.5.3.2.

8.6.1.6 **Exigences commerciales**

Décrites dans la section 8.5.1.7.

8.6.1.7 **Calendrier de vacances**

Il est important que l'équipe Scrum effectue le suivi des dates clés et de la disponibilité de tous les membres de l'équipe. Cela peut être accompli en utilisant un calendrier partagé qui indique les jours fériés, les vacances, projets de voyage, événements, etc. Ce calendrier aidera l'équipe à planifier et réaliser les sprints.

8.6.1.8 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

Lors du processus de *planification de la release*, les recommandations du Scrum Guidance Body peuvent être relatives aux règles, règlements, standards et meilleures pratiques concernant le développement du plan de release. Le Guidance Body peut être la meilleure autorité capable de définir les directives relatives à la valeur commerciale, aux attentes de la release, aux stratégies de déploiement, à la qualité et à la sécurité.

8.6.1.9 **Les estimations de haut niveau pour les Epics**

Lors des sessions de planification de release, les estimations de haut niveau pour les Epics sont utilisées pour planifier les releases en utilisant soit une approche axée sur le calendrier, soit une approche axée sur les fonctionnalités. Pour plus d'informations sur ces estimations, voir les sections 8.5.2.5 et 8.5.3.4.

8.6.2 Outils

8.6.2.1 Sessions de planification de la release*

Les sessions de planification de la release sont utilisées pour développer un plan de release. Le plan définit quand certains ensembles de fonctionnalités ou produits utilisables seront livrés au client. Dans Scrum, l'objectif principal d'une session de planification de la release est de permettre aux membres de l'équipe Scrum d'avoir une vue d'ensemble du calendrier des releases et des livraisons pour le produit qu'ils développent de façon à respecter les attentes du Product Owner et des parties prenantes de l'entreprise concernées (principalement le sponsor du projet).

La majorité des organisations disposent d'une stratégie concernant la release des produits. Certaines organisations préfèrent le déploiement continu, où une release est effectuée après la création d'une fonctionnalité utilisable spécifique. D'autres organisations préfèrent le déploiement par phases, où les releases sont effectuées à des intervalles prédéfinis. Selon la stratégie de l'organisation, les sessions de planification des releases d'un projet peuvent être motivées par la fonctionnalité, auquel cas l'objectif est de livrer une release dès qu'un ensemble prédéfini de fonctionnalités a été développé ; ou motivées par les échéances, auquel cas la release a lieu à une date prédéfinie.

Puisque l'approche Scrum encourage la prise de décision itérative basée sur les informations plutôt que la planification préalable détaillée pratiquée dans la gestion de projet traditionnelle en cascade, les sessions de planification de la release ne doivent pas nécessairement produire un programme de release détaillé pour le projet dans son entier. Le plan de release peut être mis à jour de façon continue au fur et à mesure que les informations pertinentes deviennent disponibles.

8.6.2.2 Méthodes de priorisation de la release*

Ces méthodes sont utilisées pour développer un plan de mise à disposition. Ces méthodes sont spécifiques au secteur et à l'organisation et sont normalement déterminées par les cadres supérieurs au sein de l'organisation.

8.6.2.3 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

8.6.3 Résultats

8.6.3.1 Calendrier de planification de la release*

Un calendrier de planification de la release est le résultat principal du processus de *planification de la release*. Le calendrier de planification de la release établit quels éléments livrables doivent être mis à disposition des clients, ainsi que les intervalles prévus et dates de release. Il est possible qu'une mise à disposition ne soit pas prévue à la fin de chaque sprint. Une release peut parfois être planifiée après qu'un groupe d'itérations de sprint a été réalisé. L'élément livrable doit être mis à disposition lorsqu'il offre suffisamment de valeur commerciale au client.

Une release comprend généralement un groupe de User Stories dans le Product Backlog priorisé qui doivent être réalisées et expédiées ensemble dans le cadre de la release. Il est important de noter, cependant, que les User Stories d'une release n'ont pas toujours la fonctionnalité complète des Epics. Certaines Epics peuvent n'être que partiellement achevées, car toutes les User Stories dans une Epic ne sont pas nécessairement d'une priorité suffisamment élevée pour faire partie de cette release particulière. Pendant le processus de *planification de la release*, il peut être utile de tenir compte de la durée du sprint et des hypothèses relatives à la vitesse de l'équipe afin d'obtenir un meilleur calendrier de planification de la release.

8.6.3.2 Durée du Sprint*

Le Product Owner et l'équipe Scrum décident de la durée des sprints pour le projet en se basant sur les informations disponibles, y compris les exigences commerciales et le calendrier de planification de la release. Une fois déterminée, la durée du sprint reste souvent la même tout au long du projet. Cependant, elle peut être modifiée si cela est jugé nécessaire par le Product Owner et par l'équipe Scrum. Au début du projet, ils peuvent encore être en train d'expérimenter afin d'établir la meilleure durée de sprint. Plus tard dans le projet, une modification de la durée du sprint signifie normalement que les sprints peuvent être réduits grâce aux améliorations dans l'environnement du projet. Pour tirer le maximum de bénéfices d'un projet Scrum et offrir une flexibilité maximale pour le changement, la durée d'un sprint doit être aussi courte que possible. Un sprint est généralement limité dans le temps et dure de une à quatre semaines. La plupart des projets Scrum ont généralement des sprints limités dans le temps d'une durée de deux ou trois semaines. Toutefois, pour les projets dont les exigences sont très stables, les sprints peuvent s'étendre jusqu'à quatre semaines, si cela est justifié.

L'impact des changements prévus sur la durée du sprint est décrit dans la section 6.5.1.

8.6.3.3 Cibler les clients pour la release

Toutes les releases ne ciblent pas tous les utilisateurs ou parties prenantes de l'entreprise. Parties prenantes de l'entreprise peuvent choisir de limiter certaines releases à un sous-ensemble d'utilisateurs. Le calendrier des releases doit préciser les clients ciblés par la release.

8.6.3.4 Backlog de produit priorisé affiné

Le backlog de produit développé au cours du processus de *création du backlog de produit priorisé* est affiné au cours de ce processus. Il peut y avoir des précisions supplémentaires sur les user stories du backlog de produit priorisé après que l'équipe Scrum principale a mené les sessions de planification de la release avec parties prenantes de l'entreprise.

8.7 Lancement de la phase de diagramme de flux de données

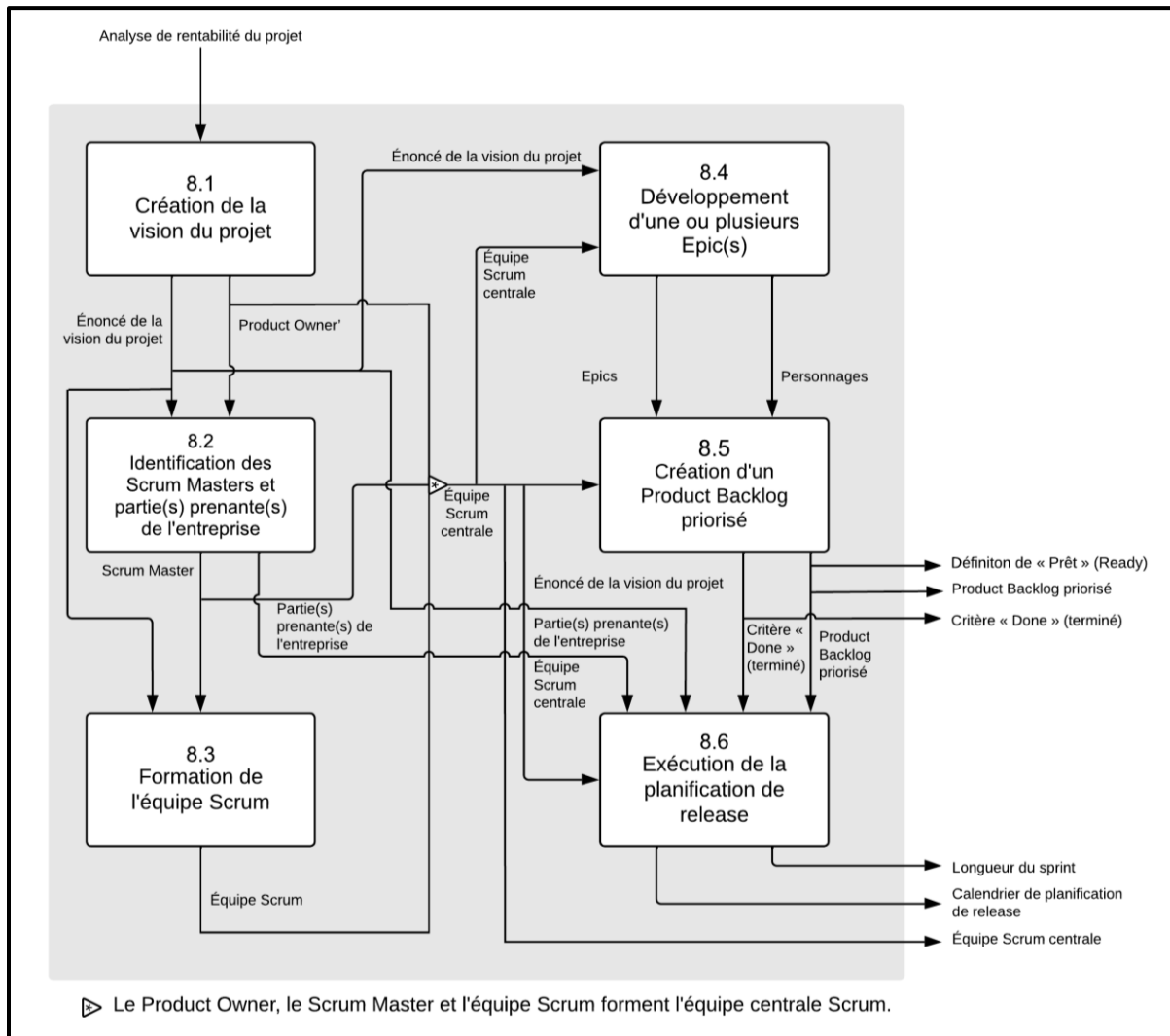


Figure 8-16 : Phase d'initiation — Diagramme de flux de données

9. PLANIFICATION ET ESTIMATION

La phase de planification et d'estimation comporte les processus associés à la planification et à l'estimation des tâches, notamment : la *création des user stories*, l'*estimation des user stories*, la *validation des user stories*, l'*identification et l'estimation des tâches* ainsi que la *Mise à jour du Sprint Backlog*.

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la *phase de planification et d'estimation* s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Afin de faciliter la meilleure application possible de l'approche Scrum, ce chapitre identifie les contributions, outils et résultats de chaque processus comme étant soit « obligatoires » ou « facultatifs ». Les contributions, outils et résultats accompagnés d'un astérisque (*) sont obligatoires ou essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont facultatifs.

Il est recommandé que l'équipe Scrum et les personnes qui découvrent l'approche et les processus Scrum se concentrent principalement sur les contributions, outils et résultats obligatoires. En revanche, les Product Owners, Scrum Masters et autres praticiens Scrum expérimentés pourront aspirer à améliorer leurs connaissances grâce à l'ensemble des informations contenues dans ce chapitre. Il est également important de noter que bien que tous les processus ne soient définis qu'une seule fois dans le *Guide SBOK®*, ils ne sont pas nécessairement réalisés de façon séquentielle ou séparément. À certains moments, il peut s'avérer utile de combiner certains processus, en fonction des exigences spécifiques de chaque projet.

Ce chapitre est écrit depuis la perspective de l'équipe Scrum travaillant sur un sprint afin de produire des éléments potentiellement livrables, pouvant faire partie d'un plus gros projet, programme ou portefeuille. Des informations supplémentaires concernant le passage à l'échelle de Scrum pour les gros projets sont disponibles dans le chapitre 13 et celles sur le passage à l'échelle pour l'entreprise dans le chapitre 14.

Une fois la phase de lancement terminée, les cycles itératifs de sprint peuvent commencer. Planifier et estimer est la première des trois phases qui sont effectuées de manière répétitive, dans chaque cycle de sprint. Au début d'un sprint, le Product Owner et l'équipe Scrum, sous la direction du Scrum Master, planifient le sprint.

Le Product Owner affine les Epics les plus prioritaires en un ensemble de User Stories bien écrites et estimées, que l'équipe de Scrum s'engage à réaliser au cours du prochain Sprint en fonction d'hypothèses de vélocité de l'équipe. Le Scrum Master et l'équipe de Scrum créent et mettent à jour le Sprint Backlog avec la liste des User Stories validées qui doivent être livrées dans le cadre du sprint.

L'équipe Scrum planifie ensuite son travail de manière plus détaillée en identifiant et, éventuellement, en estimant les tâches qu'elle doit accomplir afin de livrer les User Stories pour le Sprint. Comme étape finale de la planification du sprint, l'équipe complète le Sprint Backlog avec le détail des tâches et, si possible, leurs estimations. Le Sprint Backlog sera utilisé dans la phase d'implémentation pour suivre les progrès de l'équipe pendant le sprint.

Il est également important de réaliser que, bien que toutes les phases et tous les processus soient définis de manière unique dans le guide SBOK®, ils ne sont pas nécessairement exécutés séquentiellement ou séparément. Parfois, il peut être plus approprié de faire chevaucher certaines phases et/ou processus, en fonction des exigences spécifiques de chaque projet.

La figure 9-1 offre un aperçu des processus de la phase de planification et d'estimation qui sont les suivants :

9.1 Création des user stories — Dans ce processus, les User Stories et leurs critères d'acceptation associés sont créés par le Product Owner (élaborés à partir des Epics définies précédemment) et incorporés dans le Product Backlog priorisé. En règle générale, les User Stories sont écrites par le Product Owner et sont conçues pour s'assurer que les exigences du client sont clairement expliquées et sont compréhensibles pour toutes les parties prenantes de l'entreprise. Les User Stories doivent être suffisamment tangibles et doivent satisfaire à la définition de « Ready » (Prêt) avant de pouvoir être estimées et développées par l'équipe Scrum.

9.2 Estimation des user stories — Dans ce processus, l'équipe Scrum, soutenue par le Scrum Master estime les User Stories et identifie les efforts requis pour développer la fonctionnalité décrite dans chaque User Story. Seules les User Stories qui satisfont à la définition de « Ready » (Prêt) et qui sont correctement définies par le Product Owner sont estimées par l'équipe.

9.3 Validation des user stories — Dans ce processus, l'équipe Scrum s'engage à livrer un ensemble de User Stories pour le sprint. Le choix des User Stories est basé sur la priorité relative basée sur la valeur des User Stories ainsi que l'effort estimé et la vélocité de l'équipe pour un sprint. Dans le cadre de ce processus, l'équipe Scrum commence à créer le Sprint Backlog, qui contient les User Stories validées qui sont assignées à un sprint particulier. Le backlog est raffiné avec des détails au niveau des tâches au fur et à mesure de la planification du sprint. Avec la validation de l'équipe Scrum donné au début d'un sprint dans le cadre de la planification du sprint, le contenu du sprint est défini et ne peut être modifié une fois que la phase d'implémentation du sprint a commencé.

9.4 Identification des tâches — Dans ce processus, les User Stories validées sont décomposées en tâches spécifiques et compilées dans une liste de tâches. L'identification des tâches peut se faire soit au début du sprint pour toutes les User Story validées, soit avant que l'équipe ne commence à travailler sur les tâches requises pour chaque User Story.

9.5 Estimation des tâches — Il s'agit d'un processus optionnel qui implique la création d'estimations de tâches si l'équipe Scrum en voit l'intérêt. Dans ce processus, l'équipe Scrum estime l'effort requis pour accomplir chaque tâche de la liste des tâches. Les estimations des tâches peuvent être déterminées soit au début du Sprint pour toutes les User Stories/tâches pertinentes pour ce sprint ou pour chaque tâche juste avant que l'équipe ne commence à travailler sur la User Story/tâche en question. L'estimation peut être faite en utilisant les mêmes méthodes que celles utilisées pour le processus d'estimation des User Stories.

9.6 Mise à jour du Sprint Backlog — Dans ce processus, l'équipe centrale Scrum met à jour le Sprint Backlog avec les détails des tâches et, si elles sont disponibles, les estimations des tâches. Le Sprint Backlog mis à jour sera utilisé dans la phase d'implémentation pour suivre les progrès de l'équipe au cours du prochain sprint.

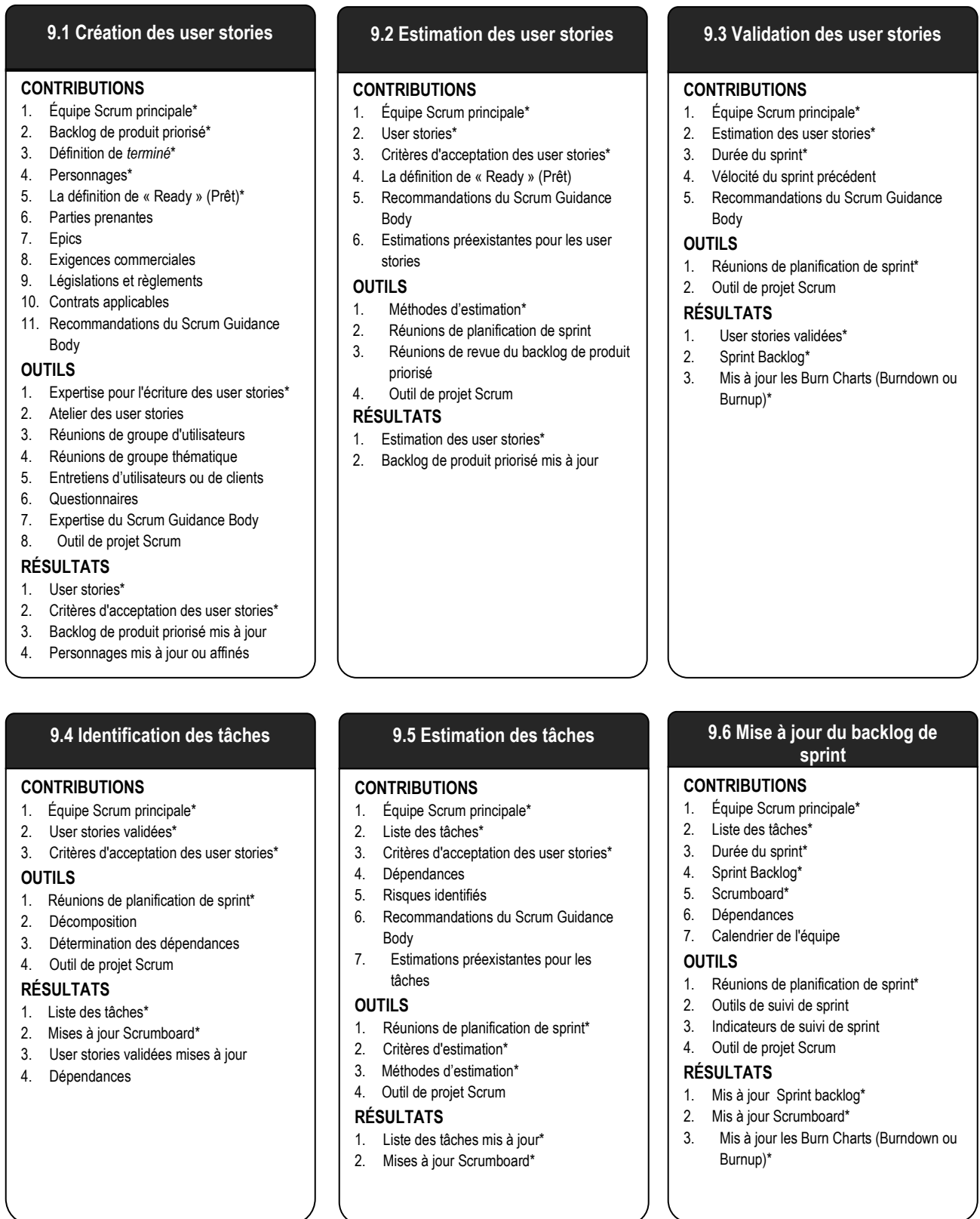


Figure 9-1 : Aperçu de la phase de planification et d'estimation

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

La figure 9-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de la phase de planification et d'estimation.

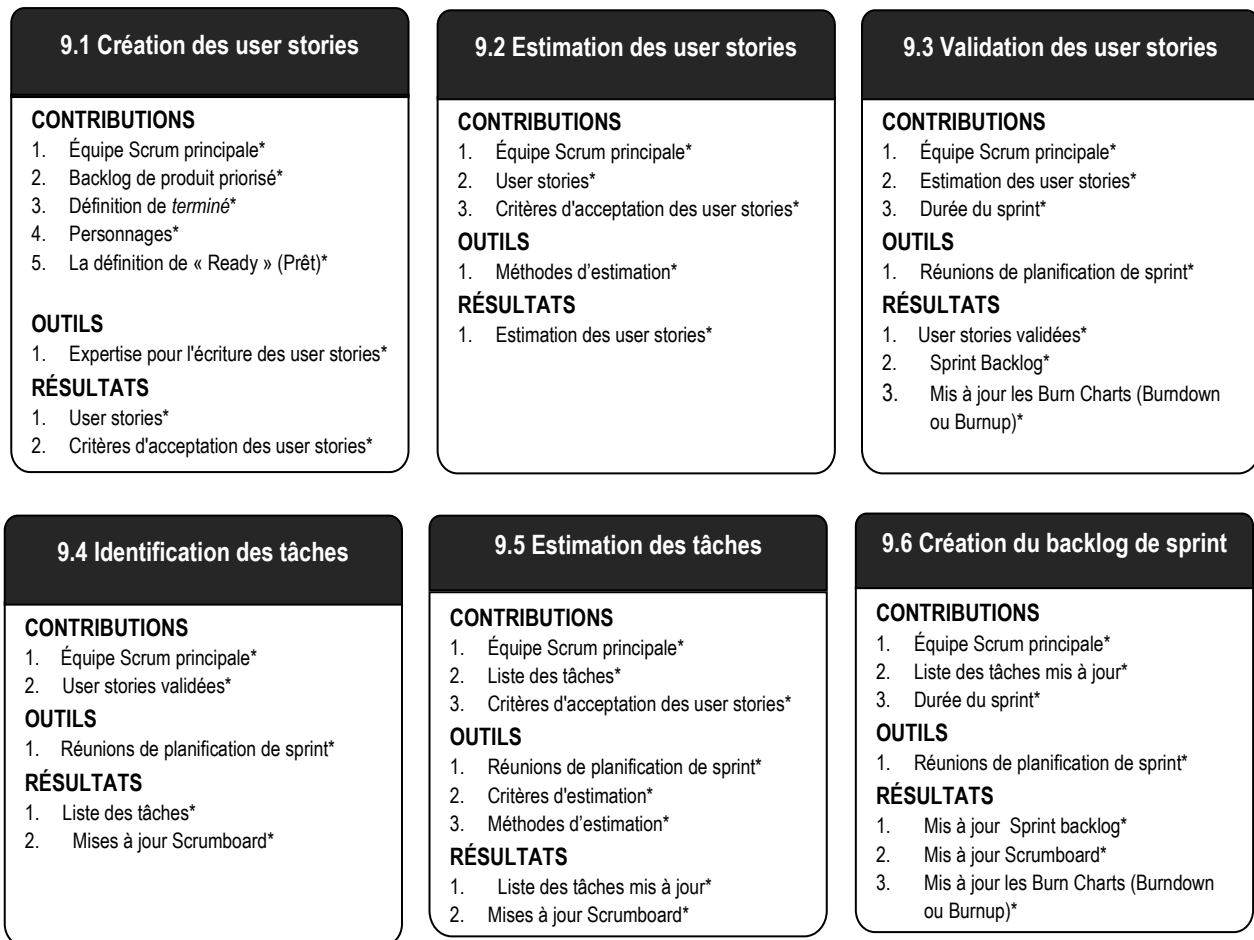


Figure 9-2 : Aperçu de la phase de planification et d'estimation (éléments essentiels)

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

9.1 Création des user stories

Dans ce processus, les User Stories et leurs critères d'acceptation sont créés par le Product Owner (élaborés à partir des Epics précédemment définies) et incorporés dans le Product Backlog priorisé. Les User Stories sont conçues pour garantir que les exigences du client sont clairement décrites et peuvent être pleinement comprises par toutes les parties prenantes du projet. Les User Stories doivent être suffisamment tangibles et doivent satisfaire à la définition de « Ready » (Prêt) avant de pouvoir être estimées et développées par l'équipe Scrum.

Les ateliers sur les User Stories peuvent être utilisés pour aider les membres de l'équipe Scrum à mieux comprendre les User Stories créées par le Product Owner.

La figure 9-3 illustre les contributions, outils et résultats du processus de *création des user stories*.

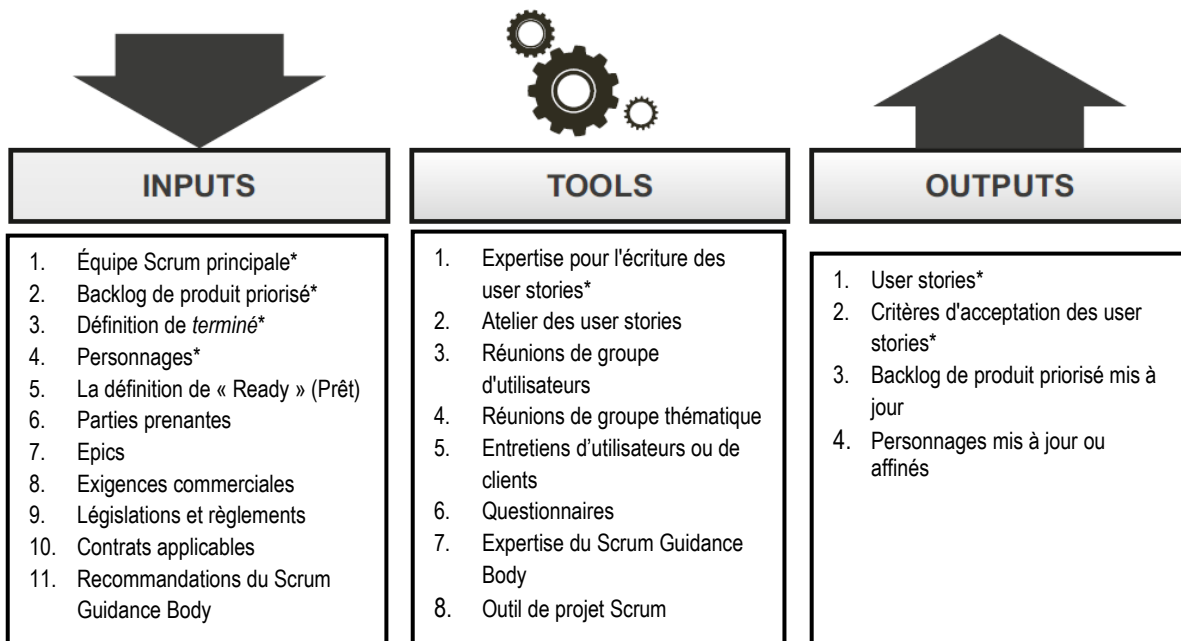


Figure 9-3 : Création des user stories — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

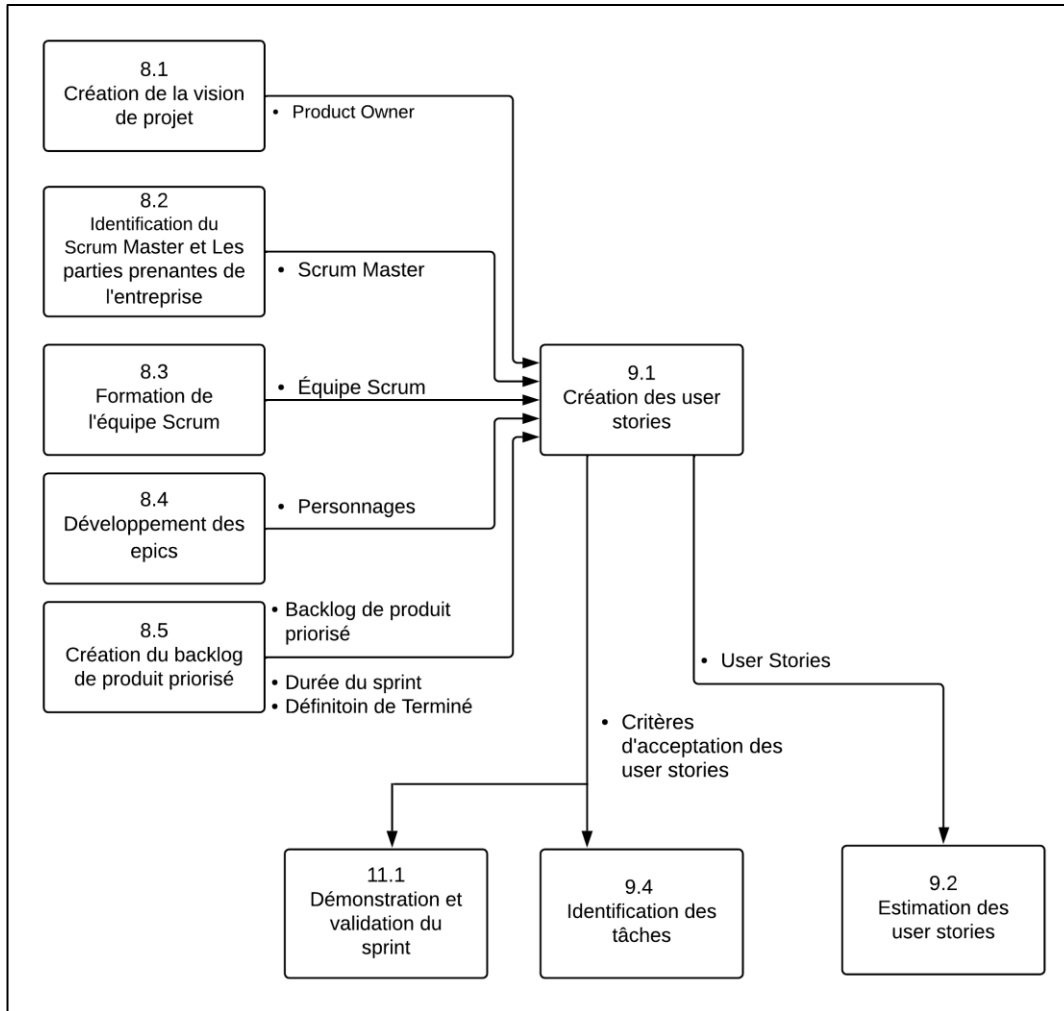


Figure 9-4 : Création des user stories — diagramme de flux de données

9.1.1 Contributions

9.1.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

9.1.1.2 Backlog de produit priorisé*

Décrit dans la section 8.5.3.1.

9.1.1.3 Définition de *terminé**

Décrite dans la section 8.5.3.2.

9.1.1.4 Personnages*

Décrits dans la section 8.4.3.2.

9.1.1.5 La définition de « Ready » (Prêt)

La définition de « Ready » (Prêt) définit les critères qu'une User Story devra remplir avant d'être éventuellement estimée et incluse dans un sprint.

9

9.1.1.6 Parties prenantes de l'entreprise

Décrites dans la section 8.2.3.2.

9.1.1.7 Epics

Décrits dans la section 8.4.3.1.

9.1.1.8 Exigences commerciales

Décrites dans la section 8.5.1.7.

9.1.1.9 Législations et règlements

Décrits dans la section 8.4.1.8.

9.1.1.10 Contrats applicables

Décrits dans la section 8.4.1.9.

9.1.1.11 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

Lors du processus de *création des user stories*, les recommandations du Scrum Guidance Body peuvent inclure des informations sur les règles, législations, standards et meilleures pratiques nécessaires afin de créer des user stories efficaces.

9.1.2 **Outils**

9.1.2.1 **Expertise pour l'écriture des user stories***

Le Product Owner, en fonction de ses interactions avec parties prenantes de l'entreprise, de ses connaissances ainsi que de son expertise commerciale, et des contributions de l'équipe, développe les user stories qui formeront le backlog de produit priorisé initial pour le projet. Le backlog de produit priorisé représente la somme totale de tout ce qui doit être réalisé pour le projet. L'objectif de cet exercice est de créer des user stories élaborées et affinées qui peuvent être estimées et que l'équipe Scrum peut s'engager à réaliser. Le Product Owner peut parfois faire appel à un analyste commercial afin d'aider à l'écriture des user stories.

Bien que le Product Owner soit le principal responsable de l'écriture des user stories et qu'il effectue souvent cet exercice seul, un atelier d'écriture des user stories peut être organisé, le cas échéant.

9.1.2.2 **Atelier des user stories**

Décrit dans la section 8.4.2.2.

9.1.2.3 **Réunions de groupe d'utilisateurs**

Décrites dans la section 8.4.2.1.

9.1.2.4 **Réunions de groupe thématique**

Les réunions de groupe thématique sont une technique qualitative permettant de jauger et de comprendre les besoins et les attentes des utilisateurs concernant le produit proposé. Un petit groupe d'utilisateurs est choisi pour former le groupe thématique.

Les membres du groupe peuvent être sélectionnés de façon aléatoire à partir d'un large panel d'utilisateurs ou ils peuvent être spécialement choisis afin de représenter les principaux personnages ciblés. Les réunions de groupe thématique respectent normalement un certain format où il est demandé aux membres du groupe de répondre à des questions dont ils doivent débattre entre eux. Chaque réunion de groupe thématique peut disposer de ses propres règles de discussions décidées par les organisateurs. Ces réunions se déroulent généralement en présence d'un modérateur.

9.1.2.5 Entretiens d'utilisateurs ou de clients

Décrits dans la section 8.4.2.4.

9.1.2.6 Questionnaires

Décrits dans la section 8.4.2.5.

9.1.2.7 Expertise du Scrum Guidance Body

Décrite dans la section 8.4.2.7.

Lors de la création des user stories, l'expertise du Scrum Guidance Body peut faire référence aux règles et législations documentées ou aux standards et meilleures pratiques relatifs à la création des user stories. Il peut également y avoir une équipe d'experts sur le sujet, capable d'assister ou d'offrir des conseils au Product Owner sur la façon de créer les user stories. Cette équipe peut inclure des analystes commerciaux, des architectes principaux, des développeurs expérimentés, des experts Scrum et d'autres personnes expérimentées. Ce groupe d'experts n'est normalement pas l'équipe qui travaillera au développement de ce projet. Ces experts tendent à naviguer de projet en projet et offrent des conseils lorsque cela s'avère nécessaire.

9.1.2.8 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

9.1.3 Résultats

9.1.3.1 User stories*

Les user stories répondent à une structure spécifique prédéfinie et sont une façon simplifiée de documenter les exigences et fonctionnalités souhaitées pour l'utilisateur. Une user story offre trois types d'informations concernant les exigences : Qui, quoi et pourquoi. Les exigences exprimées dans les user stories sont des énoncés courts, simples et faciles à comprendre. Ce format standard prédéfini permet une meilleure communication entre parties prenantes de l'entreprise et une meilleure estimation par l'équipe. Certaines user stories peuvent être trop larges pour être traitées en un seul sprint. Ces larges user stories sont souvent appelées epics. Une fois que les epics apparaissent dans le backlog de produit priorisé pour être réalisées lors d'un sprint à venir, elles sont décomposées en user stories plus petites.

Le backlog de produit priorisé est une liste dynamique qui est continuellement mise à jour du fait de la repriorisation et de l'ajout, de la mise à jour, de l'affinage ou parfois de la suppression des user stories. Ces mises à jour du backlog sont généralement le résultat d'une modification des exigences commerciales.

Voir également la section 8.5.3.1 pour en savoir plus sur le backlog de produit priorisé.

Format des user stories :

En tant que <rôle/personnage>, je dois pouvoir <exigence> de façon à <avantage>.

Exemple d'user story :

En tant qu'administrateur de bases de données, je dois pouvoir rétablir un certain nombre de mises à jour de façon à restaurer la version désirée de la base de données.

9.1.3.2 Critères d'acceptation des user stories*

Chaque user story possède des critères d'acceptation qui lui sont associés. Les user stories sont subjectives et les critères d'acceptation offrent l'objectivité requise afin de déterminer si l'user story est terminée ou non pendant la revue du sprint. Les critères d'acceptation offrent à l'équipe de la clarté sur ce qui est attendu de l'user story. Ils éliminent les ambiguïtés relatives aux exigences et permettent de faire correspondre les attentes. Le Product Owner définit et communique les critères d'acceptation à l'équipe Scrum. Lors des réunions de revue de sprint, les critères d'acceptation fournissent au Product Owner le contexte nécessaire afin de décider si une user story a été réalisée de façon satisfaisante. Il est important et relève de la responsabilité du Scrum Master de s'assurer que le Product Owner ne modifie pas les critères d'acceptation d'une user story validée au milieu d'un sprint.

9.1.3.3 Backlog de produit priorisé mis à jour

Le backlog de produit priorisé créé lors du processus de *création du backlog de produit priorisé* est mis à jour avec les informations relatives aux user stories, aux epics, à l'estimation des user stories et aux critères d'acceptation des user stories.

Le backlog de produit priorisé est décrit dans la section 8.5.3.1.

9.1.3.4 Personnages mis à jour ou affinés

Les personnages sont initialement créés lors du processus de *développement des epics*. Lorsqu'elle écrit les user stories, l'équipe Scrum peut parvenir à la décision collective que certains personnages initiaux sont inappropriés et nécessitent d'être affinés. S'il est nécessaire d'affiner des personnages, cela est normalement fait vers la fin du processus de *création des user stories*.

Les personnages sont décrits dans la section 8.4.3.2.

9.2 Estimation des user stories

Dans ce processus, l'équipe Scrum, soutenue par le Scrum Master, estime les User Stories et identifie les efforts requis pour développer la fonctionnalité décrite dans chaque User Story. Seules les User Stories qui satisfont à la définition de « Ready » (Prêt) et qui sont correctement définies par le Product Owner sont estimées par l'équipe.

Le Product Owner ne joue pas un rôle actif dans l'estimation des User Stories, mais il peut être consulté pour clarifier toute question que l'équipe Scrum pourrait avoir concernant les User Stories en cours d'estimation.

La figure 9-5 illustre les contributions, outils et résultats du processus d'*estimation des user stories*.

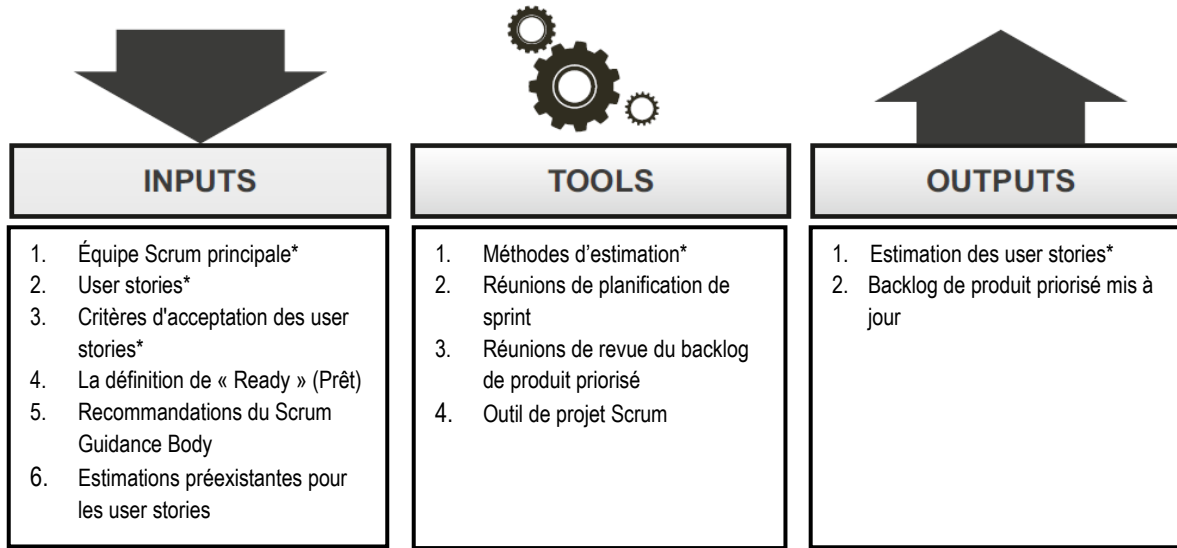


Figure 9-5 : Estimation des user stories — Contributions, outils et résultats

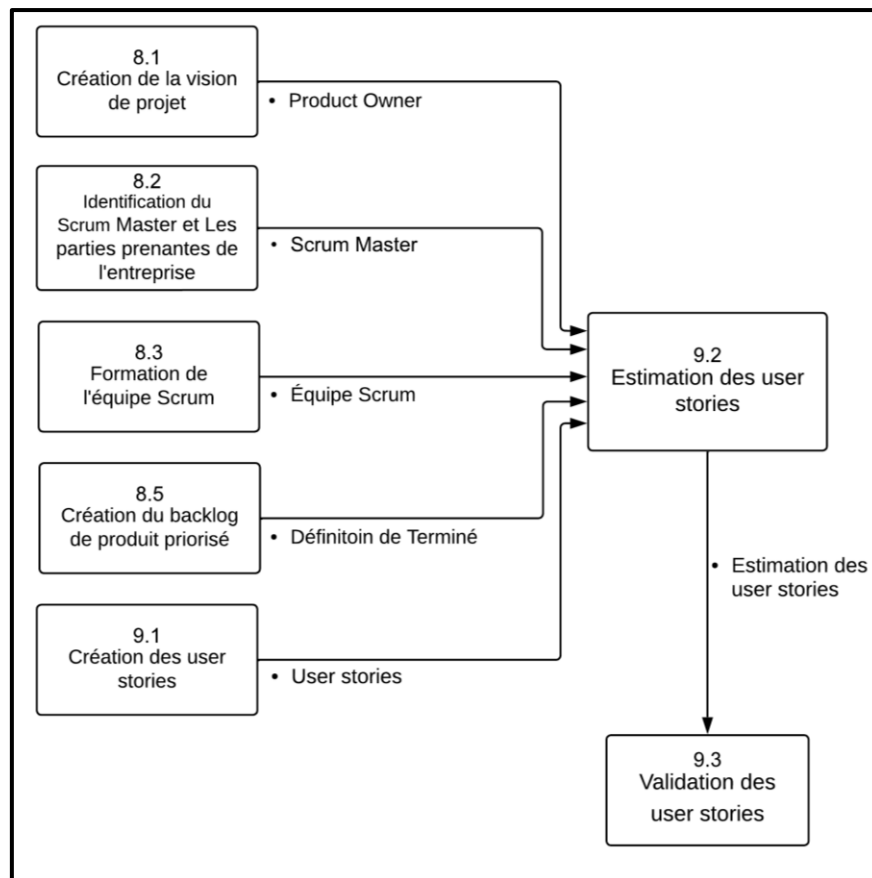


Figure 9-6 : Estimation des user stories — diagramme de flux de données

9.2.1 Contributions

9.2.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

9.2.1.2 User stories*

Décrites dans la section 9.1.3.1.

9.2.1.3 Critères d'acceptation des user stories*

Décrites dans la section 9.1.3.2.

9.2.1.4 La définition de « Ready » (Prêt)*

La définition de « Ready » (Prêt) définit les critères qu'une User Story devra remplir avant d'être éventuellement estimée et incluse dans un sprint.

9.2.1.5 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Lors du processus d'*estimation des user stories*, les recommandations du Scrum Guidance Body peuvent inclure des informations sur les règles, législations, standards et meilleures pratiques nécessaires afin d'estimer les user stories de façon efficace.

9.2.1.6 Estimations préexistantes pour les user stories

Certaines estimations des efforts nécessaires pour exécuter les User Stories existent peut-être déjà, en particulier si des User Stories similaires ont déjà été réalisées dans le cadre du même projet ou de projets antérieurs apparentés. L'effort et le temps pris pour réaliser des User Stories similaires peuvent être utilisés pour dériver des estimations du temps nécessaire pour terminer les User Stories existantes. Les experts qui ont implémenté des exigences de projet similaires dans le passé peuvent également être en mesure de fournir des estimations d'efforts pour les User Stories. Ces estimations préexistantes aideront l'équipe Scrum à estimer de manière approfondie et à valider les User Stories pour le sprint.

Il est important de s'assurer que l'équipe Scrum crée ses propres estimations des User Stories avant que celles-ci ne soient validées dans le cadre du sprint, au lieu de se fier uniquement aux estimations préexistantes disponibles pour les User Stories.

9.2.2 Outils

9.2.2.1 Méthodes d'estimation*

Au fur et à mesure que de nouvelles User Stories ou mises à jour sont raffinées dans le Product Backlog, l'équipe Scrum attribue ou met à jour les estimations préexistantes pour chaque User Story. Le dimensionnement relatif ou les story points peuvent être utilisés pour estimer la taille globale d'une User Story ou d'une fonctionnalité. Cette approche attribue une valeur en story point basée sur l'évaluation globale de la taille d'une User Story en tenant compte des risques, des efforts requis pour sa réalisation et de son niveau de complexité. Cette évaluation sera effectuée par l'équipe de Scrum et une valeur de point de récit sera attribuée. Une fois qu'une estimation est faite sur une User Story dans le Product Backlog priorisé, l'équipe Scrum peut alors estimer les autres User Stories par rapport à cette première Story. Il convient de noter que le calibrage des points d'histoire pour chaque équipe est différent, de sorte que le nombre de points de User Stories achevés ne peut pas être utilisé pour la comparaison entre les équipes. En outre, la méthode d'estimation choisie dépend du niveau de détail de l'estimation requis par l'équipe.

Il existe plusieurs méthodes d'estimation pouvant être utilisées comme outils afin d'estimer les user stories. Quelques outils importants incluent :

1. Méthode Wideband Delphi

La méthode Delphi est une technique d'estimation de groupe permettant de déterminer le volume de travail requis et le temps nécessaire à sa réalisation. Chaque personne au sein de l'équipe fournit anonymement ses estimations pour chaque caractéristique et les estimations initiales sont tracées sur un diagramme. L'équipe discute ensuite des facteurs qui ont influencé leurs estimations et poursuit avec une seconde vague d'estimations. Ce processus est répété jusqu'à ce que les estimations individuelles soient proches les unes des autres et que l'équipe puisse parvenir à un consensus sur l'estimation finale.

2. Planning Poker

Le planning poker, également appelé poker d'estimation est un dérivé de la méthode Wideband Delphi. C'est une technique d'estimation qui utilise le consensus afin d'estimer la taille relative des user stories ou les efforts requis afin de les créer.

Dans le planning poker, chaque membre de l'équipe se voit attribuer un jeu de cartes. Chaque carte est numérotée en séquence et les nombres représentent la complexité du problème, telle qu'elle est estimée par le membre de l'équipe, en termes de temps ou d'efforts.

Les membres de l'équipe Scrum évaluent l'élément (user story ou tâche) afin de mieux le comprendre avant d'offrir leur estimation pour son développement. Puis, chaque membre choisit dans la pile une carte qui représente son estimation pour l'élément. Si la majorité ou tous les membres de l'équipe choisissent la même carte, alors l'estimation indiquée sur la carte sera l'estimation pour cet élément. S'il n'y a pas de consensus, alors les membres de l'équipe discutent des raisons qui les ont poussés à choisir différentes cartes ou estimations. Après cette discussion, ils choisissent à nouveau une carte. Cette séquence continue jusqu'à ce que les présuppositions soient comprises, les incompréhensions résolues et que l'équipe parvienne à un consensus ou un accord.

Le planning poker revendique une meilleure interaction et une communication améliorée entre les participants. Il encourage les participants à réfléchir de façon indépendante, permettant ainsi d'éviter le phénomène de la pensée unique.

3. Fist of five

Le fist of five est un mécanisme simple et rapide pouvant être utilisé comme méthode d'estimation ou comme technique générale permettant de parvenir à un consensus en groupe. Après une discussion initiale sur l'estimation d'un élément donné, les membres de l'équipe Scrum doivent voter sur une échelle allant de 1 à 5 en utilisant leurs doigts. Utilisé comme outil d'estimation, le nombre de doigts affichés indique une valeur d'estimation relative. Les membres de l'équipe qui fournissent les estimations les plus extrêmes (les valeurs les plus basses ou élevées) expliquent le raisonnement de leur estimation au reste du groupe. Ces estimations sont ensuite débattues. Une fois que l'équipe a fini de débattre, celle-ci mène un autre tour de fist of five, ou prend une décision collective.

L'intérêt associé à l'utilisation de cette technique ne relève pas seulement de la recherche de consensus, elle permet également de motiver la discussion car les membres de l'équipe doivent expliquer le raisonnement derrière leurs estimations. Ils ont également l'opportunité d'exprimer leurs problèmes ou leurs inquiétudes. Utilisée comme technique générale pour la recherche de consensus, la proposition ou décision en suspens en cours d'examen est initialement discutée, puis les membres de l'équipe votent en fonction de leur degré de convergence et leur désir de débattre.

Un doigt : Je ne suis pas d'accord avec la conclusion du groupe et ai des inquiétudes importantes.

Deux doigts : Je ne suis pas d'accord avec la conclusion du groupe et aimerais discuter que quelques problèmes mineurs.

Trois doigts : Je ne suis pas sûr et aimerais suivre le consensus du groupe.

Quatre doigts : Je suis d'accord avec la conclusion du groupe et aimerais discuter de quelques problèmes mineurs.

Cinq doigts : Je suis entièrement d'accord avec la conclusion du groupe.

4. Estimation d'affinité

L'estimation d'affinité, comme la taille d'un T-shirt, est une technique utilisée afin d'estimer rapidement un grand nombre d'user stories. En utilisant des feuillets autocollants ou des cartes d'index et du ruban adhésif, l'équipe dispose les user stories sur un mur, ou sur une autre surface, de la plus petite à la plus large. Pour cela, chaque membre de l'équipe débute avec un sous-ensemble d'user stories, tirées du backlog de produit priorisé, et à organiser par taille relative. Ce classement initial se fait en silence. Une fois que chacun a disposé ses user stories sur le mur, l'équipe évalue les classements et peut décider de déplacer les user stories si cela est nécessaire. La deuxième partie de cet exercice implique des discussions. Enfin, le Product Owner indique des catégories de taille sur le mur. Ces catégories peuvent être petites, moyennes ou larges, ou peuvent être numérotées avec des valeurs en story points afin d'indiquer leur taille relative. Pour la dernière étape de ce processus, l'équipe dispose les user stories dans ces catégories. Les principaux avantages de cette approche sont sa transparence, sa visibilité grand public ainsi que sa facilité de mise en œuvre.

9.2.2.2 Réunions de planification de sprint

Lors des réunions de planification de sprint, les user stories sont reprises afin d'être discutées par l'équipe Scrum principale. Si cela n'a pas déjà été fait lors de la création de l'affinage du backlog de produit, chaque user story est évaluée et se voit attribuer une estimation de haut niveau sur la base des story points relatifs.

Voir également les sections 9.3.2.1, 9.4.2.1, 9.5.2.1 et 9.6.2.1.

9.2.2.3 Réunions de revue du backlog de produit priorisé

Des réunions de révision de Product Backlog priorisé sont organisées dans le cadre du processus de raffinement du Product Backlog priorisé. Les informations provenant de ces réunions apportent des précisions supplémentaires sur les User Stories et aident à déterminer leurs estimations.

9.2.2.4 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

9.2.3 Résultats

9.2.3.1 User stories estimées*

Après que les User Stories sont estimées par l'équipe Scrum en utilisant les différentes techniques d'estimation abordées dans cette section, elles sont considérées comme des User Stories estimées.

9.2.3.2 Backlog de produit priorisé mis à jour

Décrit dans la section 9.1.3.3.

9.3 Validation des user stories

Dans ce processus, l'équipe Scrum s'engage à livrer un ensemble de User Stories pour le sprint. Le choix des User Stories est basé sur la priorité relative basée sur la valeur des User Stories ainsi que l'effort estimé et la vélocité de l'équipe pour un sprint. Dans le cadre de ce processus, l'équipe Scrum commence à créer le Sprint Backlog, qui contient les User Stories validées qui sont assignées à un sprint particulier. Le backlog est raffiné avec des détails au niveau des tâches au fur et à mesure de la planification du sprint.

Avec la validation de l'équipe Scrum donné au début d'un sprint dans le cadre de la planification du sprint, le contenu du sprint est défini et ne peut être modifié une fois que la phase d'implémentation du sprint a commencé.

La figure 9-7 illustre les contributions, outils et résultats du processus de *validation des user stories*.

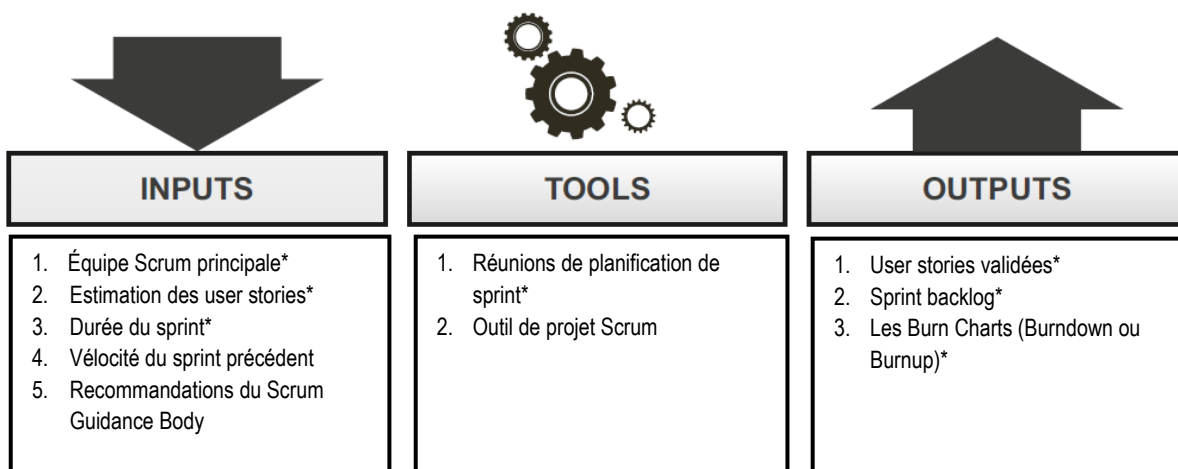


Figure 9-7 : Validation des user stories — Contributions, outils et résultats

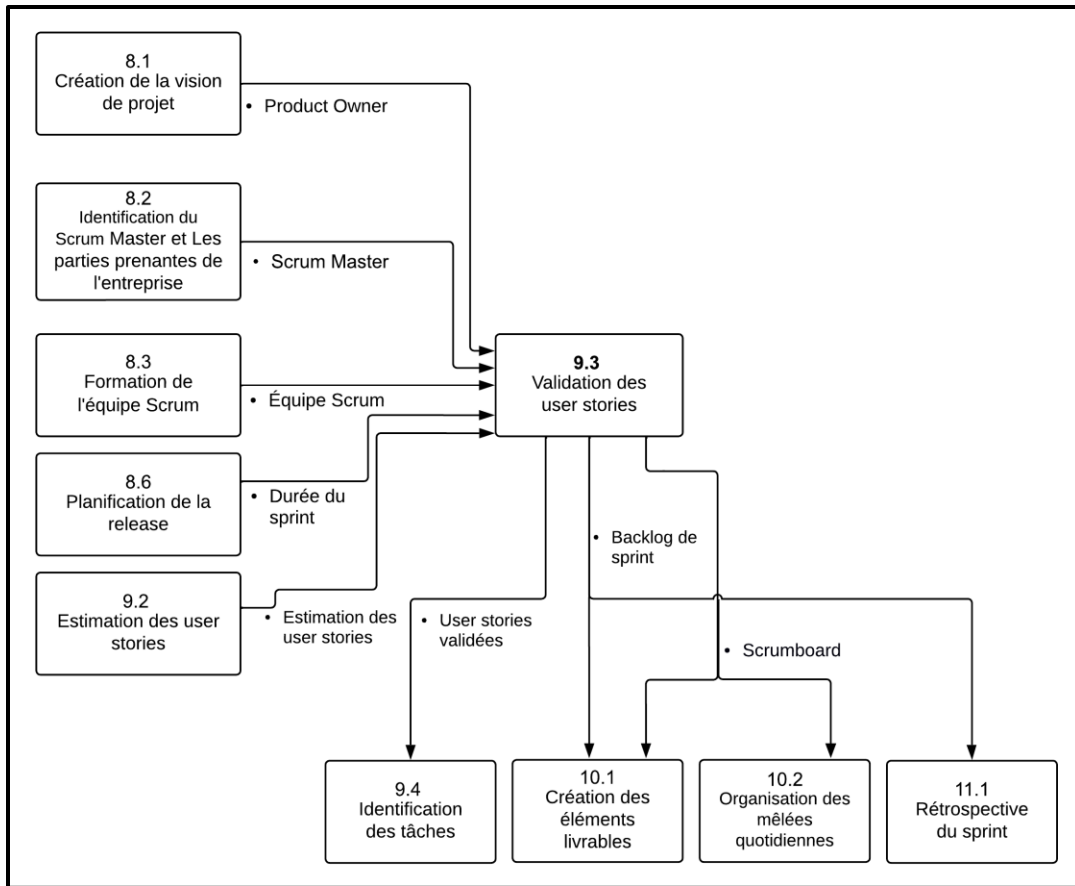


Figure 9-8 : Validation des user stories — diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

9.3.1 Contributions

9.3.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

9.3.1.2 User stories estimées*

Décrites dans la section 9.2.3.1.

9.3.1.3 Durée du Sprint*

Décrite dans la section 8.6.3.2.

9.3.1.4 Vitesse du sprint précédent

La vitesse de sprint mesure la capacité de l'équipe à réaliser le travail d'un sprint. Elle est généralement exprimée dans la même unité que celle utilisée pour les estimations, normalement en story points ou en durée idéale. Un registre de la vitesse de sprint de l'équipe pour chaque sprint est maintenu et utilisé comme référence lors des futurs sprints. La vitesse des sprints précédents devient le facteur le plus important afin de déterminer la quantité de travail sur laquelle l'équipe peut s'engager lors d'un prochain sprint. Tout changement en termes de situation ou de conditions depuis le dernier sprint est pris en compte afin de garantir une estimation précise de la vitesse de sprint pour le prochain sprint.

9.3.1.5 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Lors du processus de *validation des user stories*, les recommandations du Scrum Guidance Body peuvent inclure des informations sur les règles, législations, standards et meilleures pratiques nécessaires afin de valider les user stories de façon efficace.

9.3.2 Outils

9.3.2.1 Réunions de planification de sprint*

Lors des réunions de planification de sprint, l'équipe Scrum se réunit afin de planifier le travail à réaliser durant le sprint. L'équipe passe en revue les user stories estimées en haut du backlog de produit priorisé. Le Product Owner est présent lors de cette réunion au cas où une clarification des user stories ou des priorités est nécessaire. Afin de garantir que le groupe reste dans le sujet, la durée de cette réunion doit être limitée, la durée standard est de deux heures par semaine de sprint. Cela permet d'éviter la dérive des discussions vers des sujets qui devraient en fait être abordés lors d'autres réunions telles que les réunions de planification de la release ou de revue de sprint. Au cours de cette réunion, la totalité de l'équipe Scrum s'engage à réaliser un sous-ensemble d'user stories du backlog de produit priorisé au cours du sprint.

Voir également les sections 9.2.2.1, 9.4.2.1, 9.5.2.1 et 9.6.2.1.

9.3.2.2 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

9.3.3 Résultats

9.3.3.1 User stories validées*

En fonction de sa vélocité, l'équipe Scrum s'engage sur un sous-ensemble d'user stories estimées qu'elle pense être capable de réaliser durant le prochain sprint. Les user stories validées doivent toujours être sélectionnées en fonction des priorités établies par le Product Owner.

9.3.3.2 Sprint Backlog

La liste des User Stories à exécuter par l'équipe Scrum dans le prochain Sprint est appelée Sprint Backlog. Il s'agit d'un sous-ensemble du Product Backlog priorisé qui contient les User Stories validées qui sont assignées à un sprint particulier. Le Sprint Backlog sera raffiné avec des détails au niveau des tâches au fur et à mesure de la planification du sprint.

Il est habituel de représenter les User Stories du Sprint Backlog (et les tâches associées) sur un Scrumboard ou un tableau de tâches similaire, pour fournir une représentation constamment visible de l'état actuel des User Stories dans le Product Backlog.

Une fois que le Sprint Backlog est finalisé et validé par l'équipe Scrum, aucune nouvelle User Story n'est ajoutée. Si de nouvelles exigences apparaissent pendant un sprint, elles doivent être ajoutées au Product Backlog priorisé afin qu'elles soient prises en compte lors d'un prochain sprint.

9.3.3.3 Scrumboard*

Il est important de suivre la progression d'un sprint et de savoir où en est l'équipe Scrum en termes de réalisation des User Stories (et des tâches) dans le Sprint Backlog. Il existe différents outils permettant de suivre la progression d'un sprint mais le plus courant est le Scrumboard, aussi connu sous le nom de tableau des tâches (task board) ou de tableau d'évolution. La transparence de Scrum provient d'outils d'information ouvertement consultables comme le Scrumboard, qui montre les progrès continus de l'équipe. L'équipe utilise un Scrumboard pour planifier et suivre la progression de chaque sprint.

Dans la version la plus élémentaire d'un Scrumboard, le tableau est divisé en trois sections : « Travail non commencé » (également appelé « À faire »), « Travail en cours » (également appelé « En cours ») et « Travail terminé » (également appelé « Terminé »). Des notes autocollantes représentant chaque User Story et les tâches qui y sont associées sont placées dans la catégorie appropriée pour refléter l'état du travail à accomplir dans le sprint en cours. Les notes indiquant les tâches sont déplacées vers la catégorie suivante au fur et à mesure de l'avancement des travaux.

Un Scrumboard classique est illustré à la figure 9-9. Le Scrumboard affiche toutes les User Stories dans la colonne de gauche et comporte trois colonnes intitulées « À faire » (To do), « En cours » (In Progress) et

« Terminé » (Complete). Au fur et à mesure que les tâches associées aux User Stories sont identifiées et traitées dans les processus ultérieurs, ces tâches sont représentées dans leurs colonnes respectives.

| USER STORIES | | TÂCHES | | |
|--------------|---------|----------|-------|---------|
| 1 | À faire | En cours | Tests | Terminé |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

Figure 9-9: Scrumboard classique

Des variantes du Scrumboard classique peuvent être utilisées pour décrire plus précisément l'état d'avancement du travail de l'équipe. Par exemple, une variante contient une colonne supplémentaire « Tests » (Testing) pour indiquer que la tâche est terminée mais que le résultat du travail est en cours de test. Dans ce cas, la colonne « Terminé » est utilisée pour représenter les tâches qui sont entièrement terminées et testées avec succès. Au lieu d'une colonne « Tests », l'équipe peut inclure toute autre colonne sur le Scrumboard qu'elle juge utile pour suivre les progrès. Un exemple de Scrumboard à quatre colonnes est présenté à la figure 9-10.

| USER STORIES | | TÂCHES | | |
|--------------|---------|----------|-------|---------|
| 1 | À faire | En cours | Tests | Terminé |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

Figure 9-10: Scrumboard avec quatre sections

Le Scrumboard peut être tenu manuellement sur papier ou sur un grand tableau blanc, ou bien électroniquement dans une feuille de calcul ou à l'aide d'un outil de projet Scrum. L'équipe doit modifier ou terminer le Scrumboard selon les besoins, afin que celui-ci continue à fournir des informations visuelles précises et un contrôle de l'état d'avancement du travail en cours (tel que convenu et validé par l'équipe).

9.4 Identification des tâches

Dans ce processus, les User Stories validées sont décomposées en tâches spécifiques et compilées dans une liste de tâches. L'identification des tâches peut se faire soit au début du sprint pour toutes les User Story validées, soit avant que l'équipe ne commence à travailler sur les tâches requises pour chaque User Story.

Le Product Owner ne joue pas un rôle actif dans l'identification des tâches, mais doit être disponible pour répondre à toute question de l'équipe Scrum qui pourrait survenir lors de la décomposition des User Stories en tâches.

La figure 9-9 illustre les contributions, outils et résultats du processus d'identification des tâches.



Figure 9-11 : Identification des tâches — Contributions, outils et résultats

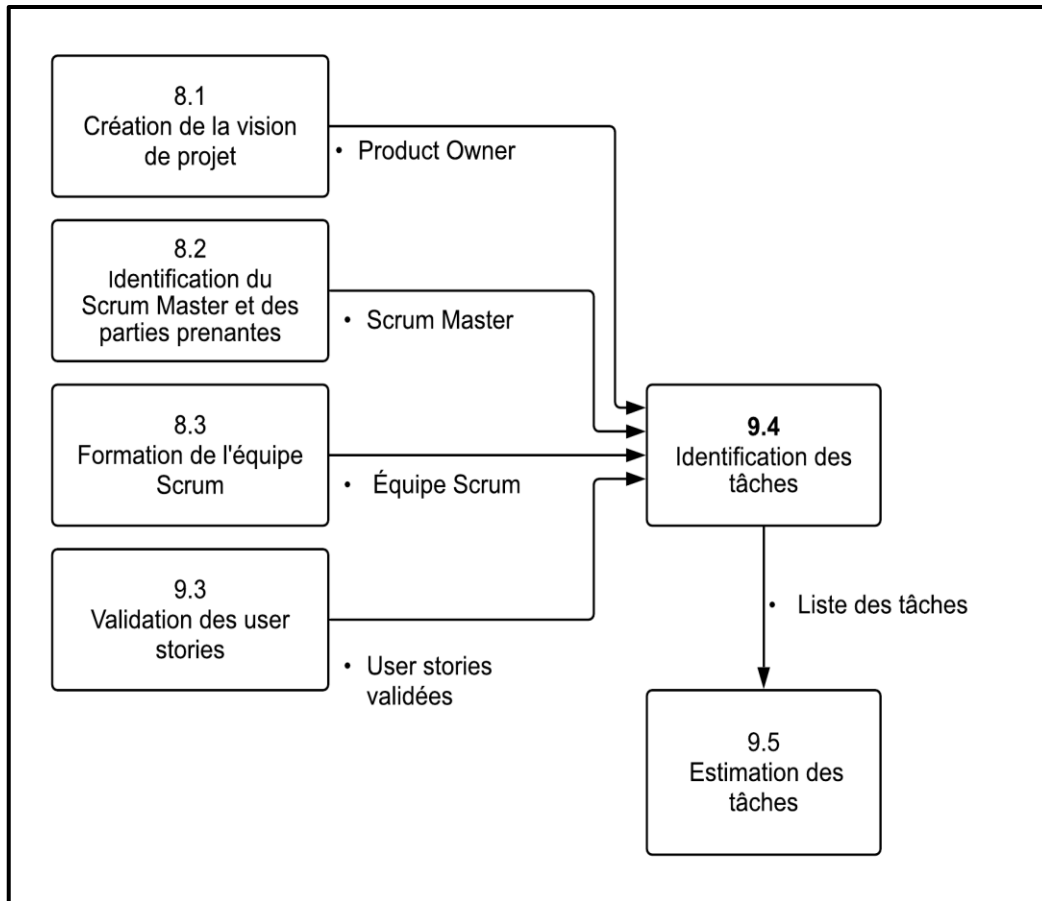


Figure 9-12 : Identification des tâches — Diagramme de flux de données

9.4.1 Contributions

9.4.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

9.4.1.2 User stories validées*

Décrites dans la section 9.3.3.1.

9.4.1.3 Critères d'acceptation des user stories*

Description dans la section 9.1.3.2.

Le Product Owner doit s'assurer que les critères d'acceptation définis sont appropriés pour les User Stories et doit également fournir des éclaircissements sur les exigences à l'équipe Scrum. La compréhension des critères d'acceptation par l'équipe Scrum l'aide à déterminer quelles tâches sont nécessaires pour satisfaire aux exigences de la User Story.

Les tests d'acceptation se réfèrent à l'évaluation de la capacité du produit livrable terminé à répondre à ses critères d'acceptation. Cela fournit des informations au Product Owner pour l'aider à prendre la décision d'approuver ou de rejeter les produits livrables.

Les critères d'acceptation doivent être clairs, non ambigus et spécifiques. Ils doivent être définis de manière à ce que l'équipe soit en mesure de vérifier que les résultats sont conformes aux buts et objectifs de l'organisation commanditaire.

9

9.4.2 Outils

9.4.2.1 Réunions de planification de sprint*

Lors des réunions de planification de sprint, l'équipe Scrum se réunit afin de planifier le travail à réaliser durant le sprint. L'équipe passe en revue chacune des user stories validées pour le sprint et identifie les activités réalisables ou les tâches requises afin de développer les éléments livrables nécessaires pour satisfaire l'user story et remplir les critères d'acceptation. Le Product Owner assiste à cette réunion au cas où des clarifications sont requises concernant les user stories validées afin d'aider l'équipe à prendre des décisions de conception.

Voir également les sections 9.2.2.1, 9.3.2.1, 9.5.2.1 et 9.6.2.1.

9.4.2.2 Décomposition

La décomposition est un outil permettant de diviser les tâches importantes en plusieurs petites tâches plus détaillées. La décomposition est utilisée par l'équipe Scrum pour décomposer les User Story du prochain sprint en tâches détaillées.

9.4.2.3 Détermination des dépendances

Une fois que l'équipe Scrum a sélectionné les user stories pour un sprint spécifique, elle doit ensuite prendre en compte toutes les dépendances qui y sont associées, y compris celles relatives à la disponibilité du personnel ainsi que toutes les dépendances techniques. La bonne documentation des dépendances aide l'équipe Scrum à déterminer l'ordre relatif dans lequel les tâches doivent être effectuées pour créer les incréments de produit du sprint. Les dépendances mettent aussi en relief les relations et interactions entre les tâches à la fois au sein de l'équipe Scrum travaillant sur un sprint spécifique et entre les différentes équipes Scrum au sein du projet.

Il existe plusieurs types de dépendances : obligatoires et facultatives, internes et externes ou une combinaison des deux. Par exemple, une dépendance peut être à la fois obligatoire et externe.

- **Dépendances obligatoires** — ces dépendances sont soit inhérentes à la nature du travail, telles que les contraintes physiques, ou peuvent être dues à des obligations contractuelles ou légales. Par exemple, la construction du rez-de-chaussée d'un bâtiment ne peut pas commencer avant que les fondations n'aient été réalisées. Les dépendances obligatoires sont également souvent décrites comme de la logique pure.
- **Dépendances facultatives** — ce sont des dépendances qui sont ajoutées au flux de travail par choix. Généralement, les dépendances facultatives sont décidées par l'équipe Scrum sur la base des expériences précédentes et des meilleures pratiques pour un domaine ou secteur particulier. L'équipe peut décider de réaliser une tâche avant de travailler sur une autre car cela fait partie des meilleures pratiques mais n'est pas obligatoire. Par exemple, l'équipe peut choisir de construire les menuiseries des portes et des fenêtres avant que la structure complète du mur ne soit en place.
- **Dépendances externes** — les dépendances externes sont les dépendances relatives aux tâches, activités ou produits situés en dehors de la portée du travail à effectuer par l'équipe Scrum mais qui sont nécessaires à la réalisation d'une tâche du projet ou à la création d'un élément livrable. Généralement, les dépendances externes sont hors du contrôle de l'équipe Scrum. Par exemple, si l'équipe Scrum n'est pas responsable de l'acquisition des matériaux requis pour la construction des murs, alors ces matériaux et les tâches associées à leur acquisition sont considérés comme externes.
- **Dépendances internes** — ce sont les dépendances entre les tâches, produits ou activités qui sont sous le contrôle de l'équipe Scrum. Par exemple, la cloison en placoplâtre doit être installée avant de peindre le mur. C'est un exemple de dépendance interne car les deux tâches font partie du projet. Dans ce cas, cette dépendance est également obligatoire car elle repose sur une contrainte physique. Il n'est pas possible de peindre un mur avant que la cloison en placoplâtre ne soit installée.

9.4.2.4 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

9.4.3 Résultats

9.4.3.1 Liste des tâches*

C'est une liste complète qui contient toutes les tâches sur lesquelles l'équipe Scrum s'est engagée pour le sprint en cours. Elle contient une description de chaque tâche et de l'estimation qui en dérive après le processus d'*identification des tâches*. La liste des tâches doit prendre en compte tous les efforts de test et d'intégration de façon à ce que l'incrément de produit du sprint puisse être intégré avec succès aux autres éléments livrables des sprints précédents.

Bien que les tâches soient souvent basées sur les activités, le niveau de granularité auquel les tâches sont décomposées est décidé par l'équipe Scrum.

9.4.3.2 Mises à jour Scrumboard*

Au fur et à mesure que les tâches sont identifiées, le Scrumboard est mis à jour pour montrer les tâches associées à chaque User Story. Les tâches sont généralement indiquées sur des notes autocollantes placées sur un Scrumboard physique ou sous la forme d'entrées dans les User Stories applicables lorsqu'on utilise un outil de projet Scrum électronique. Pendant l'implémentation, à mesure que l'équipe ajoute, assigne et met à jour les tâches sur lesquelles elle travaille, le Scrumboard est constamment mis à jour avec les tâches supplémentaires et l'état d'avancement de chaque tâche. Si l'équipe a estimé les tâches, les estimations des tâches sont également représentées sur le Scrumboard.

Dans l'exemple de la figure 9-13, le Scrumboard montre que trois User Stories. les User Story 1, 2 et 3 ont été décomposées en tâches, mais que la User Story 4 n'a pas encore été décomposée en tâches. Au début d'un Sprint, toutes les tâches de ce Sprint sont placées dans la colonne « À faire » et sont ensuite avancées en fonction de leur évolution. Par exemple, la User Story 1 comporte 7 tâches, toutes classées dans la catégorie « À faire », ce qui indique que l'équipe Scrum n'a commencé à travailler sur aucune de ces tâches.

Pour plus d'informations sur le Scrumboard, voir les sections 9.3.3.3 and 10.1.1.3.




| USER STORIES | TASKS | | |
|--------------|---|--------------------|-----------------|
| | <i>To Do</i> | <i>In Progress</i> | <i>Complete</i> |
| 1 |  | | |
| 2 |  | | |
| 3 |  | | |
| 4 | | | |

Figure 9-13: Scrumboard with Identified Tasks

9.4.3.3 User stories validées mises à jour

Les user stories sont mises à jour au cours de ce processus. Les mises à jour peuvent inclure la révision de l'estimation originale de l'user story en fonction des tâches créées et des facteurs de complexité discutés durant la réunion de planification de sprint. Les user stories validées sont décrites dans la section 9.3.3.

9.4.3.4 Dépendances

Les dépendances décrivent la relation et les interactions entre les différentes tâches d'un projet et peuvent être classifiées comme obligatoires ou facultatives ; internes ou externes ; telles que décrites dans la section 9.4.2.3.

Il existe de nombreuses façon d'identifier, de définir et de présenter les tâches et leurs dépendances. Deux méthodes courantes impliquent d'utiliser des diagrammes de flux de produit et des diagrammes de Gantt.

9.5 Estimation des Tâches

Il s'agit d'un processus optionnel qui implique la création d'estimations de tâches si l'équipe Scrum en voit l'intérêt. Dans ce processus, l'équipe Scrum estime les efforts requis afin d'effectuer chaque tâche sur la liste des tâches. Les estimations des tâches peuvent être déterminées soit au début du Sprint pour toutes les User Stories/tâches pertinentes pour ce sprint ou pour chaque tâche juste avant que l'équipe ne commence à travailler sur la User Story/tâche en question. L'estimation peut être faite en utilisant les mêmes méthodes que celles utilisées pour le processus d'estimation des User Stories.

La figure 9-11 illustre les contributions, outils et résultats du processus d'estimation des tâches.

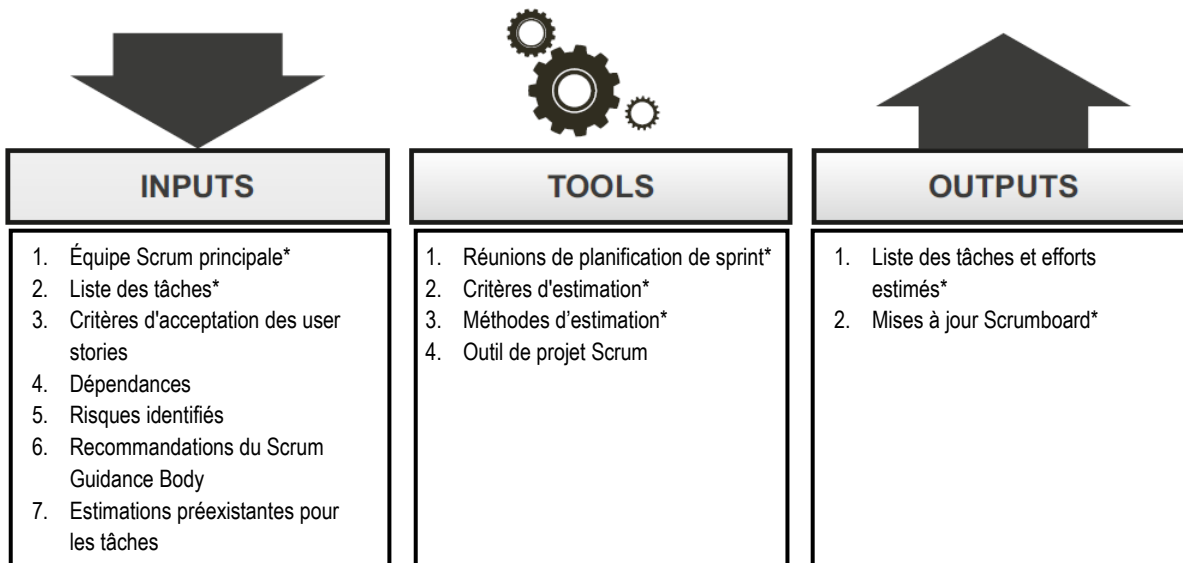


Figure 9-14 : Estimation des tâches — Contributions, outils et résultats

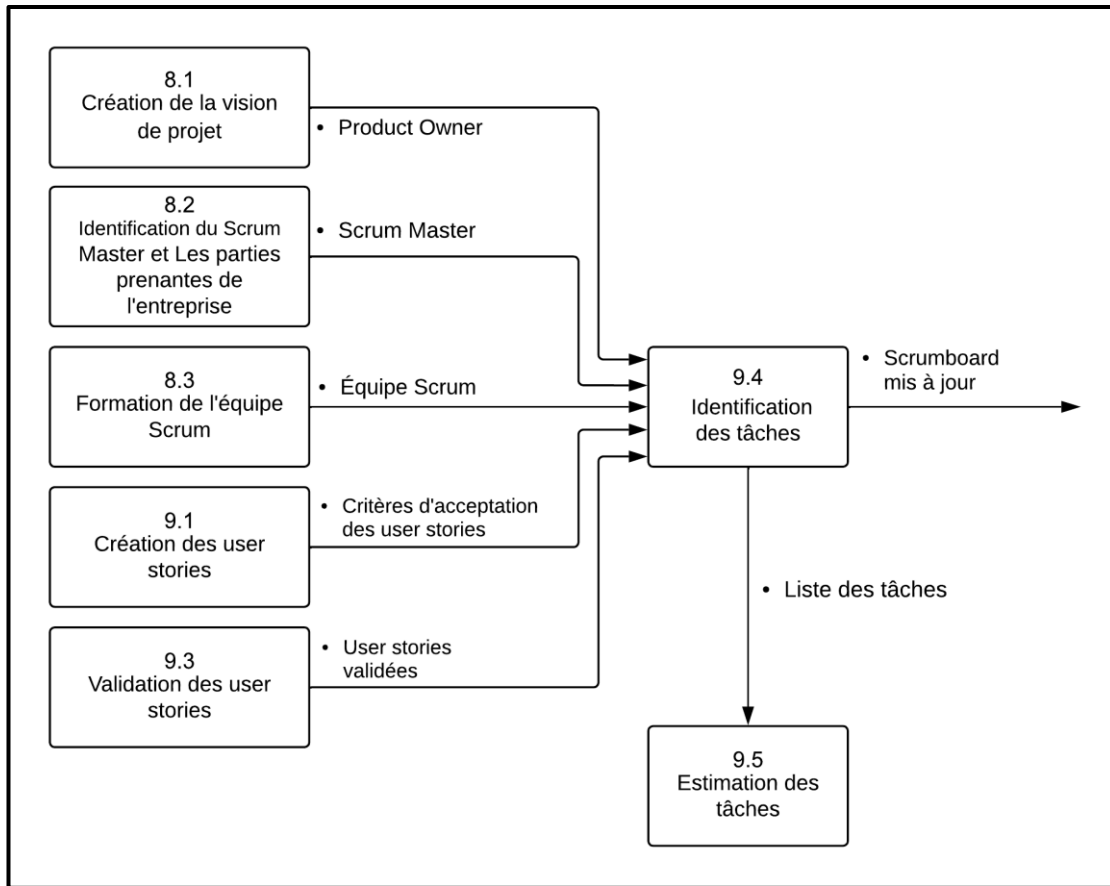


Figure 9-15 : Estimation des tâches — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

9.5.1 Contributions

9.5.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

9.5.1.2 Liste des tâches*

Décrite dans la section 9.4.3.1.

9.5.1.3 Critères d'acceptation des user stories

Décrits dans la section 9.1.3.2.

L'équipe Scrum doit s'assurer que les critères d'acceptation définis sont adéquats pour les user stories et clarifient les exigences pour l'équipe Scrum. Le test d'acceptation fait référence à l'évaluation de la capacité des éléments livrables réalisés à satisfaire les critères d'acceptation. Cela fournit des informations au Product Owner afin d'aider à la prise de décision concernant l'approbation ou le rejet des éléments livrables.

Les éléments suivants doivent être pris en compte lors du développement des critères d'acceptation des user stories :

- Les critères d'acceptation ne doivent pas être vagues, ambigus ou trop généralisés.
- Les critères d'acceptation définis doivent garantir que l'équipe est capable de vérifier que les résultats respectent les buts et objectifs de l'organisation du sponsor.

9.5.1.4 Dépendances

Décrites dans la section 9.4.3.3.

9.5.1.5 Risques identifiés

Décrits dans la section 8.4.3.4.

9.5.1.6 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Lors du processus d'*estimation des tâches*, les recommandations du Scrum Guidance Body peuvent inclure des informations sur les règles, législations, standards et meilleures pratiques nécessaires afin d'estimer les tâches contenues dans la liste des tâches de façon efficace.

9.5.1.7 Estimations préexistantes pour les tâches

Certaines estimations de tâches préexistantes peuvent être prises en compte par les membres de l'équipe Scrum lors de l'estimation des tâches. Des tâches similaires peuvent avoir été réalisées précédemment dans le même projet ou dans d'autres projets antérieurs, et les efforts et le temps nécessaires à la réalisation de ces tâches similaires peuvent aider l'équipe Scrum à obtenir de meilleures estimations des tâches au cours de ce processus. Les experts qui ont travaillé sur des tâches similaires dans le passé peuvent également être en mesure de fournir des estimations des efforts à fournir pour les tâches. Cependant, il est important de s'assurer que l'équipe Scrum crée ses propres estimations de tâches, au lieu de se fier uniquement à des estimations préexistantes. Les estimations de tâches préexistantes peuvent également aider l'équipe Scrum à réévaluer la validation au regard du Product Owner au niveau de la User Story.

9.5.2 Outils

9.5.2.1 Réunions de planification de sprint

Durant les réunions de planification de sprint, l'équipe Scrum estime les efforts requis pour compléter une tâche ou une série de tâches et pour estimer l'effort humain ainsi que les autres ressources requises afin d'effectuer les tâches d'un sprint donné. Les membres de l'équipe Scrum utilisent la liste des tâches afin d'estimer la durée et les efforts nécessaires afin de réaliser les user stories du sprint.

Un des principaux avantages de cette technique est qu'elle permet à l'équipe d'avoir une perspective partagée des user stories et de leurs exigences afin que l'équipe puisse estimer les efforts requis avec fiabilité.

Voir également les sections 9.2.2.1, 9.3.2.1, 9.4.2.1 et 9.6.2.1.

9.5.2.2 Critères d'estimation*

Les critères d'estimation peuvent être exprimés de différentes façons, deux exemples courants sont la valeur en story points et la durée idéale. Les valeurs en story points sont utilisées pour représenter les efforts relatifs ou comparatifs afin de réaliser les tâches. En revanche, la durée idéale décrit normalement le nombre d'heures passées par un membre de l'équipe Scrum à travailler exclusivement au développement des éléments livrables du projet, sans inclure le temps passé sur d'autres activités ou travaux externes au projet. Les critères d'estimation permettent à l'équipe Scrum d'estimer les efforts plus facilement et de répondre aux insuffisances lorsque cela est nécessaire.

9.5.2.3 Méthodes d'estimation*

Décrites dans la section 9.2.2.3.

Les méthodes d'estimation utilisées pour estimer les user stories peuvent également être appliquées aux tâches.

9.5.2.4 Outil de projet Scrum

Décrites dans la section 2.5.3.1.

9.5.3 Résultats

9.5.3.1 Liste des tâches mise à jour*

La liste de tâches développée lors du processus d'*identification des tâches* inclut les estimations initiales des user stories qui doivent être révisées sur la base de l'estimation plus détaillée des activités effectués lors du processus d'*estimation des tâches*. Les réestimations peuvent également être le fruit d'une revue des sprints précédents ou d'un changement dans la compréhension collective des user stories et des exigences par l'équipe Scrum.

9.5.3.2 Mises à jour Scrumboard*

Au fur et à mesure que les tâches sont estimées, les estimations de l'effort sont mises à jour dans le Scrumboard. Pour plus d'informations sur le Scrumboard, voir les sections 9.3.3.3 and 9.4.3.2.

9.6 Mise à jour du Sprint Backlog

Dans ce processus, l'équipe centrale Scrum met à jour le Sprint Backlog avec les détails des tâches et, si elles sont disponibles, les estimations des tâches. Le Sprint Backlog mis à jour sera utilisé dans la phase d'implémentation pour suivre les progrès de l'équipe au cours du prochain sprint.

La figure 9-13 illustre les contributions, outils et résultats du processus de *Mise à jour du Sprint Backlog*.

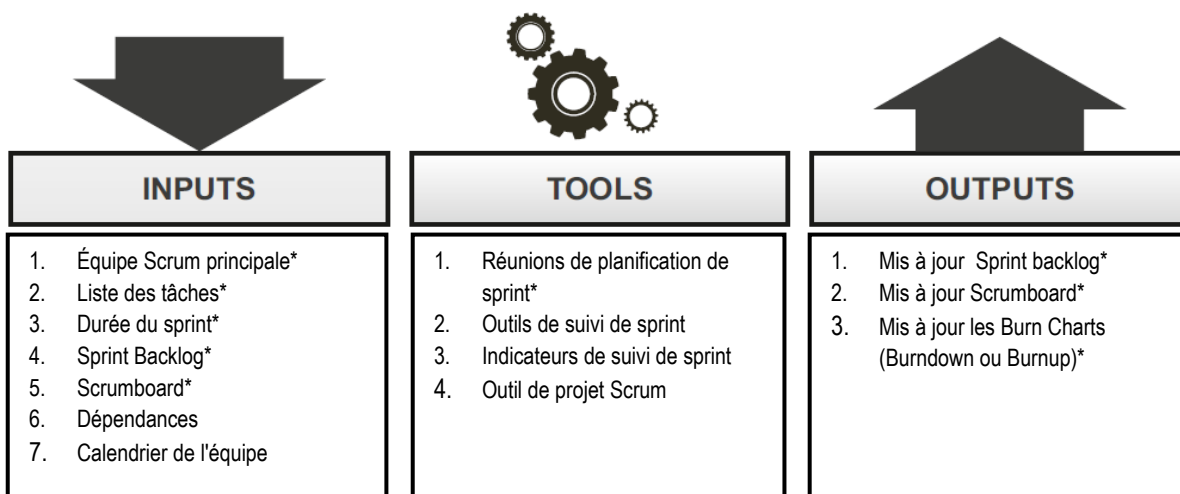


Figure 9-16: Mise à jour du Sprint Backlog — Contributions, outils et résultats

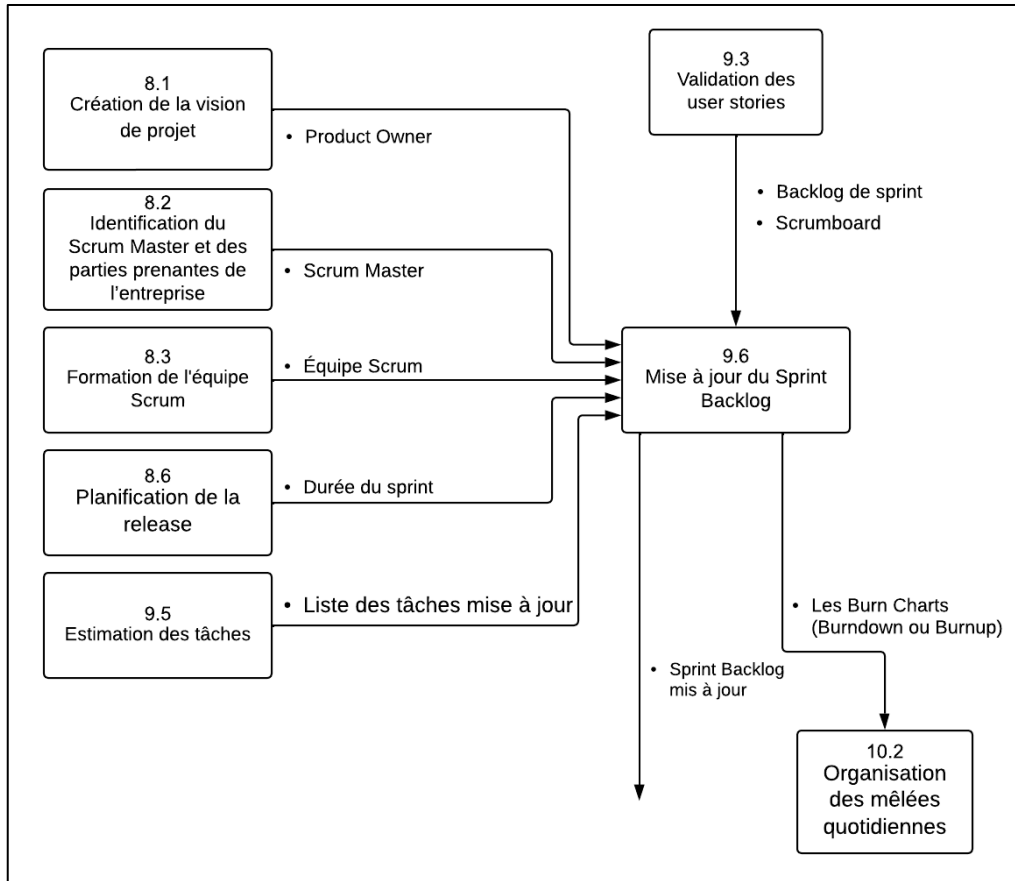


Figure 9-17 : Mise à jour du Sprint Backlog — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

9.6.1 Contributions

9.6.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

9.6.1.2 Liste des tâches estimés*

Décrite dans la section 9.5.3.1.

9.6.1.3 Durée du sprint*

Décrite dans la section 8.6.3.2.

9.6.1.4 **Sprint Backlog***

Décrites dans la section 9.3.3.2.

9.6.1.5 **Scrumboard***

Décrites dans la section 9.3.3.3.

9.6.1.6 **Dependances**

Décrites dans la section 9.4.3.3.

9.6.1.7 **Calendrier de l'équipe**

Un calendrier d'équipe contient des informations au sujet de la disponibilité des membres de l'équipe, dont les informations au sujet des vacances des employés, des congés, des événements importants et des jours fériés.

Un des principaux objectifs relatif à l'utilisation d'un calendrier d'équipe est de suivre ce sur quoi chaque membre de l'équipe travaille tout au long du projet. Cela aide non seulement l'équipe à planifier et exécuter les sprints de manière efficace mais aussi à faire correspondre les sprints avec les dates de release.

9.6.2 **Outils**

9.6.2.1 **Réunions de planification de sprint***

Lors des réunions de planification de sprint, les user stories sont validées pour un sprint et les tâches sont identifiées et estimées par l'équipe Scrum. Chaque membre de l'équipe Scrum utilise également la liste des tâches et efforts estimés afin de sélectionner les tâches sur lesquelles il prévoit de travailler durant le sprint en fonction de ses compétences et de son expérience. L'équipe Scrum crée également le backlog de sprint et le sprint burndown chart en utilisant les user stories et la liste des tâches et efforts estimés lors de réunions de planification de sprint.

Voir également les sections 9.2.2.1, 9.3.2.1, 9.4.2.1 et 9.5.2.1.

9.6.2.2 Outils de suivi de Sprint

Il est important de suivre les progrès d'un sprint et de savoir où en est l'équipe Scrum dans la réalisation des tâches contenues dans le backlog de sprint. Il existe différents outils permettant de suivre la progression d'un sprint mais le plus courant est le scrumboard (aussi connu sous le nom de tableau des tâches ou task board) ou le tableau d'évolution. La version la plus basique d'un scrumboard est divisée en trois sections : Travail « à faire » (parfois appelé « travail pas encore commencé »), travail « en cours » et travail « terminé ». Des feuillets autocollants représentant chaque tâche ou user story sont disposés dans la colonne appropriée afin de refléter le statut du travail. Ils sont déplacés dans la catégorie suivante au fur et à mesure que le travail progresse.

9.6.2.3 Indicateurs de suivi de sprint

Les indicateurs utilisés dans les projets Scrum incluent la vélocité, la valeur commerciale offerte et le nombre de stories.

Vélocité — représente le nombre d'user stories ou le nombre de fonctionnalités livrées en un seul sprint.

Valeur commerciale offerte — mesure la valeur des user stories livrées d'un point de vue commercial.

Nombre de stories — fait référence au nombre de stories réalisées au cours d'un même sprint. Ce nombre peut être exprimé de façon simple ou pondérée.

9.6.2.4 Outil de projet Scrum

Décrites dans la section 2.5.3.1.

9.6.3 Résultats

9.6.3.1 Backlog de sprint mis à jour*

L'équipe centrale Scrum met à jour le Sprint Backlog avec les détails des tâches associées aux User Stories validées dans le Sprint Backlog. Si elles sont disponibles, les estimations des tâches sont également mises à jour dans le Sprint Backlog. Le Sprint Backlog est utilisé dans la phase d'implémentation pour suivre les progrès de l'équipe pendant le sprint. Une fois que le Sprint Backlog est finalisé et validé par l'équipe Scrum, aucune nouvelle User Story ne doit être ajoutée ; cependant, les tâches qui pourraient avoir été oubliées ou négligées dans les User Stories validées peuvent être ajoutées. Si de nouvelles exigences apparaissent au cours d'un sprint, elles seront ajoutées au Product Backlog priorisé et incluses dans un futur sprint et seront prises en compte lors d'un prochain sprint.

9.6.3.2 Scrumboard mis à jour*

Le Scrumboard est mis à jour pour refléter les informations contenues dans le Sprint Backlog mis à jour y compris toute mise à jour des tâches, de leur état et de leur estimation, le cas échéant.

9.6.3.3 Les Burn Charts (Burndown ou Burnup)*

Les Burn Charts (Burndown ou Burnup) sont utilisés pour suivre la progression d'un projet Scrum. Un Burndown Chart est un graphique qui représente la quantité de travail restant par rapport au temps restant. À la différence d'un Burndown Chart, un Burnup Chart décrit ce qui a été réalisé par rapport au temps restant. Les Burn Charts sont utilisés dans la phase d'implémentation pour suivre la progression de l'équipe Scrum pendant un sprint et pour obtenir une indication précoce afin de savoir si une équipe sera en mesure de terminer toutes les User Stories qui ont été validées pour ce sprint. Si les membres de l'équipe pensent qu'ils ne seront pas en mesure de terminer toutes les User Stories validées, ils peuvent prendre des mesures au début du sprint pour obtenir le meilleur résultat possible. Le Sprint Burndown Chart initial montre comment l'équipe envisage d'accomplir le travail. Parce que l'équipe vient de s'engager à produire un ensemble de User Stories et les tâches associées pour le sprint en cours, et parce que l'on s'attend à ce que l'équipe respecte ses engagements, le Burndown initial planifié montre que le dernier jour du sprint, il ne restera aucune tâche à effectuer. Cela signifie que, dans l'idéal, tout le travail sera terminé le dernier jour. Le Burndown Chart doit être mis à jour par l'équipe à la fin de chaque journée pour montrer l'avancement du travail.

La figure 9-18 ci-dessous présente un exemple de Sprint Burndown Chart :

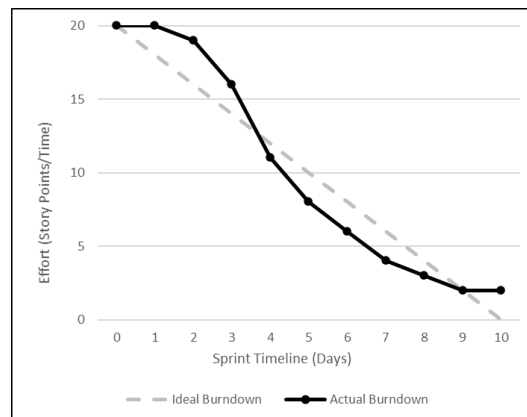


Figure 9-18: Sprint Burndown Chart

Un exemple de Sprint Burnup Chart est illustré à la figure 9-19 ci-dessous :

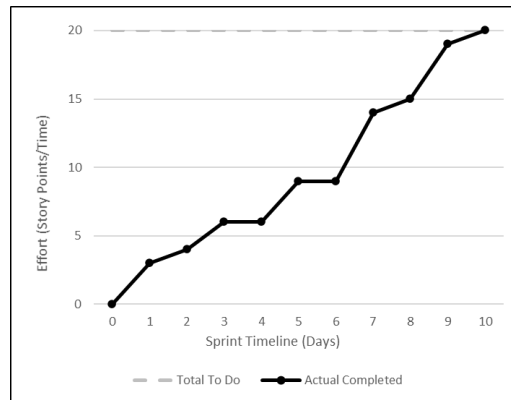


Figure 9-19: Sprint Burnup Chart

9.7 Diagramme de flux de données de la phase de planification et d'estimation

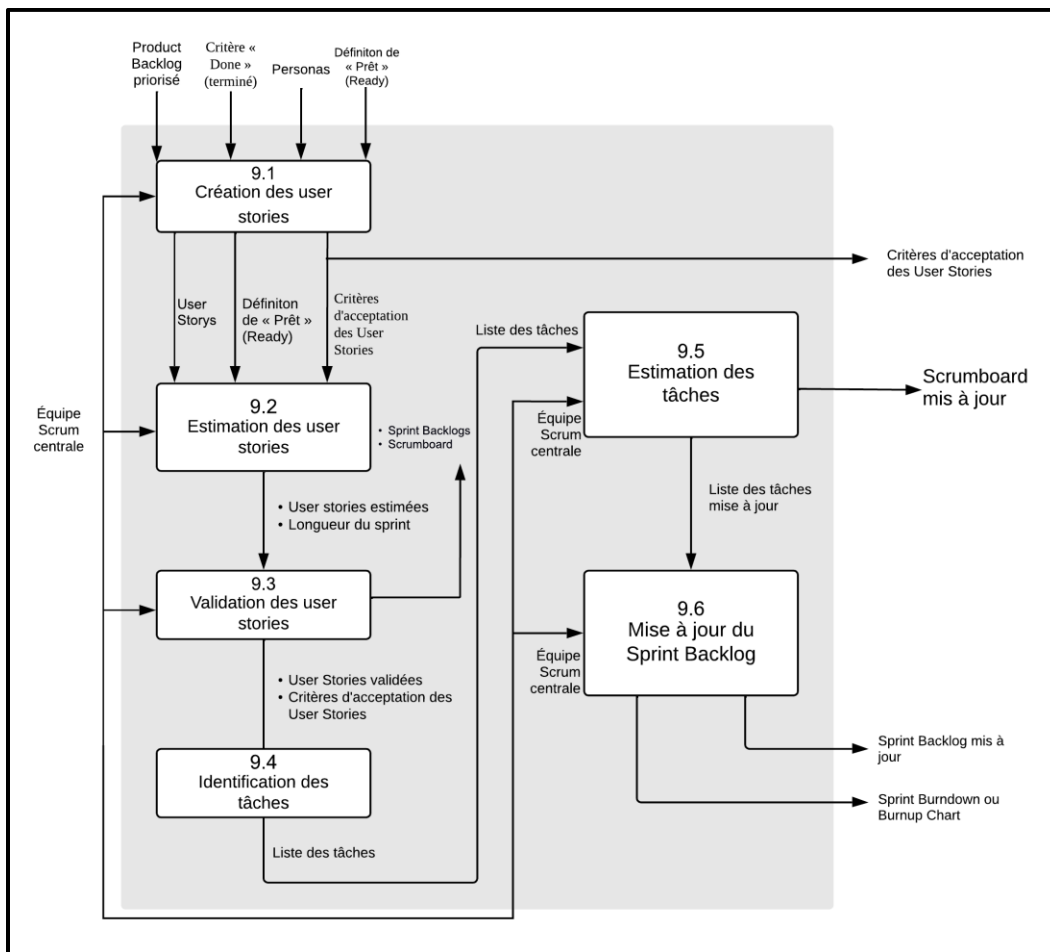


Figure 9-20 : Phase de planification et d'estimation — Diagramme de flux de données

10. IMPLÉMENTATION

La phase d'implémentation est relative à l'exécution des tâches et des activités pour créer le produit d'un projet. Ces activités incluent la création des différents éléments livrables, les mêlées quotidiennes, et l'affinage (c-à-d. la revue, l'ajustement et la mise à jour régulière) du backlog de produit à intervalles réguliers.

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la phase d'implémentation s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Afin de faciliter la meilleure application possible de l'approche Scrum, ce chapitre identifie les contributions, outils et résultats de chaque processus comme étant soit « obligatoires » ou « facultatifs ». Les contributions, outils et résultats accompagnés d'un astérisque (*) sont obligatoires ou essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont facultatifs.

Il est recommandé que l'équipe Scrum et les personnes qui découvrent l'approche et les processus Scrum se concentrent principalement sur les contributions, outils et résultats obligatoires. En revanche, les Product Owners, Scrum Masters et autres praticiens Scrum expérimentés pourront aspirer à améliorer leurs connaissances grâce à l'ensemble des informations contenues dans ce chapitre. Il est également important de noter que bien que tous les processus ne soient définis qu'une seule fois dans le *Guide SBOK®*, ils ne sont pas nécessairement réalisés de façon séquentielle ou séparément. À certains moments, il peut s'avérer utile de combiner certains processus, en fonction des exigences spécifiques de chaque projet.

Ce chapitre est écrit depuis la perspective de l'équipe Scrum travaillant sur un sprint afin de produire des éléments potentiellement livrables, pouvant faire partie d'un plus gros projet, programme ou portefeuille. Des informations supplémentaires concernant le passage à l'échelle Scrum pour les gros projets sont disponibles dans le chapitre 13 et celles sur le passage à l'échelle pour l'entreprise dans le chapitre 14.

Implement is the second of the three phases that are done repetitively in every Sprint. This phase begins after Sprint planning is complete. It is the core of every Scrum project where the bulk of the work is done.

The Scrum Team, facilitated by the Scrum Master, creates the deliverables that are associated with the committed User Stories by working on and completing the tasks the team identified in the previous phase.

Pendant que l'équipe Scrum crée les livrables du Sprint, le Product Owner met à jour et affine le Product Backlog priorisé pour le tenir à jour avec tout changement d'exigences et/ou de priorités et pour s'assurer que l'ensemble des User Stories que le Product Owner comme l'équipe à laquelle s'engager dans le prochain Sprint sera prête pour l'engagement.

Il est également important de réaliser que bien que toutes les phases et tous les processus soient définis de manière unique dans le guide SBOK®, ils ne sont pas nécessairement exécutés séquentiellement ou séparément. Parfois, il peut être plus approprié de chevaucher certaines phases et/ou processus, selon les exigences spécifiques de chaque projet.

10.1 Création des incréments de produit — Dans ce processus, l'équipe Scrum crée les produits livrables du sprint en travaillant sur les tâches du Sprint Backlog. C'est le processus le plus chronophage pour l'équipe et le Scrum Master. L'équipe est soutenue par le Scrum Master, qui anime les réunions de l'équipe, gère les obstacles auxquels elle est confrontée et fait tout ce qu'il peut pour permettre aux membres de l'équipe de se concentrer sur la création des produits livrables du sprint.

L'équipe Scrum utilise un Scrumboard pour suivre sa progression pendant le sprint. L'équipe Scrum utilise les informations sur sa progression pour obtenir une bonne indication de sa capacité à respecter ses engagements et, si nécessaire, pour prendre des mesures afin de garantir le meilleur résultat possible pour le sprint dans les circonstances données.

10.2 Organisation des mêlées quotidiennes — Au cours de ce processus, une mêlée quotidienne de mise au point très ciblée est organisée. Elle est appelée « mêlée quotidienne ». Cette réunion limitée dans le temps permettant aux membres de l'équipe Scrum de se tenir informés de leurs progrès et des obstacles auxquels ils font face.

10.3 Raffinage du Product Backlog priorisé — Au cours de ce processus, le Product Owner met continuellement à jour et maintient le Product Backlog priorisé. Une réunion de revue du backlog de produit priorisé peut être organisée pour discuter des modifications et mises à jour à apporter au backlog, le cas échéant.

Afin de maintenir à jour le Product Backlog priorisé des changements dans les exigences et/ou les priorités, le Product Owner travaille continuellement avec le client et les autres parties prenantes de l'entreprise pour saisir et comprendre tout changement dans leurs besoins.

Pour s'assurer que l'ensemble des User Stories que le Product Owner voudrait voir validées dans le prochain sprint seront prêtes à être validées, le Product Owner affine les Epics et les User Stories existantes dans le Product Backlog priorisé et s'assure que les User Stories répondent à la définition de « Ready » (Prêt).

Dans le cadre du raffinage du Product Backlog priorisé, le Product Owner travaille également avec l'équipe Scrum pour obtenir des commentaires et des questions concernant les mises à jour du Product Backlog priorisé, y compris les estimations.

Si des changements dans les exigences et/ou la progression générale de l'équipe Scrum nécessitent des modifications du calendrier de release et/ou de justification commerciale, le Product Owner apportera également ces modifications au cours de ce processus. C'est le processus le plus chronophage pour le Product Owner.

La figure 10-1 offre un aperçu des processus de la phase d'implémentation qui sont les suivants :

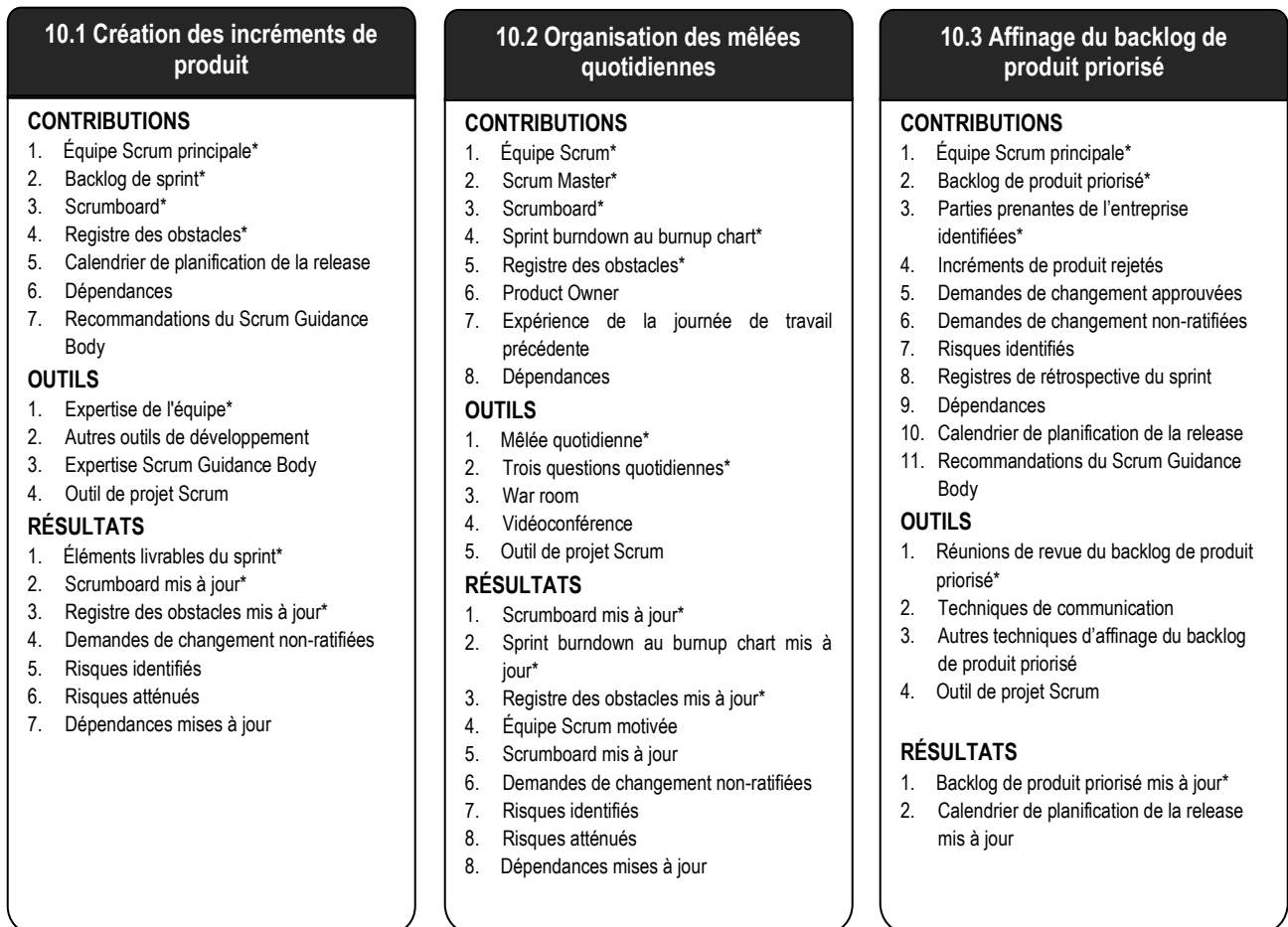


Figure 10-1 : Aperçu de la phase d'implémentation

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus

La figure 10-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de la phase d'implémentation.

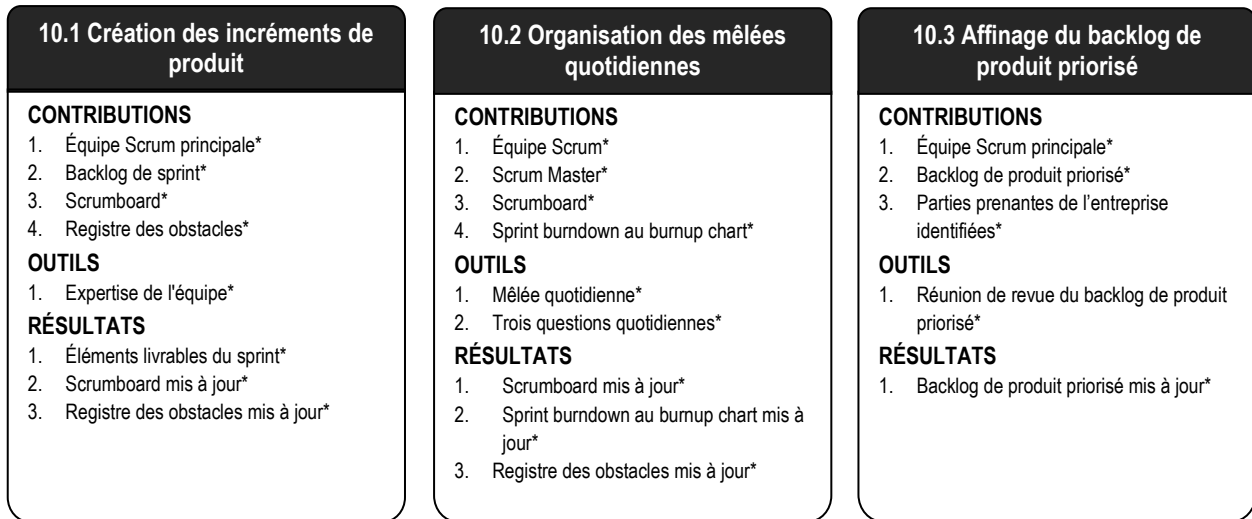


Figure 10-2 : Aperçu de la phase d'implémentation (éléments essentiels)

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

10.1 Création des éléments livrables

Dans ce processus, l'équipe Scrum crée les produits livrables du sprint en travaillant sur les tâches du Sprint Backlog. C'est le processus le plus chronophage pour l'équipe et le Scrum Master. L'équipe est soutenue par le Scrum Master, qui anime les réunions de l'équipe, gère les obstacles auxquels elle est confrontée et fait tout ce qu'il peut pour permettre aux membres de l'équipe de se concentrer sur la création des produits livrables du sprint. L'équipe Scrum utilise un Scrumboard pour suivre sa progression pendant le sprint. L'équipe Scrum utilise les informations sur sa progression pour obtenir une bonne indication de sa capacité à respecter ses engagements et, si nécessaire, pour prendre des mesures afin de garantir le meilleur résultat possible pour le sprint dans les circonstances données.

La figure 10-3 illustre les contributions, outils et résultats du processus de *création des éléments livrables*.

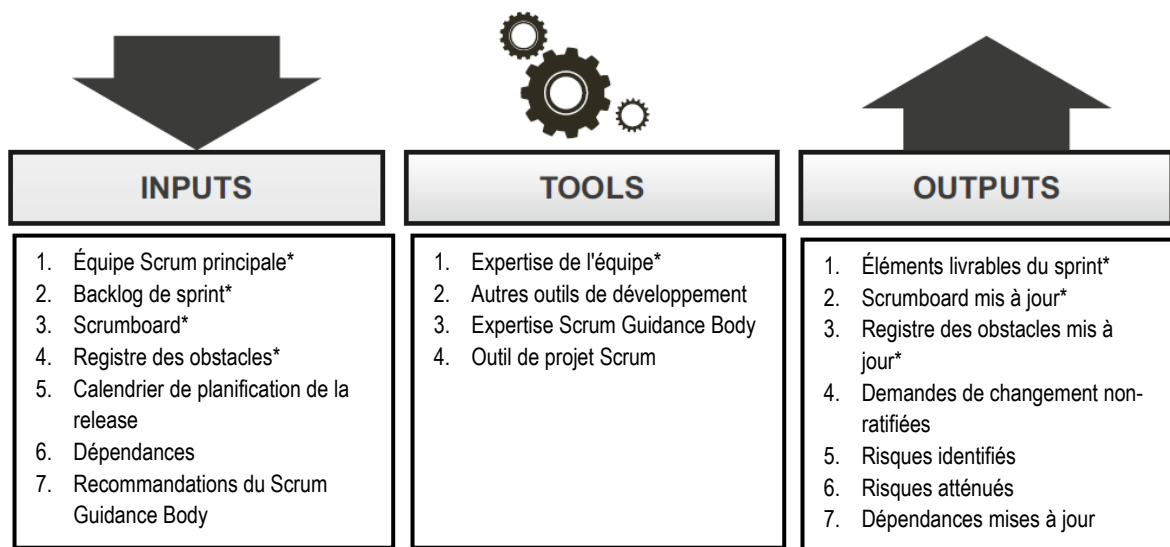


Figure 10-3 : Création des éléments livrables — Contributions, outils et résultats

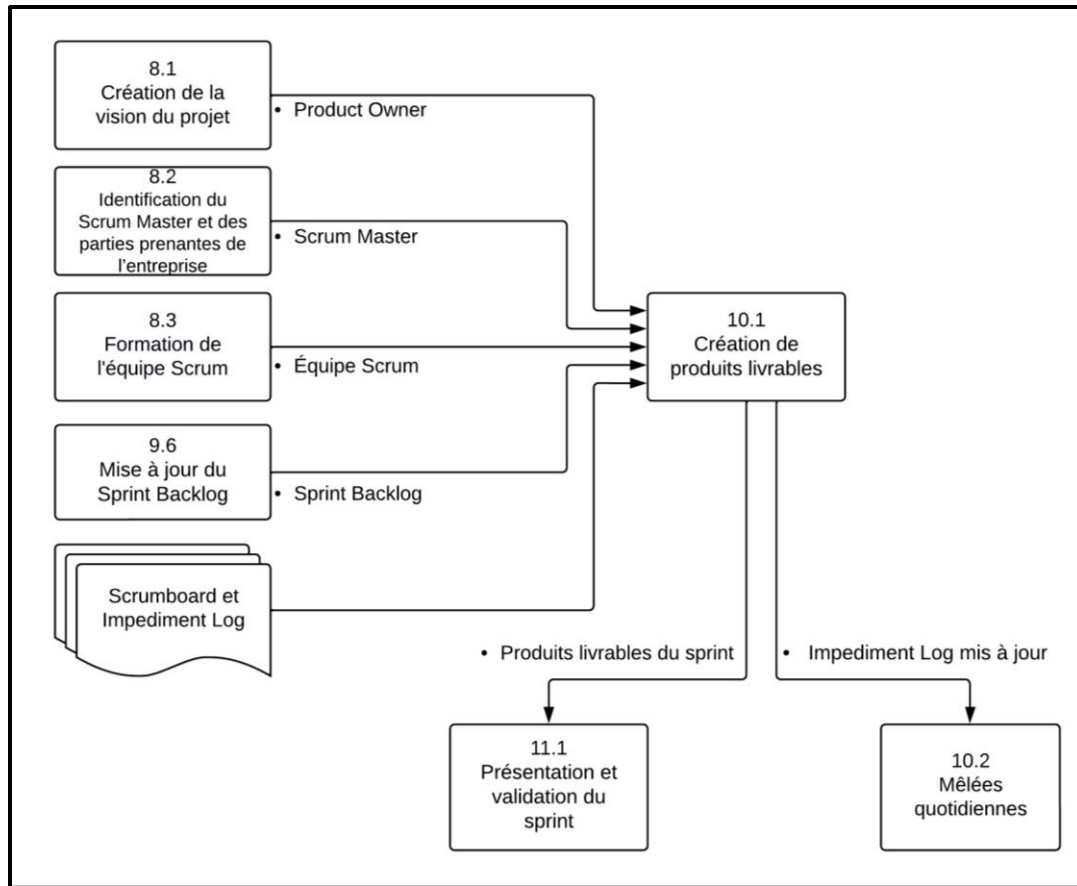


Figure 10-4 : Création des éléments livrables — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

10.1.1 Contributions

10.1.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

10.1.1.2 Backlog de sprint*

Décrit dans la section 9.6.3.1.

10.1.1.3 Scrumboard*

Description dans les sections 9.3.3.3 et 9.4.3.2.

La transparence de Scrum provient d'outils d'information librement consultables comme le Scrumboard, qui montre les progrès de l'équipe. L'équipe utilise un Scrumboard pour planifier et suivre les progrès au cours de chaque Sprint. L'équipe Scrum doit mettre à jour le Scrumboard selon les besoins, afin que celui-ci fournisse des informations visuelles précises et un contrôle sur le travail en cours, comme convenu et validé par l'équipe. L'équipe utilise le Scrumboard créé à partir du Sprint Backlog dans la phase de planification et d'estimation, qui contient initialement toutes les tâches dans la colonne « À faire » au début du sprint. Les membres de l'équipe Scrum passent en revue les User Stories et les tâches du Scrumboard et déplacent les tâches dans les colonnes « En cours » et « Terminé » au fur et à mesure de l'avancement du travail. Une colonne supplémentaire (comme « Tests ») peut être ajoutée au Scrumboard en fonction du flux de travail de l'équipe de Scrum lors de la création des produits livrables.

Au fur et à mesure que l'équipe ajoute/met à jour des tâches et assigne des tâches sur lesquelles travailler, le Scrumboard est mis à jour avec les tâches supplémentaires et l'état des tâches. Par exemple, la figure 10-5 montre que toutes les tâches de la User Story 1 sont terminées, mais que l'équipe travaille actuellement sur certaines tâches des User Stories 2 et 3. La User Story 4 a été décomposée en tâches, mais l'équipe Scrum n'a pas encore commencé à travailler sur les tâches de cette User Story.






| USER STORIES | Tâches | | |
|--------------|--|--|---|
| | En cours | Test | Terminé |
| 1 | | |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  | | |

Figure 10-5: Scrumboard

Pour s'assurer que les membres de l'équipe Scrum s'approprient leur travail, il est recommandé que le membre de l'équipe Scrum qui travaille sur une tâche particulière déplace la tâche de « À faire » à « En cours » et mette son nom sur la tâche, de sorte que la tâche soit auto-assignée à la personne responsable de sa réalisation. De plus, un seul membre de l'équipe Scrum doit être responsable de la réalisation de chaque tâche. Ainsi, les User Stories doivent être décomposées en tâches de sorte qu'une seule personne puisse être responsable d'une tâche jusqu'à son achèvement. Lorsque l'équipe Scrum commence à travailler sur une User Story, elle peut avoir acquis une meilleure compréhension des tâches nécessaires à la réalisation de la User Story. Cela peut nécessiter l'ajout, la mise à jour ou la suppression de tâches dans le Scrumboard, selon la décision de l'équipe Scrum.

Lorsque toutes les tâches d'une User Story sont terminées (par exemple, comme dans la User Story 1 dans la figure ci-dessus), l'User Story est considérée comme terminée par l'équipe Scrum. Les User Stories terminées sont ensuite disponibles pour que le Product Owner les examine et les approuve ou les rejette. La révision des User Stories par le Product Owner peut être faite par le Product Owner soit après que les User Stories sont terminées, soit pendant le processus de *présentation et validation du sprint*. Si une User Story est approuvée par le Product Owner, elle est considérée comme « Done » (Terminée) par l'équipe Scrum (et plus aucun travail ne doit être effectué par l'équipe Scrum sur cette User Story).

Si le Product Owner rejette une User Story, le Product Owner doit indiquer la raison pour laquelle l'histoire d'utilisateur a été rejetée (c'est-à-dire, quels éléments des critères d'acceptation et/ou des critères « Done » (Terminé) n'ont pas été satisfaits). En fonction du temps restant dans le Sprint, après qu'une User Story est rejetée, et une fois que les raisons du Product Owner pour le rejet de la User Story soient fournies, il y aura deux options disponibles pour l'équipe Scrum :

- Travailler sur la User Story rejetée dans le sprint actuel (sur la base des inputs fournis par le Product Owner) puis, lorsque toutes les tâches requises pour la User Story sont terminées, l'équipe Scrum pourra soumettre à nouveau la User Story au Product Owner pour approbation au cours du même sprint.
- Ne pas travailler sur la User Story rejetée dans le même Sprint. Dans ce cas, la User Story retourne dans le Product Backlog priorisé afin qu'elle puisse être affectée à un autre Sprint. La User Story peut être assignée à nouveau à la même équipe de Scrum, ou une autre équipe de Scrum peut devenir responsable de cette User Story dans un prochain Sprint.

Le Scrumboard peut être tenu manuellement sur papier ou sur un grand tableau blanc, mais il peut aussi être tenu électroniquement dans une feuille de calcul ou à l'aide d'un outil de projet Scrum. Un Scrumboard est valable pour la durée d'un sprint. L'équipe Scrum créera un nouveau Scrumboard lors du prochain sprint.

10.1.1.4 **Registre des obstacles***

Un obstacle est tout blocage ou barrière qui réduit la productivité de l'équipe Scrum. Les obstacles doivent être identifiés, résolus et éliminés pour que l'équipe continue à travailler efficacement. Les obstacles peuvent être internes à l'équipe, tels qu'un flux de travail inefficace ou un manque de communication, ils peuvent également être externes. Des exemples d'obstacles externes incluent notamment les problèmes relatifs aux licences logicielles ou à la documentation superflue des exigences. L'approche Scrum avec sa transparence inhérente facilite l'identification des obstacles de façon fluide et simple. L'incapacité à identifier ou gérer les obstacles peut s'avérer très coûteuse. Les obstacles doivent être formellement enregistrés par le Scrum Master dans le registre des obstacles. Selon les cas, ils peuvent être discutés lors des mêlées quotidiennes et des réunions de revue de sprint.

10

10.1.1.5 **Calendrier de planification de la release**

Décrit dans la section 8.6.3.1.

10.1.1.6 **Dépendances**

Décrites dans la section 9.4.3.3.

10.1.1.7 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

Lors du processus de *création des éléments livrables*, les recommandations du Scrum Guidance Body peuvent inclure les meilleures pratiques permettant de créer des éléments livrables de façon efficace, notamment les méthodes préférées relatives à l'organisation des revues, des tests, de la documentation, etc.

10.1.2 Outils

10.1.2.1 Expertise de l'équipe*

Cela fait référence à l'expertise de l'équipe Scrum en termes de sa capacité à comprendre les user stories et les tâches du backlog de sprint dans le but de créer les incréments de produit finaux. L'expertise de l'équipe est utilisée pour évaluer les contributions nécessaires à l'exécution du travail prévu pour le projet. Cette évaluation et cette expertise sont appliquées à tous les aspects techniques et de gestion du projet lors du processus de *création des éléments livrables*. Les membres de l'équipe Scrum ont l'autorité et la responsabilité de déterminer les meilleurs moyens de convertir les éléments du backlog de produit priorisé en produits finis, sans avoir besoin d'impliquer parties prenantes de l'entreprise extérieures à l'équipe. Lorsque cela est nécessaire, le Scrum Guidance Body représente une source d'expertise supplémentaire.

10.1.2.2 Autres outils de développement

En fonction des exigences spécifiques du projet et des spécifications industrielles, d'autres outils de développement peuvent être utilisés en conséquence.

1. Remaniement

Le remaniement est un outil spécifique pour les projets de logiciels. Le but de cette technique est d'améliorer la maintenabilité du code existant, de le simplifier, de le rendre plus court et plus flexible. Remanier implique d'améliorer la conception du code existant sans changer la façon dont celui-ci se comporte. Cela implique :

- d'éliminer les codes redondants et répétitifs
- de diviser les méthodes et fonctions en de plus petites routines
- de définir clairement les variables et les noms de méthodes
- de simplifier la conception du code
- de rendre le code plus facile à comprendre et à modifier

Le remaniement régulier optimise la conception du code petit à petit sur une période de temps. À terme, le remaniement résulte en un code plus propre et maintenable tout en préservant toutes les fonctionnalités.

2. Schéma de conception

Les schémas de conception sont une solution formelle de suivi pour la résolution des problèmes de conception dans un domaine d'expertise spécifique. Ces schémas assurent le suivi à la fois du procédé utilisé et de la résolution effective et peuvent ensuite être réutilisés afin d'améliorer le processus de décision et de production.

10.1.2.3 Expertise du Scrum Guidance Body

Décrite dans la section 8.4.2.7.

Lors des processus de *création des éléments livrables*, *d'estimation des user stories* et de *validation des user stories*, l'expertise du Scrum Guidance Body peut être relative à des règles, législations et directives de développement documentées ou à des standards et meilleures pratiques (p.ex. des conseils sur la manière de mener les revues ou les tests). Il peut également y avoir une équipe d'experts sur un sujet, capable d'offrir des conseils à l'équipe Scrum sur la création des éléments livrables. Cette équipe peut inclure des architectes principaux, des développeurs expérimentés, des experts en sécurité ou d'autres personnes expérimentées.

10.1.2.4 Outil de projet Scrum

Décrites dans la section 2.5.3.1.

10

10.1.3 Résultats

10.1.3.1 Éléments livrables du sprint*

À la fin de chaque sprint, un incrément de produit ou élément livrable est terminé. L'élément livrable doit posséder toutes les fonctionnalités définies dans les user stories incluses dans le sprint et doit avoir été testé avec succès.

10.1.3.2 Scrumboard mis à jour*

Le scrumboard est régulièrement mis à jour au fur et à mesure que l'équipe complète les tâches. À la fin du sprint, le scrumboard est réinitialisé ou effacé et un nouveau scrumboard est créé pour le sprint suivant.

10.1.3.3 Registre des obstacles mis à jour

Décrit dans la section 10.1.1.4.

10.1.3.4 Demandes de changement non-ratifiées

Décrites dans la section 8.4.1.6.

10.1.3.5 Risques identifiés

Décrits dans la section 8.4.3.4.

10.1.3.6 Risques atténués

Pendant que l'équipe Scrum effectue le travail consistant à créer les incréments de produit en fonction des user stories dans le backlog de produit, elle effectue les actions relatives à l'atténuation des risques qui ont été définies en réponse aux risques identifiés précédemment. Tout au long du processus de *création des éléments livrables*, l'équipe documente tous les risques nouvellement identifiés et les actions d'atténuations qui ont été prises. Le registre des risques du projet est un document vivant, continuellement mis à jour par l'équipe durant le projet afin de refléter le statut actuel de tous les risques.

Des informations supplémentaires relatives à la gestion des risques sont consultables dans la section 7.4.3.

10.1.3.7 Dépendances mises à jour

Décrites dans la section 9.4.3.3.

10.2 Organisation des mêlées quotidiennes

Au cours de ce processus, une mêlée quotidienne de mise au point très ciblée est organisée. Cette réunion limitée dans le temps permet à l'équipe Scrum de se tenir informée de ses progrès et des obstacles auxquels elle est confrontée.

La figure 10-6 illustre les contributions, outils et résultats des *mêlées quotidiennes*.

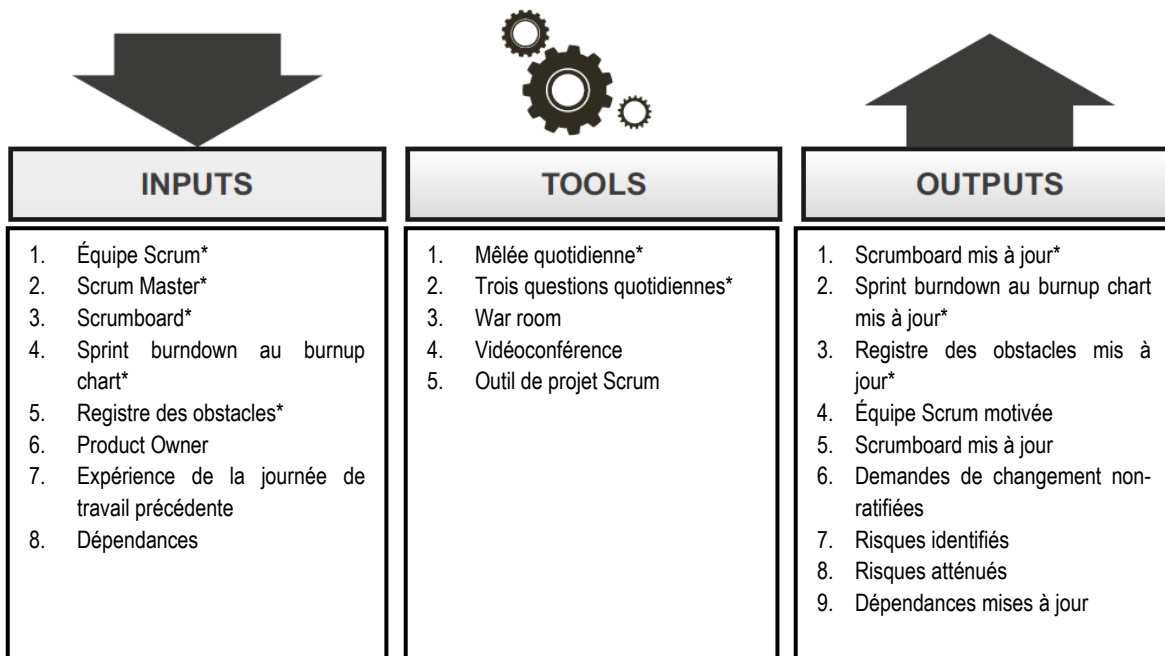


Figure 10-6 : Mêlées quotidiennes — Contributions, outils et résultats

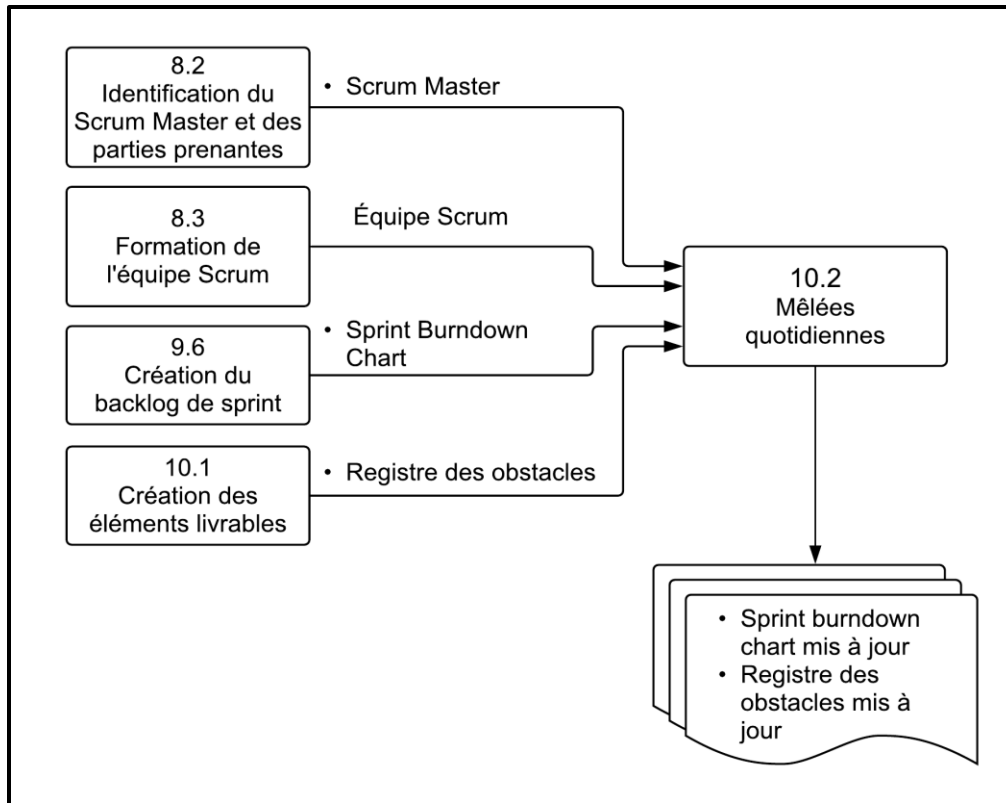


Figure 10-7 : Mêlées quotidiennes — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

10.2.1 Contributions

10.2.1.1 Équipe Scrum*

Décrite dans la section 8.3.3.1.

10.2.1.2 Scrum Master*

Décrit dans la section 8.2.3.1.

10.2.1.3 Scrumboard*

Décrite dans les sections 9.3.3.3, 9.4.3.2, and 10.1.1.3.

10.2.1.4 **Registre des obstacles***

Décrit dans la section 10.1.1.4.

10.2.1.5 **Sprint burndown au burnup chart ***

Décrit dans la section 9.5.3.2.

10.2.1.6 **Registre des obstacles***

Décrit dans la section 10.1.1.4.

10.2.1.7 **Product Owner**

Décrit dans la section 8.1.3.1.

10.2.1.8 **Expérience de la journée de travail précédente**

Les membres de l'équipe Scrum fournissent des mises à jour de statut aux autres membres de l'équipe lors de la mêlée quotidienne. Cette session est également appelée « standup » car les membres restent debout pendant la réunion. Les membres de l'équipe discutent les succès et l'expérience de la journée de travail précédente. Cette expérience contribue de façon importante à la mêlée quotidienne.

10.2.1.9 **Scrumboard**

Décrit dans la section 10.1.1.3.

10.2.1.10 **Dépendances**

Décrites dans la section 9.4.3.3.

10.2.2 Outils

10.2.2.1 Mêlée quotidienne*

La mêlée quotidienne est une réunion quotidienne courte, limitée à 15 minutes. Les membres de l'équipe se réunissent afin de présenter un rapport sur leurs progrès dans le sprint et leur programme pour les activités de la journée. La réunion est très courte et tous les membres de l'équipe Scrum doivent y participer. La réunion n'est pas pour autant annulée ou reportée si un ou plusieurs membres ne peuvent pas y assister. La mêlée quotidienne est gérée par l'équipe Scrum tandis que le Scrum Master anime la réunion, selon les besoins.

Durant la réunion, chaque membre de l'équipe Scrum doit répondre aux trois questions quotidiennes mentionnées dans la section 10.2.2.2. Les discussions entre le Scrum Master et l'équipe Scrum ou entre les membres de l'équipe Scrum sont encouragées, mais de telles discussions doivent avoir lieu après la réunion afin de garantir la durée limitée de la mêlée quotidienne.

10.2.2.2 Trois questions quotidiennes*

Lors des mêlées quotidiennes facilitées par le Scrum Master, chaque membre de l'équipe Scrum fournit des informations en répondant à trois questions spécifiques :

- Qu'ai-je fait depuis la dernière réunion ?
- Qu'ai-je prévu de faire d'ici à la prochaine réunion ?
- Le cas échéant, à quels blocages et obstacles dois-je faire face en ce moment ?

En se concentrant sur ces trois questions, l'équipe dans son ensemble peut clairement comprendre le statut du travail réalisé. À l'occasion, d'autres éléments peuvent être abordés mais cela doit être réduit à un minimum étant donné que la réunion est limitée dans le temps.

Il est fortement recommandé que les membres de l'équipe répondent aux deux premières questions de façon quantifiable, si cela est possible, plutôt que sous la forme de réponses longues et qualitatives. Les membres de l'équipe peuvent organiser des réunions supplémentaires après les mêlées quotidiennes afin d'aborder les éléments qui nécessitent davantage de discussion.

10.2.2.3 War room

Dans Scrum, il est préférable que l'équipe soit regroupée. Dans ce cas, tous les membres de l'équipe travaillent au même endroit. Le terme communément utilisé pour décrire cet endroit est « war room » (ou salle de décision). En général, le lieu est conçu de façon à ce que les membres de l'équipe puissent se déplacer librement et communiquer facilement en étant tous proches les uns des autres. Généralement des cartes d'index, des feuillets autocollants et d'autres outils low-tech tactiles sont disponibles dans cette salle afin de faciliter le flux de travail, la collaboration et la résolution des problèmes. La salle est parfois bruyante en raison des conversations de l'équipe, mais ces conversations contribuent aux progrès de l'équipe. Une bonne war room n'est pas compartimentée, elle permet à l'équipe entière d'être installée ensemble et garantit la communication en face-à-face, ce qui contribue à une meilleure cohésion au sein de l'équipe et à l'ouverture. La war room est également l'endroit idéal pour organiser les mêlées quotidiennes. Parties prenantes de l'entreprise qui font partie d'autres équipes Scrum peuvent également passer dans la salle afin de discuter des problèmes pertinents.

10.2.2.4 Vidéoconférence

Dans les situations réelles, il n'est pas toujours possible pour l'équipe Scrum entière d'être regroupée. Dans de tels cas, il devient impératif d'utiliser des outils de vidéoconférence afin de favoriser la communication en face-à-face.

10

10.2.3 Résultats

10.2.3.1 Scrumboard mis à jour*

Décrite dans les sections 9.3.3.3 and 9.4.3.2.

10.2.3.2 Registre des obstacles mis à jour*

Décrit dans la section 10.1.1.4.

10.2.3.3 Sprint burndown au burnup chart mis à jour*

Le Sprint Burndown Chart doit être mis à jour quotidiennement pour montrer les progrès réalisés par l'équipe Scrum et pour permettre de détecter les estimations incorrectes. Si le Sprint Burnup Chart montre que l'équipe Scrum n'est pas sur la bonne voie pour terminer les tâches du sprint à temps, le Scrum Master doit identifier les obstacles ou les empêchements et tenter de les éliminer. Pour plus d'informations sur le Sprint Burndown Chart, voir la section 9.6.3.3.

10.2.3.4 **Registre des obstacles mis à jour***

Décrit dans la section 10.1.1.4.

10.2.3.5 **Équipe Scrum motivée**

Les mêlées quotidiennes répandent l'idée que chaque membre de l'équipe est important et un contributeur majeur, améliorant ainsi le moral individuel et le moral de l'équipe. Associé au concept d'équipes autoorganisées, cela améliore la motivation globale, engendre une meilleure performance de l'équipe et une meilleure qualité en termes d'incréments produits.

L'équipe Scrum est décrite dans la section 8.3.3.1.

10.2.3.6 **Scrumboard mis à jour**

Décrit dans la section 10.1.1.3.

10.2.3.7 **Demandes de changement non-ratifiées**

Décrites dans la section 8.4.1.6.

10.2.3.8 **Risques identifiés**

Décrits dans la section 8.4.3.4.

10.2.3.9 **Risques atténués**

Décrits dans la section 10.1.3.6.

10.2.3.10 **Dépendances mises à jour**

Décrites dans la section 9.3.3.3.

10.3 Raffinage du Product Backlog priorisé

La figure 10-8 illustre les contributions, outils et résultats du processus d'*Raffinage du Product Backlog priorisé*.

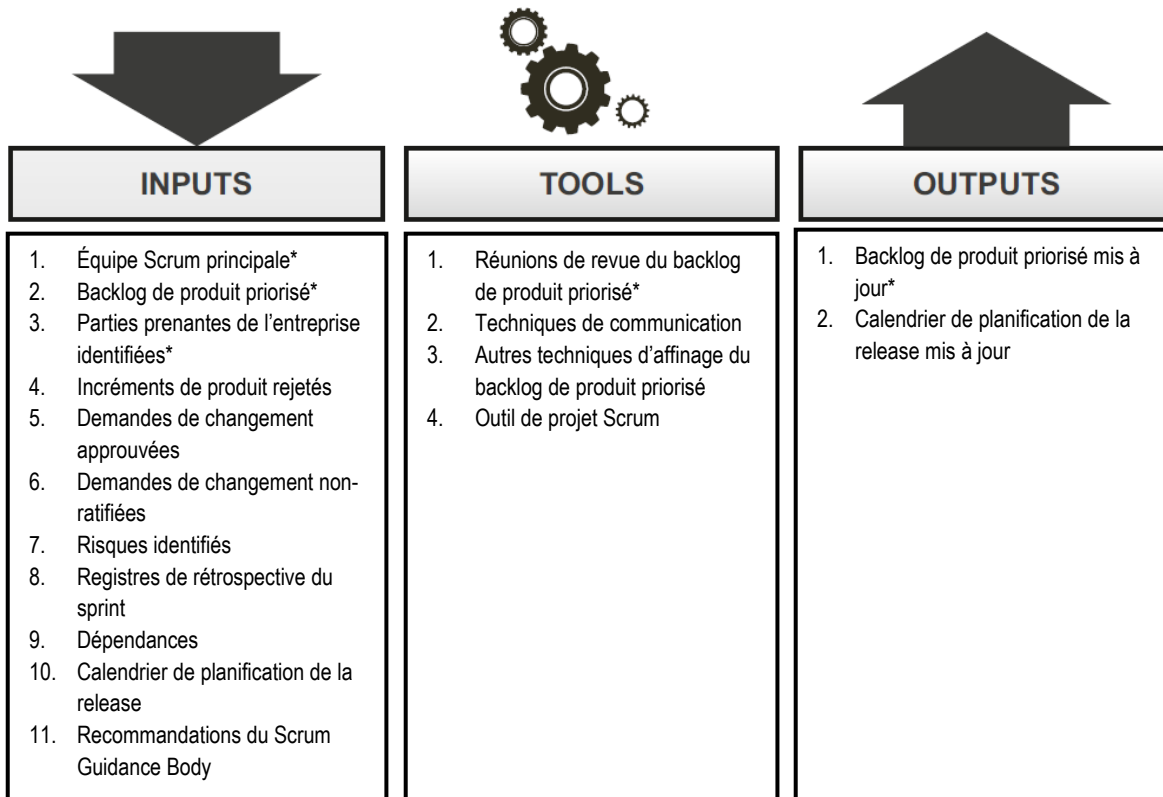


Figure 10-8 : Raffinage du Product Backlog priorisé — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus

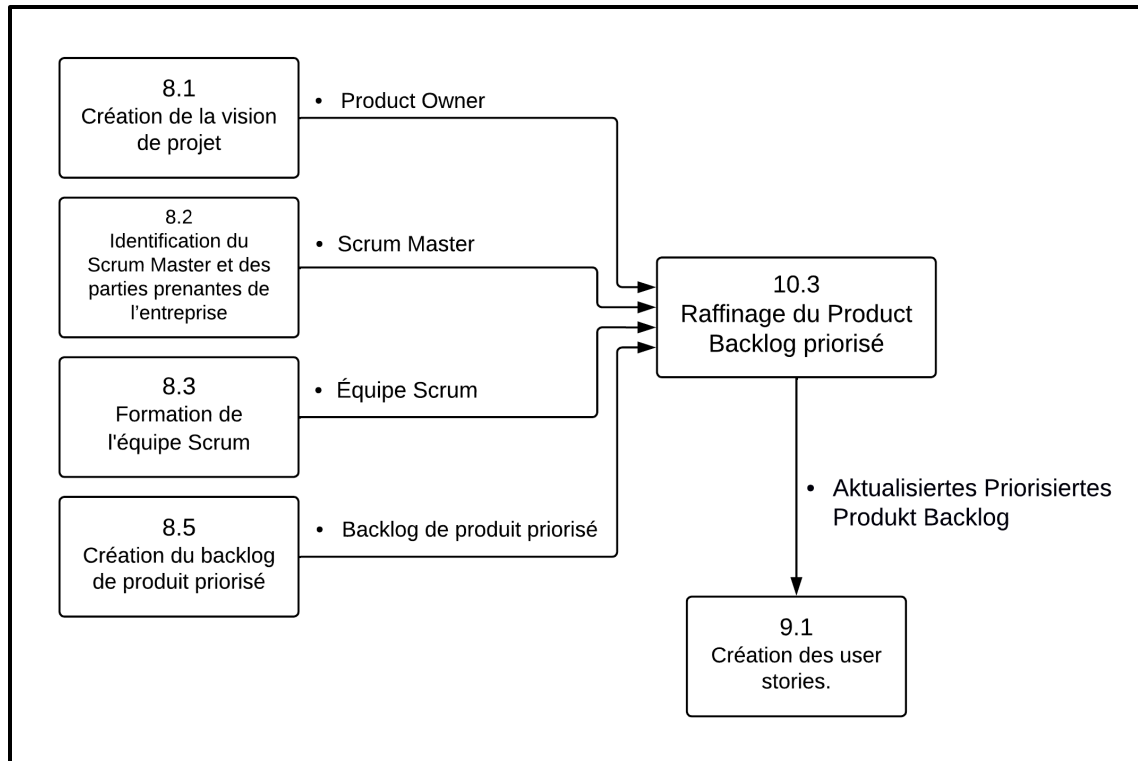


Figure 10-9: Raffinage du Product Backlog priorisé — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

10.3.1 Contributions

10.3.1.1 Équipe Scrum principale*

Décrite dans les sections 8.1.3.1, 8.2.3.1 et 8.3.3.1.

10.3.1.2 Backlog de produit priorisé*

Décrit dans la section 8.5.3.1.

10.3.1.3 Parties prenantes de l'entreprise identifiées*

Décrit dans la section 3.3.2.1.

Afin de maintenir à jour le Product Backlog priorisé des changements dans les exigences et/ou les priorités, le Product Owner travaille continuellement avec le client et les autres parties prenantes de l'entreprise pour saisir et comprendre tout changement dans leurs besoins.

10.3.1.4 Incréments de produit rejetés

Dans le cas où un élément livrable ne répond pas aux critères d'acceptation, il est considéré comme un incrément de produit rejeté. Les incréments rejetés ne sont normalement pas enregistrés sur une liste séparée. Ils demeurent simplement dans le backlog et ne sont pas marqués comme *terminés* de façon à ce qu'ils puissent être repriorisés lors de l'*Raffinage du Product Backlog priorisé* et pris en compte pour être développés lors du prochain sprint.

10.3.1.5 Demandes de changement approuvées

Décrites dans la section 8.4.1.5.

10.3.1.6 Demandes de changement non-ratifiées

Décrites dans la section 8.4.1.6.

10.3.1.7 Risques identifiés

Décrits dans la section 8.4.3.4.

10.3.1.8 Backlog de produit de programme mis à jour

De même que pour le backlog de produit du projet, le backlog de produit du programme peut également être affiné périodiquement afin d'incorporer les changements et nouvelles exigences. Les changements apportés au backlog de produit de programme peuvent être le résultat de changements de conditions externes ou internes. Les conditions externes peuvent inclure des changements en termes de scénarios de gestion, de tendances technologiques ou d'exigences de conformité légales. Les facteurs internes susceptibles d'affecter le backlog de produit de programme peuvent être associés à des changements en termes de stratégie ou de politique d'organisation, de risques identifiés ou d'autres facteurs. Les modifications des exigences du backlog de produit de programme ont souvent un impact sur les backlogs de produit des projets sous-jacents, elles doivent donc être prises en compte lors de l'*Raffinage du Product Backlog priorisé*.

10.3.1.9 Registres de rétrospective du sprint

Décrits dans la section 11.3.3.4.

10.3.1.10 Dépendances

Décrites dans la section 9.4.3.3.

10.3.1.11 Calendrier de planification de la release

Décrit dans la section 8.6.3.1.

10.3.1.12 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Lors de l'*Raffinage du Product Backlog priorisé*, les recommandations du Scrum Guidance Body peuvent inclure les meilleures pratiques relatives à la compréhension et au rassemblement systématique des exigences des parties prenantes de l'entreprise et des équipes Scrum ainsi qu'à la priorisation correcte du backlog de produit et à la communication des mises à jour à toutes les personnes concernées participant au projet Scrum.

10.3.2 Outils

10.3.2.1 Réunions de revue du backlog de produit priorisé*

Le Product Owner peut avoir plusieurs réunions distinctes avec parties prenantes de l'entreprise concernées, le Scrum Master et l'équipe Scrum afin de s'assurer qu'il dispose de suffisamment d'informations pour effectuer les mises à jour du backlog lors de l'*Raffinage du Product Backlog priorisé*.

L'objectif des réunions de revue du backlog de produit priorisé est de garantir que les user stories et les critères d'acceptation sont compris et correctement écrits par le Product Owner de façon à refléter les réelles exigences et priorités des parties prenantes de l'entreprise (clients), que les user stories sont comprises par tous les membres de l'équipe Scrum, et que les user stories à haut niveau de priorité sont suffisamment affinées pour permettre à l'équipe Scrum de les estimer et de s'engager à les réaliser.

Les réunions de revue du backlog de produit priorisé garantissent également que les user stories non-pertinentes sont supprimées et que toute demande de changement approuvée ou risque identifié est ajouté au backlog.

10.3.2.2 Techniques de communication

Scrum encourage les communications précises et efficaces principalement grâce au regroupement de l'équipe Scrum. Scrum favorise également les interactions informelles en face-à-face plutôt que les communications écrites. Lorsqu'une équipe Scrum doit être dispersée, le Scrum Master doit s'assurer que des méthodes de communication efficaces sont disponibles de façon à ce que les équipes puissent s'autoorganiser et travailler efficacement.

10.3.2.3 Autres techniques d'Raffinage du Product Backlog priorisé

D'autres outils d'Raffinage du Product Backlog priorisé incluent la majorité des outils utilisés lors des processus suivants :

- *Développement des epics* — décrit dans la section 8.4.2.
- *Création du backlog de produit priorisé* — décrite dans la section 8.5.2.
- *Planification de la release* — décrite dans la section 8.6.2.
- *Création des user stories* — décrite dans la section 9.1.2.
- *Estimation des user stories* — décrite dans la section 9.2.2.
- *Validation des user stories* — décrite dans la section 9.3.2.
- *Identification des tâches* — décrite dans la section 9.4.2.
- *Estimation des tâches* — décrite dans la section 9.5.2.

10.3.3 Résultats

10.3.3.1 Backlog de produit priorisé mis à jour

Décrit dans la section 8.5.3.1.

Le backlog de produit priorisé peut être mis à jour avec de nouvelles user stories, de nouvelles demandes de changement, de nouveaux risques identifiés, des user stories mises à jour ou une nouvelle hiérarchisation des user stories existantes.

10.3.3.2 Calendrier de planification de la release mis à jour

Décrit dans la section 8.6.3.1.

Le calendrier de planification de la release peut être mis à jour afin de refléter l'impact de nouvelles user stories ou de leur modification dans le backlog de produit priorisé.

10.4 Diagramme de flux de données de la phase d'implémentation

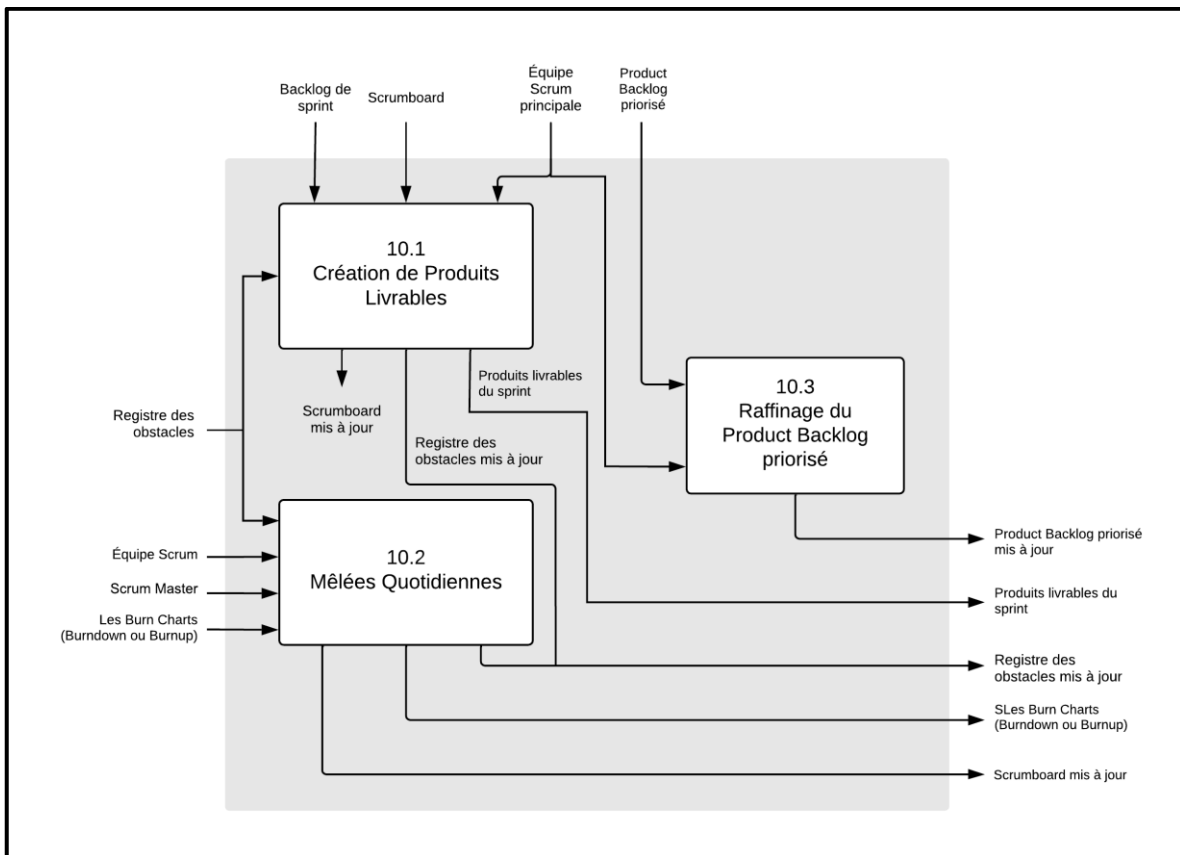


Figure 10-10 : Phase d'implémentation — Diagramme de flux de données

11. REVUE ET RÉTROSPECTIVE

La phase de revue et de rétrospective concerne la revue des incréments de produit et du travail qui a été effectué, elle implique également de déterminer les moyens d'améliorer les pratiques et méthodes utilisées afin d'effectuer le travail du projet. Dans les grosses organisations, les processus de *revue* et de *rétrospective* peuvent également impliquer de convoquer des réunions Scrum of Scrums.

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la *phase de revue et de rétrospective* s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Afin de faciliter la meilleure application possible de l'approche Scrum, ce chapitre identifie les contributions, outils et résultats de chaque processus comme étant soit « obligatoires » ou « facultatifs ». Les contributions, outils et résultats accompagnés d'un astérisque (*) sont obligatoires ou essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont facultatifs.

Il est recommandé que l'équipe Scrum et les personnes qui découvrent l'approche et les processus Scrum se concentrent principalement sur les contributions, outils et résultats obligatoires. En revanche, les Product Owners, Scrum Masters et autres praticiens Scrum expérimentés pourront aspirer à améliorer leurs connaissances grâce à l'ensemble des informations contenues dans ce chapitre. Il est également important de noter que bien que tous les processus ne soient définis qu'une seule fois dans le *Guide SBOK®*, ils ne sont pas nécessairement réalisés de façon séquentielle ou séparément. À certains moments, il peut s'avérer utile de combiner certains processus, en fonction des exigences spécifiques de chaque projet.

Ce chapitre est écrit depuis la perspective de l'équipe Scrum travaillant sur un sprint afin de produire des éléments potentiellement livrables, pouvant faire partie d'un plus gros projet, programme ou portefeuille. Des informations supplémentaires concernant le passage à l'échelle Scrum pour les gros projets sont disponibles dans le chapitre 13 et celles sur le passage à l'échelle pour l'entreprise dans le chapitre 14.

La phase de révision et rétrospective est la troisième et dernière des trois phases qui se répètent dans chaque sprint.

Le Product Owner et les parties prenantes concernées examinent les produits livrables que l'équipe a créés et transmettent leurs commentaires. Le Product Owner évalue chaque User Story et détermine si elle répond à ses critères d'acceptation respectifs et, par conséquent, l'accepte ou la rejette.

Dans la dernière partie d'un sprint, l'équipe Scrum détermine les moyens d'améliorer continuellement son travail.

Il est également important de réaliser que, bien que toutes les phases et tous les processus soient définis de manière unique dans le guide SBOK®, ils ne sont pas nécessairement exécutés séquentiellement ou séparément. Parfois, il peut être plus approprié de faire chevaucher certaines phases et/ou processus, en fonction des exigences spécifiques de chaque projet.

La figure 11-1 offre un aperçu des processus de la phase de revue et de rétrospective qui sont les suivants :

11.1 Démonstration et validation du sprint — l'équipe Scrum fait la démonstration des incréments de produit au Product Owner et aux parties prenantes de l'entreprise concernées lors d'une réunion de revue de sprint. L'objectif de cette réunion est d'obtenir l'approbation et l'acceptation User Stories du sprint par le Product Owner.

Ce processus n'est pas seulement un élément de qualité important dans un projet Scrum mais il est également un élément clé pour maintenir l'engagement des parties prenantes. Les parties prenantes de l'entreprise sont encouragées à participer à la réunion de révision de sprint afin d'acquérir une connaissance directe du produit ou du service et de son évolution, et de fournir un retour d'information. Le retour d'information des parties prenantes est un input important pour les futurs sprints.

11.2 Rétrospective de sprint — le Scrum Master et l'équipe Scrum se réunissent pour discuter des enseignements tirés lors du sprint. Ces informations sont documentées sous forme d'enseignements qui peuvent être appliqués lors des futurs sprints. Cette discussion résulte souvent en des améliorations actionnables convenues ou des mises à jour des recommandations du Scrum Guidance Body. Ce processus est un élément essentiel de l'amélioration continue dans Scrum.

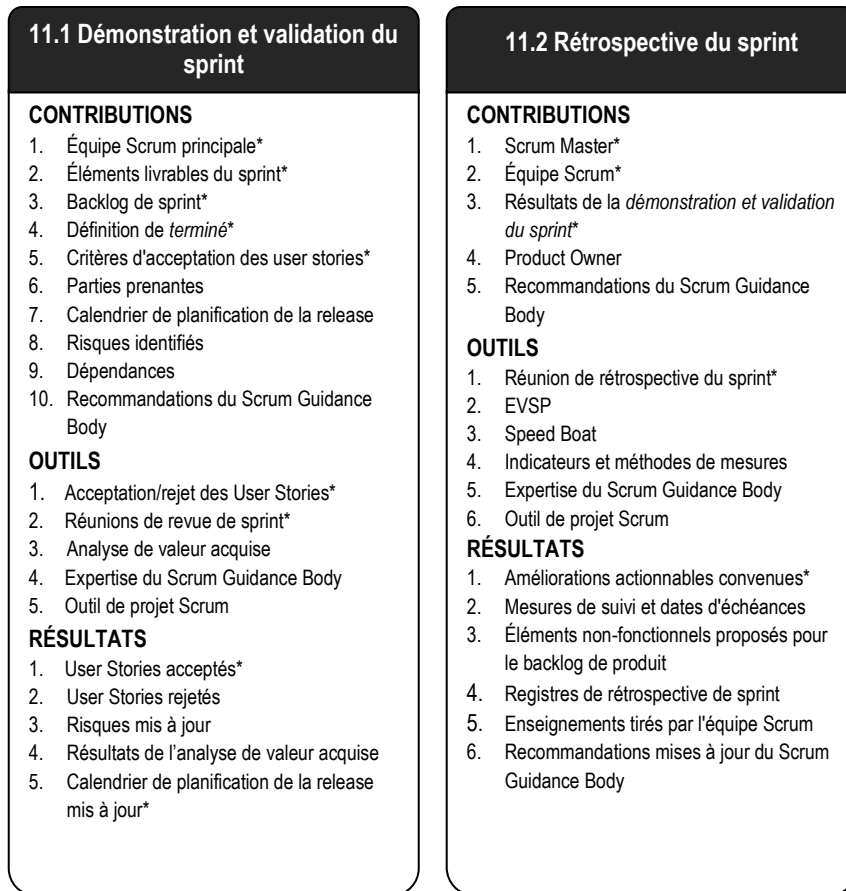


Figure 11-1 : Aperçu de la phase de revue et de rétrospective

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

La figure 11-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de la phase de revue et de rétrospective.

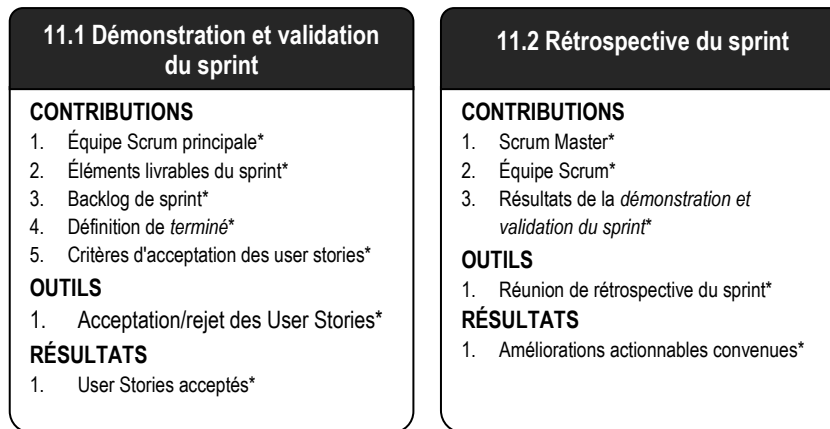


Figure 11-2 : Aperçu de la phase de revue et rétrospective (éléments essentiels)

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

11.1 Démonstration et validation du sprint

L'équipe Scrum présente les produits livrables au Product Owner et aux parties prenantes concernées lors d'une réunion de révision de sprint. Le but de cette réunion est d'assurer l'approbation des User Stories du sprint par le Product Owner.

Ce processus n'est pas seulement un élément de qualité important dans un projet Scrum, mais il est également un élément clé pour maintenir l'engagement des parties prenantes. Les parties prenantes de l'entreprise sont encouragées à participer à la réunion de révision de sprint afin d'acquérir une connaissance directe du produit ou du service et de son évolution, et de fournir un retour d'information. Le retour d'information des parties prenantes est un input important pour les futurs sprints.

La figure 11-3 illustre les contributions, outils et résultats du processus de *démonstration et validation du sprint*.

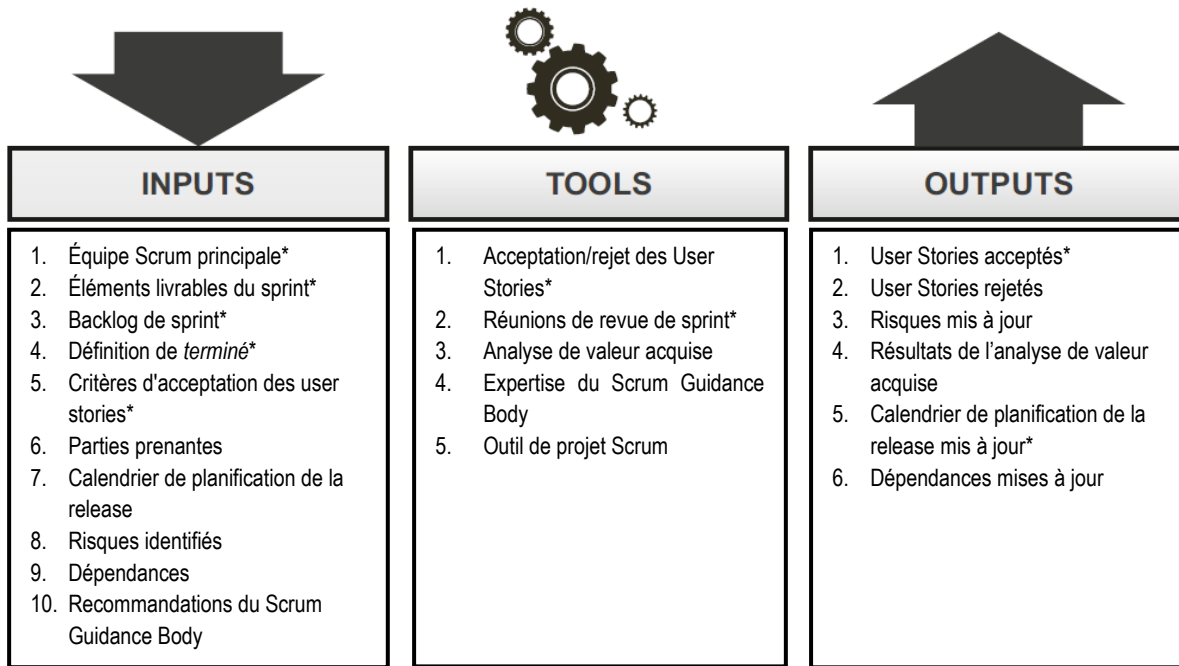


Figure 11-3 : Démonstration et validation du sprint — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

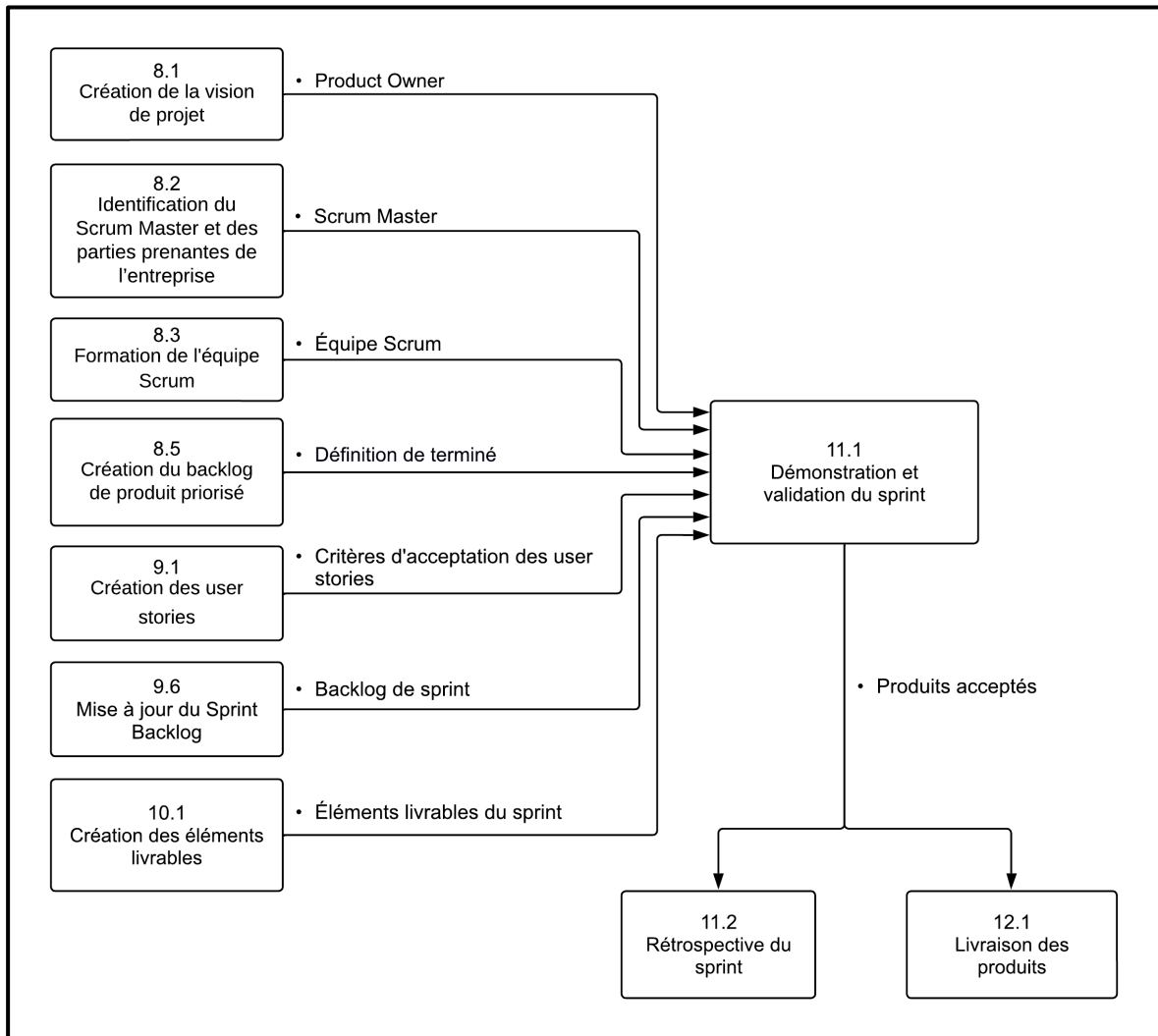


Figure 11-4 : Démonstration et validation du sprint — Diagramme de flux de données

11.1.1 Contributions

11.1.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

11.1.1.2 Éléments livrables du sprint*

Décrits dans la section 10.1.3.1.

11.1.1.3 **Backlog de sprint***

Décrit dans la section 9.5.3.1.

11.1.1.4 **Définition de *terminé****

Décrite dans la section 8.5.3.2.

11.1.1.5 **Critères d'acceptation des user stories***

Décrits dans la section 9.4.1.3.

11.1.1.6 **Parties prenantes de l'entreprise**

Décrites dans la section 8.2.3.2.

11.1.1.7 **Calendrier de planification de la release**

Décrit dans la section 8.6.3.1.

11.1.1.8 **Risques identifiés**

Décrits dans la section 8.4.3.4.

11.1.1.9 **Dépendances**

Décrites dans la section 9.4.3.3.

11.1.1.10 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

Durant le processus de *démonstration et validation du sprint* les recommandations du Scrum Guidance Body peuvent inclure les meilleures pratiques sur la façon de conduire les réunions de revue de sprint et d'évaluer les résultats de l'analyse de valeur acquise. Il peut également offrir des conseils sur la façon de partager les expériences avec d'autres personnes au sein de l'équipe Scrum principale ainsi qu'avec d'autres équipes Scrum impliquées dans le projet.

11.1.2 Outils

11.1.2.1 Acceptation/rejet des User Stories*

Après que les User Stories sont terminées par l'équipe Scrum, elles sont mises à la disposition du Product Owner pour examen. Le Product Owner peut réviser les User Stories dès qu'elles sont terminées, ou toutes à la fois lors d'une réunion de révision de sprint convoquée à la fin du sprint. Le Product Owner accepte les User Stories qui satisfont les critères d'acceptation des User Stories et les critères « Done » (Terminé) et rejette celles qui ne satisfont pas aux critères, avec un feedback sur la raison pour laquelle une User Story a été rejetée. S'il reste du temps dans le sprint en cours, les User Stories rejetées sont mises à la disposition de l'équipe afin de traiter les raisons du rejet et sont à nouveau fournies au Product Owner dans le même sprint pour un examen supplémentaire. À la fin d'un sprint, toutes les User Stories rejetées restent dans le Product Backlog priorisé afin de pouvoir être terminées lors d'un prochain sprint.

11.1.2.2 Réunion de revue de sprint*

Les réunions de révision de sprint sont convoquées à la fin de chaque sprint. Les membres de l'équipe centrale Scrum et les parties prenantes concernées participent aux réunions de révision de sprint pour présenter les produits livrables. L'équipe Scrum présente les User Stories et les produits livrables créés dans le cadre du sprint, y compris les nouvelles fonctionnalités ou les nouveaux produits créés. Les produits livrables de la User Story qui ont déjà été approuvés par le Product Owner avant la réunion de révision de sprint sont également présentés par l'équipe pendant cette réunion pour s'assurer que la/les partie(s) prenante(s) de l'entreprise examinent également la fonctionnalité respective. Le Product Owner et la/les partie(s) prenante(s) de l'entreprise inspectent tous les produits livrables et déterminent si des changements doivent être apportés dans un sprint ultérieur. À la fin de la réunion de révision de sprint, toutes les User Stories du sprint sont examinées ; certaines sont approuvées et d'autres sont rejetées par le Product Owner sur la base de leur conformité ou non aux critères d'acceptation et « Done » (Terminé).

11.1.2.3 Analyse de valeur acquise

Décrite dans la section 4.6.1.

11.1.2.4 Expertise du Scrum Guidance Body

Décrite dans la section 8.4.2.7.

Durant le processus de démonstration et validation du sprint, l'expertise du Scrum Guidance Body peut être relative aux meilleures pratiques qui documentent la façon de mener les réunions de revue de sprint. Il peut également y avoir des experts capables d'offrir des conseils sur la meilleure façon de faciliter une réunion de revue de sprint.

11.1.2.5 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

11.1.3 Résultats

11

11.1.3.1 User Stories acceptées*

Cela n'est pas souhaitable car l'objectif de chaque sprint est de produire des incréments qui remplissent les critères d'acceptation. Les User Stories qui répondent aux critères d'acceptation sont formellement acceptées par le Product Owner. Ces produits livrables acceptés de la User Story peuvent être remis au client, si désiré. Une liste des User Stories acceptées est conservée et mise à jour après chaque réunion de révision de sprint.

11.1.3.2 User Stories rejetées

Si les User Stories ne répondent pas aux critères d'acceptation, elles sont considérées comme incomplètes et sont rejetées par le Product Owner. Les user stories associées aux incréments livrables rejetés sont ajoutées au backlog de produit priorisé de façon à ce que de tels incréments puissent être pris en compte au cours d'un prochain sprint. Le travail sur les produits livrables associés aux User Stories rejetées peut être effectué par toute équipe Scrum à laquelle ces User Stories sont assignées à l'avenir.

Puisqu'une partie du travail a déjà été effectuée dans la création des produits livrables de ces User Stories rejetées, si les produits livrables partiellement achevés sont assignés pour être achevés dans un prochain sprint, l'estimation future de ces User Stories peut être inférieure à ce qui a été estimé pour les User Stories originales. Cependant, dans certains cas, les équipes Scrum peuvent choisir d'ignorer complètement les produits livrables associés aux User Stories rejetées, et considérer le travail à venir comme de nouvelles User Stories.

11.1.3.3 **Risques mis à jour**

Décrits dans la section 8.4.3.4.

11.1.3.4 **Résultats de l'analyse de valeur acquise**

Décrits dans la section 4.6.1.

11.1.3.5 **Calendrier de planification de la release mis à jour***

Décrit dans la section 10.3.3.2.

11.1.3.6 **Dépendances mises à jour**

Décrites dans la section 9.4.3.3.

11.2 Rétrospective de sprint

La figure 11-5 illustre les contributions, outils et résultats du processus de *rétrospective de sprint*.

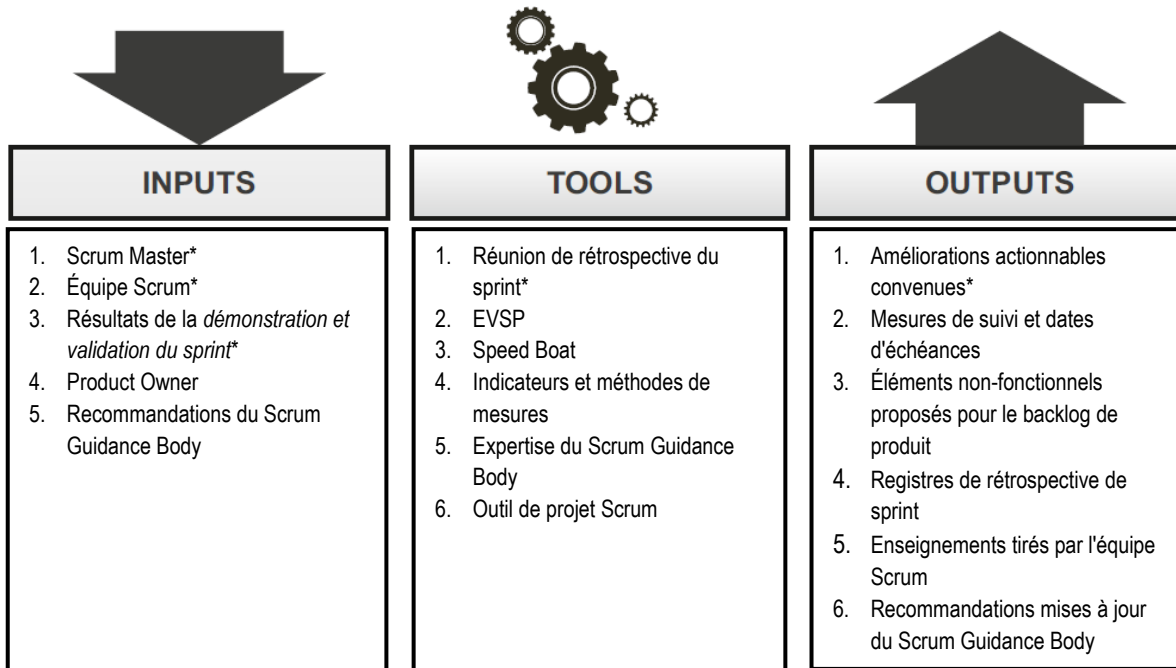


Figure 11-5 : Rétrospective de sprint — Contributions, outils et résultats

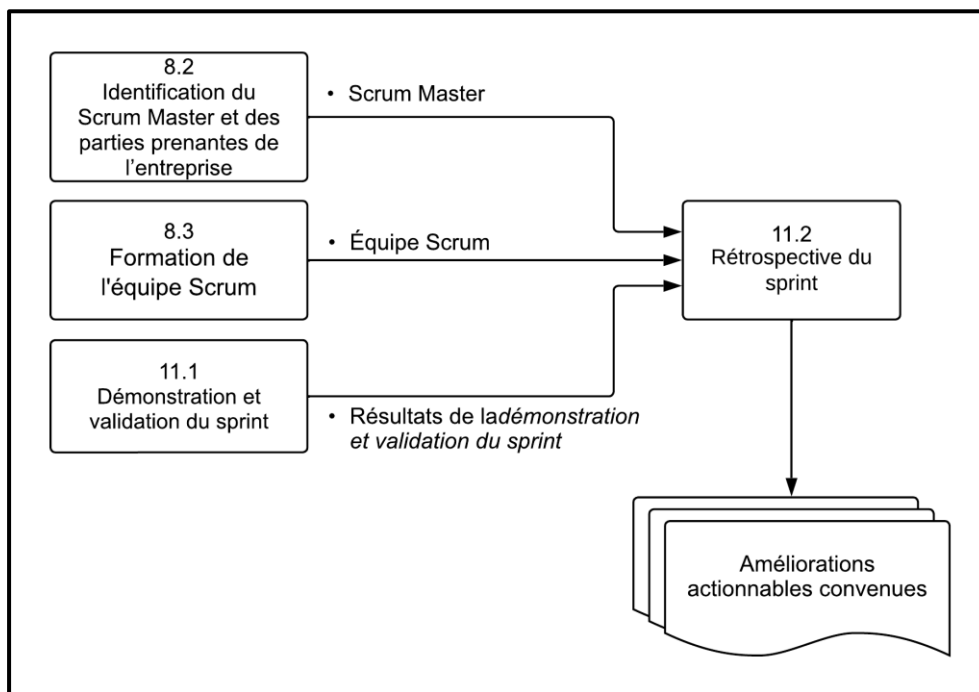


Figure 11-6 : Rétrospective de sprint — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

11.2.1 Contributions

11.2.1.1 Scrum Master*

Décrit dans la section 8.2.3.1.

11.2.1.2 Équipe Scrum*

Décrite dans la section 8.3.3.1.

11.2.1.3 Résultats de la démonstration et validation du sprint*

Décrits dans la section 11.1.3.

Les résultats du processus de *démonstration et validation du sprint* offrent des informations précieuses durant le processus de *rétrospective de sprint*.

11.2.1.4 Product Owner

Décrit dans la section 8.1.3.1.

11.2.1.5 Recommandations du Scrum Guidance Body

Le Scrum Guidance Body peut offrir des directives sur la manière de gérer les réunions de rétrospective du Sprint, y compris des suggestions concernant les outils à utiliser et la documentation ou les incréments de produits attendus lors de la réunion.

11.2.2 Outils

11.2.2.1 Réunion de rétrospective du sprint*

La réunion de rétrospective du sprint est un élément important dans le cadre de l'inspection et de l'adaptation de l'approche Scrum. C'est la dernière étape d'un sprint. Tous les membres de l'équipe Scrum participent à la réunion qui est facilitée ou modérée par le Scrum Master.

Bien que ce ne soit pas obligatoire, il est recommandé que le Product Owner participe à la réunion. Un des membres de l'équipe agit comme secrétaire et documente les discussions et éléments nécessitant des actions futures. Il est essentiel que cette réunion soit tenue dans un environnement ouvert et décontracté afin d'encourager la pleine participation de tous les membres de l'équipe. Les discussions de la réunion de rétrospective du sprint englobent ce qui c'est bien passé et ce qui s'est mal passé. Les principaux objectifs de la réunion sont l'identification de trois éléments spécifiques :

- 1) ce que l'équipe doit continuer à faire : meilleures pratiques
- 2) ce que l'équipe doit commencer à faire : amélioration des processus
- 3) ce que l'équipe doit arrêter de faire : problèmes de processus et blocages

Ces éléments sont abordés et une liste d'améliorations actionnables convenues est créée.

11.2.2.2 Explorateur — Acheteur — Vacancier — Prisonnier (ESVP)

C'est un exercice qui peut être mené au début d'une réunion de rétrospective de sprint pour comprendre l'état d'esprit des participants et donner le ton à la réunion. Il est demandé aux personnes présentes d'indiquer de façon anonyme laquelle de ces propositions représente au mieux leur attitude vis-à-vis de leur participation à la réunion.

- Explorateur — veut participer et apprendre tout ce qui est abordé durant la rétrospective
- Acheteur — veut écouter tout ce qui est dit et choisir ce qu'il retiendra de la rétrospective
- Vacancier — veut se détendre et être un touriste dans la rétrospective
- Prisonnier — veut être ailleurs et participe à la réunion car cela est obligatoire

Le Scrum Master rassemble ensuite les réponses, prépare et partage les informations avec le groupe.

11.2.2.3 Speed Boat

La technique du speed boat peut être utilisée pour mener la réunion de rétrospective du sprint. Les membres de l'équipe jouent le rôle de l'équipage d'un speed boat. Le bateau doit atteindre une île qui est symbolique de la vision du projet. Des feuillets autocollants sont utilisés par les personnes présentes afin de noter les moteurs et les obstacles. Les moteurs les aident à atteindre l'île tandis que les obstacles les empêchent. Cet exercice est limité dans le temps et ne doit prendre que quelques minutes. Une fois tous les éléments documentés, les informations sont rassemblées, discutées et priorisées lors d'un vote. Les éléments moteurs sont identifiés et des actions d'atténuation sont mises en place pour les obstacles en fonction de leur niveau de priorité.

11.2.2.4 Indicateurs et méthodes de mesures

Différents indicateurs peuvent être utilisés afin de mesurer et de contraster les performances de l'équipe lors du sprint en cours et leurs performances lors des précédents sprints. Quelques exemples de ces indicateurs incluent :

- la vélocité de l'équipe — le nombre de story points réalisés lors d'un sprint donné
- le taux de réussite de *terminé* — le pourcentage de story points *terminés* contre ceux sur lesquels l'équipe s'est engagée.
- l'efficacité d'estimation — le nombre ou pourcentage de déviations entre le temps estimé et réel passé sur des tâches et user stories.
- la revue des notes de feedback — du feedback peut être sollicité de la part des parties prenantes de l'entreprise en utilisant des systèmes de notation quantitatifs ou qualitatifs permettant de mesurer la performance de l'équipe.
- les notes du moral de l'équipe — les résultats de l'auto-évaluation du moral des membres de l'équipe.
- le feedback des pairs — les outils de feedback à 360 degrés peuvent être utilisés afin de solliciter des critiques constructives et des informations sur la performance de l'équipe.
- les progrès vers la release ou le lancement — la valeur commerciale offerte par chaque release ainsi que la valeur représentée par les progrès en cours vers la release. Cela contribue à la motivation de l'équipe ainsi qu'au niveau de la satisfaction au travail.

11.2.2.5 Expertise du Scrum Guidance Body

Décrite dans la section 8.4.2.7.

Durant le processus de *rétrospective de sprint*, l'expertise du Scrum Guidance Body peut être relative aux meilleures pratiques sur la façon de mener les réunions de *rétrospective de sprint*. Elle peut également inclure l'avis d'experts concernant l'utilisation des outils durant le processus de *rétrospective de sprint* afin d'atteindre les améliorations actionnables convenues lors des prochains sprints.

11.2.2.6 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

11.2.3 Résultats

11.2.3.1 Améliorations actionnables convenues*

Les améliorations actionnables convenues sont le principal résultat du processus de *rétrospective de sprint*. Elles constituent la liste des mesures à prendre imaginées par l'équipe afin de résoudre les problèmes et d'améliorer les processus de façon à optimiser leurs performances pour les sprints à venir.

11.2.3.2 Mesures de suivi et dates d'échéances

Une fois que les améliorations actionnables convenues ont été élaborées et acceptées, l'équipe Scrum peut considérer les mesures à prendre pour mettre en œuvre ces améliorations. Une date d'échéance pour la réalisation de chaque mesure sera définie.

11.2.3.3 Éléments non-fonctionnels proposés pour le backlog de produit priorisé

Lorsque le backlog de produit priorisé initial est développé, il est basé sur les user stories et les fonctionnalités requises. Souvent, les éléments non-fonctionnels peuvent ne pas être complètement définis au début du projet et peuvent apparaître pendant la réunion de revue ou de rétrospective du sprint. Ces éléments doivent être ajoutés au backlog de produit priorisé au fur et à mesure qu'ils sont découverts. Certains exemples d'exigences non-fonctionnelles sont les temps de réponse, les limites de capacité et les problèmes relatifs à la sécurité.

11.2.3.4 Registres de rétrospective de sprint

Le registre de rétrospective de sprint contient une trace de toutes les opinions, discussions et mesures à prendre abordées lors d'une réunion de rétrospective de sprint. Le Scrum Master peut faciliter la création de ce registre avec l'aide des membres de l'équipe Scrum principale. Le recueil de tous les registres de rétrospective de sprint devient l'agenda du projet et détaille les succès, difficultés, problèmes et résolutions du projet. Les registres sont des documents publics disponibles à toute personne au sein de l'organisation.

11.2.3.5 Enseignements tirés par l'équipe Scrum

Il est attendu de l'équipe Scrum autoorganisée et responsable qu'elle apprenne des erreurs commises durant un sprint. Ces enseignements aident les équipes à améliorer leur performance lors des futurs sprints. Ces enseignements peuvent également être documentés dans les recommandations du Scrum Guidance Body afin d'être partagés avec d'autres équipes Scrum.

Il peut y avoir plusieurs enseignements positifs tirés d'un sprint. Ces enseignements positifs sont un élément clé de la rétrospective et doivent être partagés de façon appropriée au sein de l'équipe et avec le Scrum Guidance Body, tandis que l'équipe œuvre à s'améliorer.

11.2.3.6 Recommandation mises à jour du Scrum Guidance Body

Après la réunion de rétrospective du sprint, des suggestions peuvent être formulées afin de réviser et d'améliorer les recommandations du Scrum Guidance Body. Si le Guidance Body accepte ces suggestions, celles-ci seront ajoutées en tant que mises à jour dans la documentation du Scrum Guidance Body.

11.3 Diagramme de flux de données de la phase de revue et de rétrospective

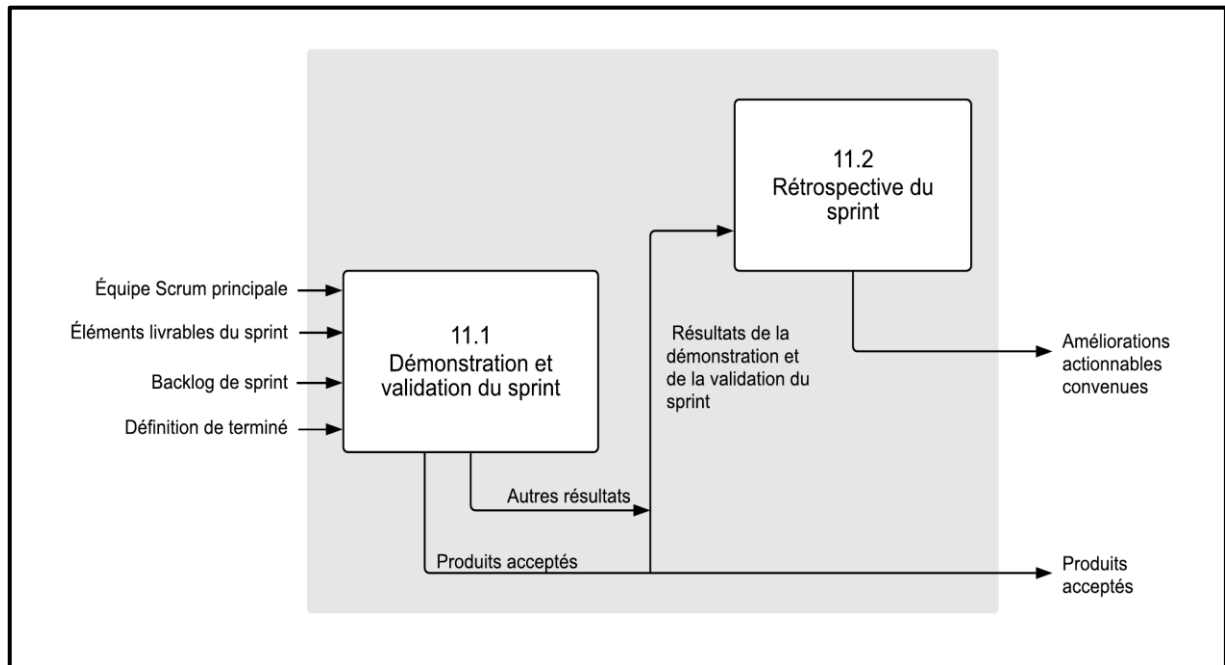


Figure 11-7 : Phase de revue et rétrospective — Diagramme de flux de données

12. RELEASE

La phase de release met l'accent sur la livraison des incréments de produits acceptés au client et sur l'identification, la documentation et l'internalisation des enseignements tirés au cours du projet.

Telle qu'elle est décrite dans *A Guide to the Scrum Body of Knowledge (Guide SBOK®)*, la phase de release s'applique aux éléments suivants :

- les portefeuilles, les programmes et/ou les projets *quel que soit* le secteur d'activité
- les produits, les services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise
- les projets de toute taille et de tout niveau de complexité

Le terme *produit* dans le *Guide SBOK®* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum s'applique efficacement à tout type de projet, quel que soit le secteur d'activité ; des petits projets impliquant des équipes de seulement six personnes jusqu'aux projets complexes avec des équipes constituées de plusieurs centaines de membres.

Afin de faciliter la meilleure application possible de l'approche Scrum, ce chapitre identifie les contributions, outils et résultats de chaque processus comme étant soit « obligatoires » ou « facultatifs ». Les contributions, outils et résultats accompagnés d'un astérisque (*) sont obligatoires ou essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont facultatifs.

Il est recommandé que l'équipe Scrum et les personnes qui découvrent l'approche et les processus Scrum se concentrent principalement sur les contributions, outils et résultats obligatoires. En revanche, les Product Owners, Scrum Masters et autres praticiens Scrum expérimentés pourront aspirer à améliorer leurs connaissances grâce à l'ensemble des informations contenues dans ce chapitre. Il est également important de noter que bien que tous les processus ne soient définis qu'une seule fois dans le *Guide SBOK®*, ils ne sont pas nécessairement réalisés de façon séquentielle ou séparément. À certains moments, il peut s'avérer utile de combiner certains processus, en fonction des exigences spécifiques de chaque projet.

Ce chapitre est écrit depuis la perspective de l'équipe Scrum travaillant sur un sprint afin de produire des éléments potentiellement livrables, pouvant faire partie d'un plus gros projet, programme ou portefeuille. Des informations supplémentaires concernant le passage à l'échelle Scrum pour les gros projets sont disponibles dans le chapitre 13 et celles sur le passage à l'échelle pour l'entreprise dans le chapitre 14.

La phase de release est généralement réalisée plusieurs fois au cours d'un projet Scrum. Bien que le résultat de chaque sprint soit un produit potentiellement expédiable, il n'y a pas nécessairement une release après chaque sprint.

Les produits livrables des User Stories acceptés d'un ou plusieurs sprints précédents sont transmis aux parties prenantes pertinentes de l'entreprise pour acceptation et utilisation, comme défini dans le calendrier de release.

En plus de la release des produits livrables, l'équipe centrale Scrum et les parties prenantes de l'organisation déterminent les moyens d'améliorer l'exécution des futures releases du projet.

Il est également important de réaliser que, bien que toutes les phases et tous les processus soient définis de manière unique dans le guide SBOK®, ils ne sont pas nécessairement exécutés séquentiellement ou séparément. Parfois, il peut être plus approprié de faire chevaucher certaines phases et/ou processus, en fonction des exigences spécifiques de chaque projet.

La figure 12-1 offre un aperçu des processus de la phase de release qui sont les suivants :

12.1 Livraison des produits — Au cours de ce processus, tous les produits livrables des User Stories acceptées des sprints précédents sont livrés ou transférés aux parties prenantes de l'entreprise concernées. Un accord officiel sur les produits finis documente la réussite du release.

12.2 Rétrospective de release — ce processus conclue le projet. Parties prenantes de l'entreprise de l'organisation et l'équipe Scrum principale se rassemblent afin d'effectuer une Rétrospective de release et d'identifier, de documenter et d'assimiler les enseignements du projet. Généralement, ces leçons mènent à la documentation d'améliorations actionnables convenues à mettre en place au cours des projets à venir.

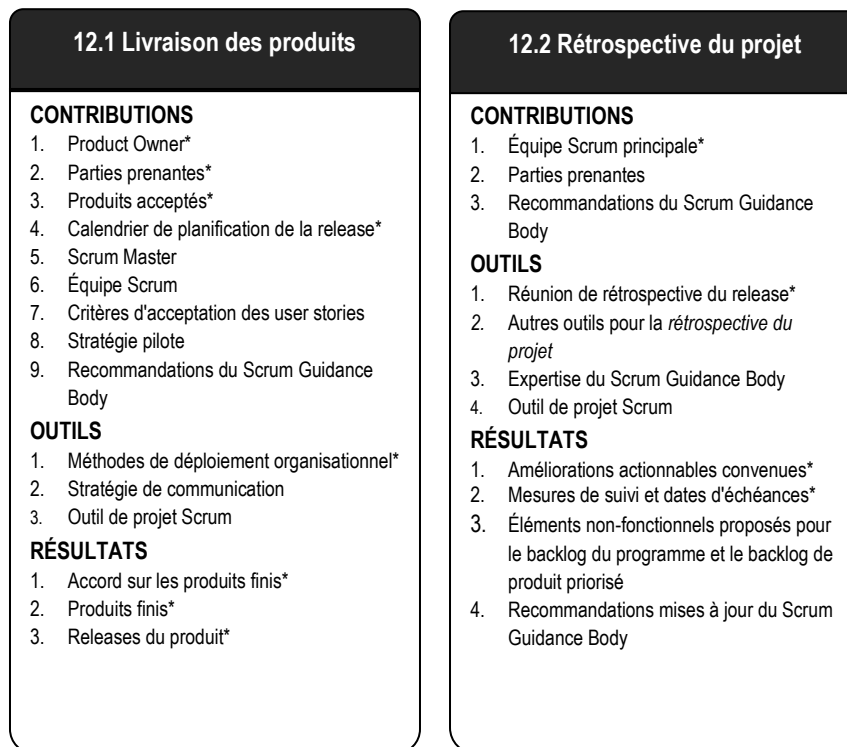


Figure 12-1 : Aperçu de la phase de release

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

La figure 12-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de la phase de release.

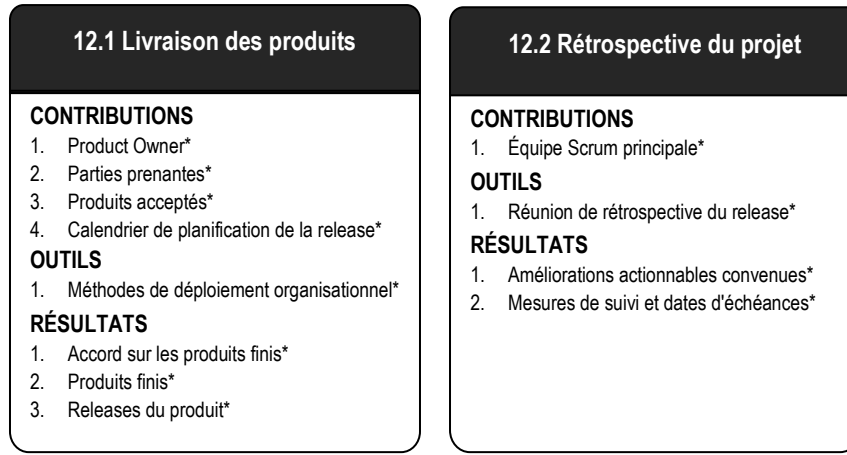


Figure 12-2 : Aperçu de la phase de release (éléments essentiels)

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

12.1 Livraison des produits

La figure 12-3 illustre les contributions, outils et résultats du processus de *livraison des produits*.

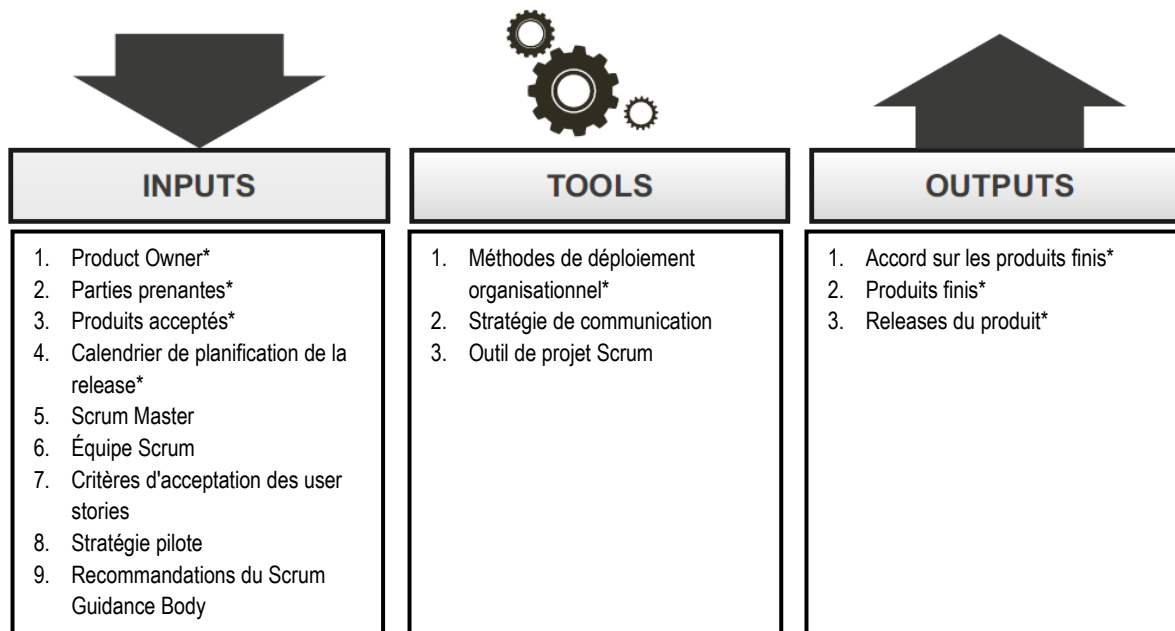


Figure 12-3 : Livraison des produits — Contributions, outils et résultats

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

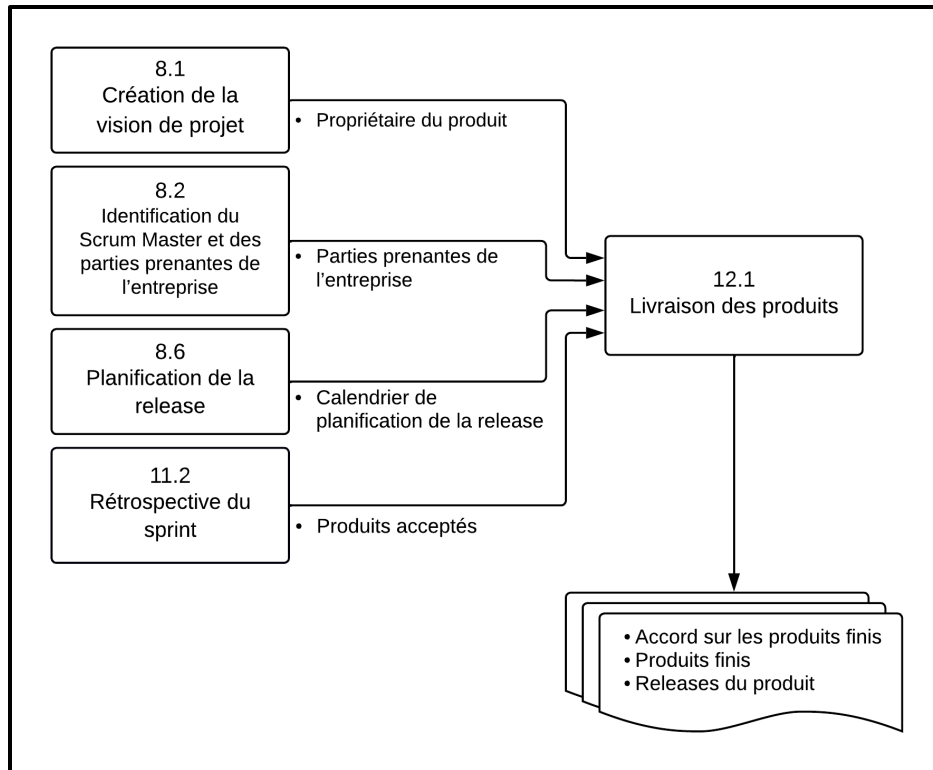


Figure 12-4 : Livraison des produits — Diagramme de flux de données

12.1.1 Contributions

12.1.1.1 Product Owner*

Décrit dans la section 8.1.3.1.

12.1.1.2 Parties prenantes de l'entreprise *

Décrites dans la section 8.2.3.2.

12.1.1.3 Produits acceptés*

Décrits dans la section 11.1.3.1.

12.1.1.4 **Calendrier de planification de la release**

Décrit dans la section 8.6.3.1.

12.1.1.5 **Scrum Master**

Décrit dans la section 8.2.3.1.

12.1.1.6 **Équipe Scrum**

Décrite dans la section 8.3.3.1.

12.1.1.7 **Critères d'acceptation des user stories**

Décrits dans la section 9.1.3.2.

12

12.1.1.8 **Stratégie pilote**

Une stratégie pilote est un résultat facultatif utilisé pour planifier un déploiement pilote en détail. La portée et les objectifs du déploiement, les utilisateurs cibles principaux du déploiement, un calendrier de déploiement, les programmes de transition, la préparation requise des utilisateurs, les critères d'évaluation pour le déploiement et d'autres éléments clés associés au déploiement sont spécifiés dans la stratégie pilote et partagés avec parties prenantes de l'entreprise.

12.1.1.9 **Recommandations du Scrum Guidance Body**

Décrites dans la section 8.1.1.11.

Durant le processus de *livraison des produits*, le Scrum Guidance Body peut fournir des recommandations et directives concernant le déploiement des produits. Ce sont des meilleures pratiques qui devraient être prises en compte lors du déploiement d'un produit auprès du client afin d'optimiser la valeur ajoutée offerte.

12.1.2 Outils

12.1.2.1 Méthodes de déploiement organisationnel*

Les méthodes de déploiement de chaque organisation ont tendance à varier en fonction du secteur, des utilisateurs cibles et du positionnement. Selon le produit qui doit être livré, le déploiement peut se dérouler à distance ou peut impliquer une livraison physique ou un produit de transition. Étant donné que le déploiement implique souvent un haut niveau de risque, les organisations disposent généralement de mécanismes de déploiement bien définis et établis avec en place des processus détaillés afin de garantir le respect des standards applicables et des mesures d'assurance qualité. Ces méthodes peuvent inclure la validation par un représentant spécifique de la direction, des mécanismes de validation par les utilisateurs et des directives concernant les fonctionnalités minimum pour une release.

12.1.2.2 Stratégie de communication

Les projets incluent souvent une stratégie de communication. Cette stratégie précise les rapports qui doivent être créés et maintenus durant le projet. Plusieurs méthodes différentes sont utilisées afin de communiquer aux parties prenantes de l'entreprise les informations importantes relatives au projet. La stratégie de communication définit ces méthodes et détermine la personne responsable des différentes activités de communication. Pendant que les incréments de produits sont testés, le statut des activités de test est communiqué selon la stratégie de communication établie par le Product Owner et le sponsor. Un mécanisme courant de communication consiste en un affichage visuel illustrant les informations importantes sous une forme facile à comprendre, disposé dans un lieu accessible et maintenu à jour avec les informations les plus récentes.

12.1.2.3 Outil de projet Scrum

Décrite dans la section 2.5.3.1.

12.1.3 Résultats

12.1.3.1 Accord sur les produits finis*

Les livrables qui sont acceptés reçoivent une signature commerciale formelle et une approbation du client et/ou du sponsor. L'obtention de l'acceptation formelle du client pour chaque produit livrable est essentielle pour la reconnaissance des bénéfices, l'acceptation des résultats globaux du projet et la réalisation des objectifs du projet.

12.1.3.2 Produits finis*

Cette version est le produit livrable final pour lequel le projet a été approuvé. Au fur et à mesure que les nouveaux incréments de produit sont créés, ils sont continuellement intégrés aux incréments précédents de façon à ce qu'un produit potentiellement livrable soit disponible à tout moment durant le projet.

12.1.3.3 Releases du produit*

Les releases du produit doivent inclure les éléments suivants :

- le contenu de la release — ce sont les informations essentielles concernant les produits qui peuvent aider l'équipe de service client.
- les notes de release — les notes de release doivent inclure les critères d'expédition externes ou adaptés au marché pour le produit à livrer.

12.2 Rétrospective de release

12

La figure 12-5 illustre les contributions, outils et résultats du processus de *rétrospective de projet*.

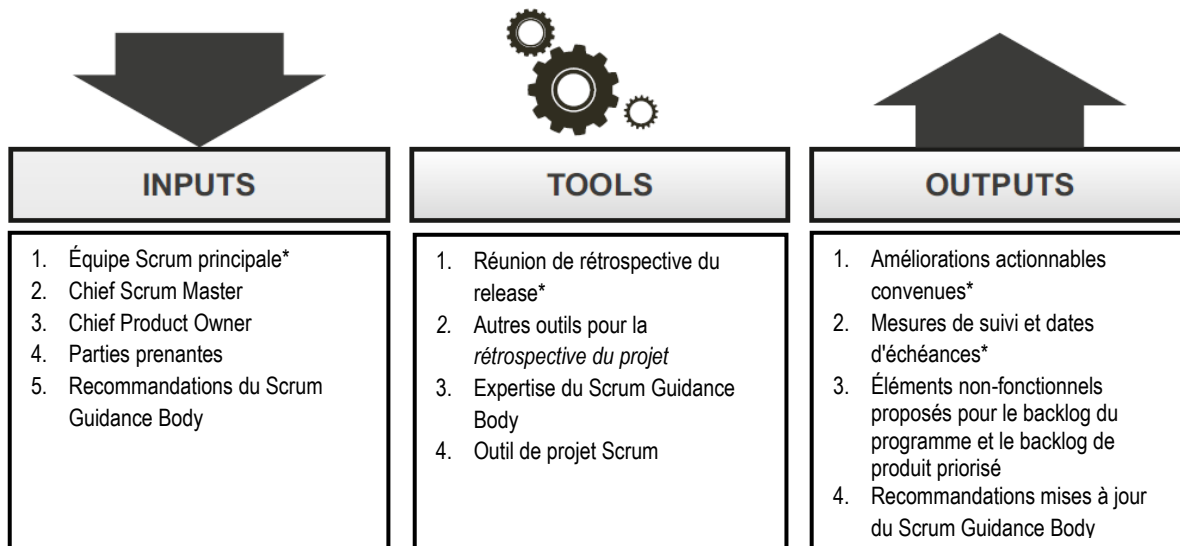


Figure 12-5 : Rétrospective de projet — Contributions, outils et résultats

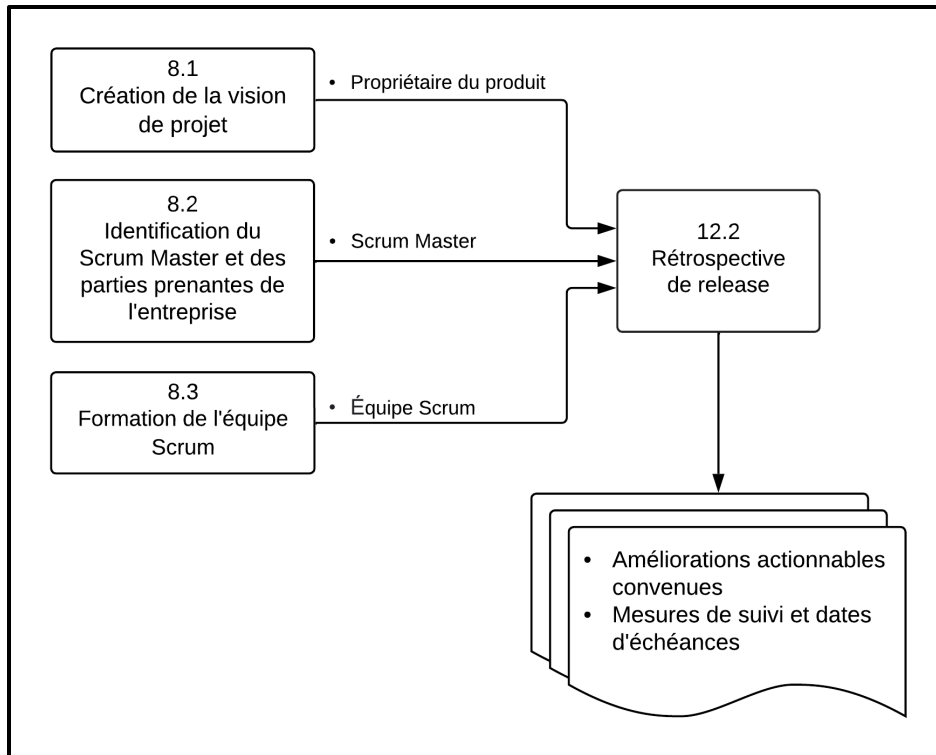


Figure 12-6 : Rétrospective de projet — Diagramme de flux de données

Remarque : Les astérisques (*) indiquent les contributions, outils ou résultats obligatoires pour le processus correspondant.

12.2.1 Contributions

12.2.1.1 Équipe Scrum principale*

Décrite dans la section 8.4.1.1.

12.2.1.2 Parties prenantes de l'entreprise

Décrites dans la section 8.2.3.2.

12.2.1.3 Recommandations du Scrum Guidance Body

Décrites dans la section 8.1.1.11.

Durant le processus de *Rétrospective de release*, les recommandations du Scrum Guidance Body peuvent inclure un répertoire de modèles internes afin de soutenir les futurs projets et des conseils afin d'organiser la réunion de *Rétrospective de release*. Les conseils fournis peuvent être relatifs aux procédures administratives, audits, évaluations et aux critères de transition du projet. Ils peuvent également souvent inclure la façon dont l'organisation maintient la base des connaissances des enseignements et informations des projets.

12.2.2 Outils

12.2.2.1 Réunion de *Rétrospective de release**

La réunion de *Rétrospective de release* est une réunion permettant de déterminer de quelle façon la collaboration au sein de l'équipe et l'efficacité peuvent être améliorées pour les futurs projets. Les points positifs, négatifs et les opportunités d'amélioration sont également abordés. La durée de cette réunion n'est pas limitée dans le temps et celle-ci peut se dérouler en personne ou virtuellement. Les participants incluent l'équipe du projet, le Chief Scrum Master, le Chief Product Owner et parties prenantes de l'entreprise. Durant la réunion, les enseignements sont documentés et les participants recherchent des opportunités pour améliorer les processus et résoudre les inefficacités.

12

12.2.2.2 Autres outils pour la *Rétrospective de release*

Certains outils utilisés lors du processus de *rétrospective de sprint* peuvent également être utilisés dans ce processus. Certains exemples incluent notamment :

- l'exercice Explorateur — Acheteur — Vacancier — Prisonnier (ESVP) (voir section 11.2.2.2)
- le speed boat (voir section 11.2.2.3)
- les indicateurs et méthodes de mesures

12.2.2.3 Expertise du Scrum Guidance Body

Abordée dans la section 8.4.2.7.

Durant le processus de *Rétrospective de release*, la principale responsabilité du Scrum Guidance Body est de garantir que les enseignements tirés de chaque projet ne sont pas perdus et sont intégrés à l'organisation. De plus, le Scrum Guidance Body peut offrir de l'expertise dans plusieurs domaines, notamment les RH et Scrum, ce qui peut faciliter le processus de *Rétrospective de release*. Les recommandations du Scrum Guidance Body peuvent également inclure des suggestions relatives à la façon de mener la réunion de *Rétrospective de release*.

12.2.2.4 **Outil de projet Scrum**

Décrite dans la section 2.5.3.1.

12.2.3 **Résultats**

12.2.3.1 **Améliorations actionnables convenues***

Décrites dans la section 11.3.3.1.

12.2.3.2 **Mesures de suivi et dates d'échéances***

Décrites dans la section 11.3.3.2.

12.2.3.3 **Éléments non-fonctionnels proposés pour le backlog du programme et le backlog de produit priorisé**

Lorsque le backlog de produit du programme ou le backlog de produit priorisé initial est développé, il est basé sur les user stories et les fonctionnalités requises. Souvent, les éléments non-fonctionnels peuvent ne pas être complètement définis au début du projet et peuvent apparaître pendant les réunions de revue de sprint, de rétrospective de sprint ou de Rétrospective de release. Ces éléments doivent être ajoutés au backlog du programme (dans le cas d'un programme) et au backlog de produit priorisé (dans le cas d'un projet) au fur et à mesure qu'ils sont découverts. Certains exemples d'exigences non-fonctionnelles sont les temps de réponse, les limites de capacité et les problèmes relatifs à la sécurité.

12.2.3.4 **Recommandations mises à jour du Scrum Guidance Body**

Décrites dans la section 11.2.3.6.

12.3 Diagramme de flux de données de la phase de release

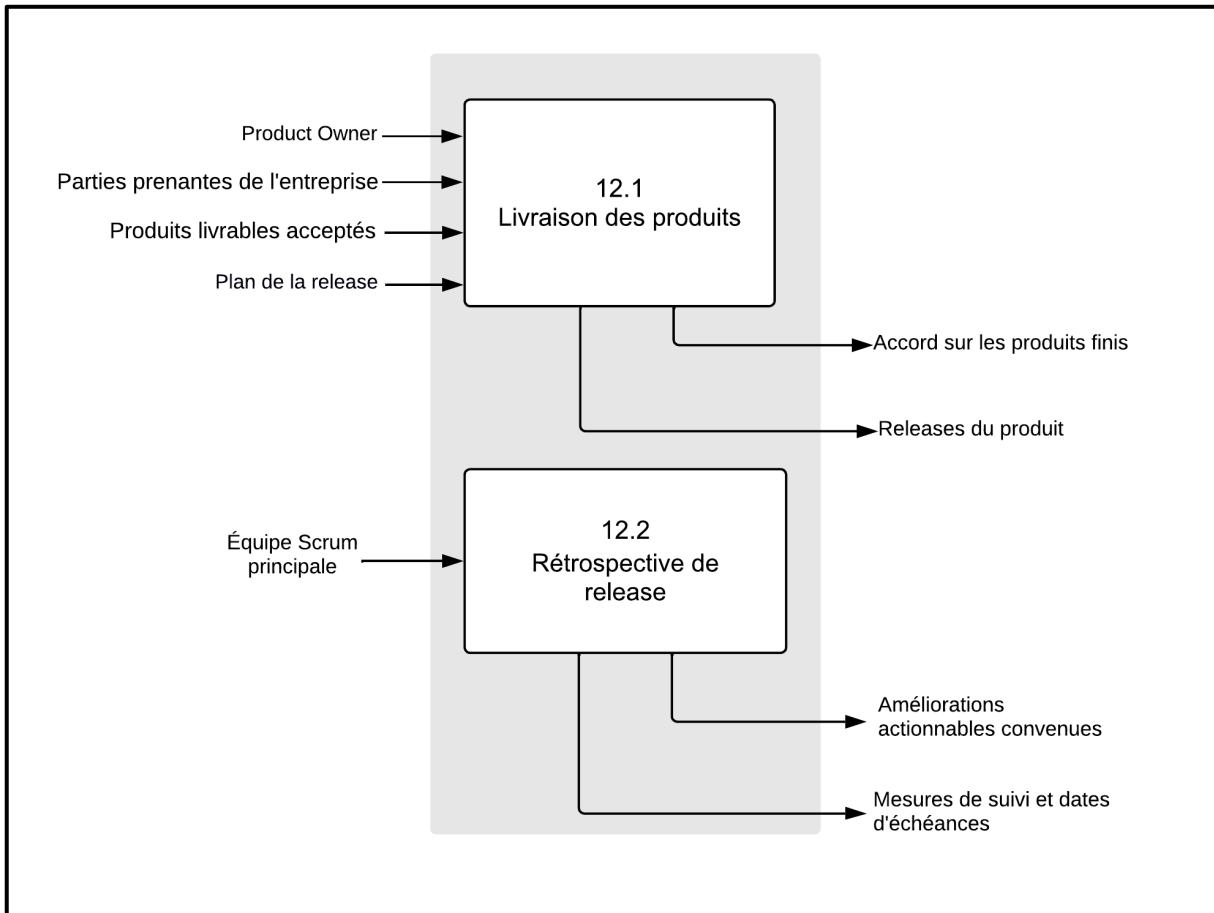


Figure 12-7 : Phase de release — Diagramme de flux de données

13. MISE À L'ÉCHELLE DE SCRUM POUR LES GRANDS PROJETS

Ce chapitre met l'accent sur des aspects supplémentaires de Scrum qui sont applicables aux grands projets. La mise à l'échelle de Scrum pour les grands projets, telle qu'elle est définie dans le *guide complet sur l'approche Scrum (guide SBOK®)*, s'applique aux projets suivants :

- Grands projets dans tous les secteurs d'activité
- Produits, services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise

Le terme « produit » dans le *guide SBOK®* peut faire référence à un produit, à un service ou à tout autre élément livrable. Scrum peut être appliqué efficacement non seulement aux petits projets dans n'importe quel secteur d'activité, mais aussi à des projets complexes et de grande envergure, dans lesquels des centaines de personnes peuvent être affectées à plusieurs équipes.

Outre les impacts qu'un grand projet peut avoir sur les processus fondamentaux de Scrum abordés dans les chapitres 8 à 12, ce chapitre présente des inputs, des outils et des outputs supplémentaires qui peuvent s'appliquer aux grands projets.

Pour faciliter l'application du cadre Scrum, ce chapitre identifie les inputs, outils et outputs supplémentaires comme étant « obligatoires » ou « optionnels ». Les inputs, outils et outputs accompagnés d'un astérisque (*) sont obligatoires ou considérés comme étant essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont optionnels.

Il est recommandé que l'équipe Scrum et les personnes qui se familiarisent avec le cadre et les processus Scrum se concentrent principalement sur les inputs, les outils et les outputs obligatoires, tandis que les Chief Product Owners, les Product Owners, les Chief Scrum Masters, les Scrum Masters et les autres praticiens de Scrum plus expérimentés s'efforcent d'acquérir une connaissance plus approfondie des informations contenues dans l'ensemble du chapitre. Il est également important de réaliser que, bien que tous les processus soient définis de manière unique dans le *guide SBOK®*, ils ne sont pas nécessairement exécutés séquentiellement ou séparément. Parfois, il peut être plus approprié d'exécuter certains processus en parallèle, en fonction des exigences spécifiques de chaque projet.

Ce chapitre est écrit du point de vue d'un grand projet qui coordonne les activités de plusieurs équipes Scrum afin de produire conjointement des incréments de produits potentiellement livrables/produits livrables. Des informations supplémentaires concernant l'application de Scrum à tout projet (grand ou petit) sont disponibles dans les chapitres 2 à 7, qui couvrent les principes de Scrum et les aspects de Scrum.

Grand projet Scrum par rapport à un petit projet Scrum classique

Les processus Scrum fondamentaux définis dans les chapitres 8 à 12 sont valables pour les projets Scrum avec un Product Owner, un Scrum Master et une à trois équipes Scrum. Ces projets sont généralement considérés comme de petits projets Scrum.

Lorsqu'il s'agit de grands projets nécessitant les efforts de quatre équipes Scrum ou plus, avec plusieurs Product Owners et plusieurs Scrum Masters, les processus fondamentaux définis dans les chapitres 8 à 12 restent valables, mais certaines considérations supplémentaires et mises à jour des inputs, des outils et des outputs peuvent être nécessaires. Cela peut inclure des efforts supplémentaires de coordination et de synchronisation. Les impacts sur les processus fondamentaux Scrum lors de la mise à l'échelle de Scrum à de grands projets sont décrits dans ce chapitre.

La définition de ce qui constitue un grand projet dépend généralement de l'organisation et/ou de la complexité des projets entrepris. Un critère clé pour déterminer si un projet est considéré comme petit ou grand est de savoir si le projet nécessite plusieurs Scrum Masters et/ou plusieurs Product Owners. Si le projet ne nécessite qu'un Scrum Master et un Product Owner, ces personnes peuvent normalement gérer les efforts supplémentaires de communication et de synchronisation requis par le projet.

Des inputs, des outils et des outputs supplémentaires peuvent être nécessaires pour les grands projets pour les raisons suivantes :

Équipes Scrum

- Interaction et dépendances accrues entre les équipes Scrum au fur et à mesure que la complexité augmente pour un grand projet
- Nécessité de gérer les conflits, de résoudre les problèmes, de gérer les dépendances et de fixer les priorités entre toutes les équipes Scrum
- Besoin de spécialisation, car certaines équipes Scrum peuvent avoir besoin de ressources spécialisées pour des tâches spécifiques (et ces compétences particulières ne sont pas nécessaires dans toutes les équipes Scrum)
- Nécessité de définir certaines lignes directrices et normes qui doivent être respectées par toutes les équipes Scrum (par exemple, les normes de sécurité au sein d'une entreprise ou les lignes directrices juridiques et gouvernementales pour des industries spécifiques) ; celles-ci peuvent être définies par le Scrum Guidance Body
- Nécessité de mettre en place un environnement ou une zone de travail pour le grand projet, qui serait ensuite utilisé par toutes les équipes Scrum
- Nécessité de coordonner les outputs de plusieurs équipes Scrum pour faciliter la release d'un grand projet

Scrum Masters

- Nécessité d'une collaboration entre les Scrum Masters pour gérer les obstacles et synchroniser le travail des différentes équipes Scrum

Product Owners

- Nécessité d'une collaboration entre les Product Owners lorsqu'ils travaillent avec les parties prenantes de l'entreprise, de raffiner le Product Backlog priorisé et de travailler avec les équipes Scrum.

Il est également important de noter qu'à mesure que Scrum est adapté aux grands projets, des services de soutien supplémentaires peuvent être nécessaires, tels que des architectes, des gestionnaires de produits, des services de conformité, de sécurité de l'information, des organes de gouvernance, etc.

13.1 Impact des grands projets sur les processus fondamentaux Scrum

Bien que les processus fondamentaux Scrum décrits dans les chapitres 8 à 12 restent valables pour les projets de grande envergure, il convient de noter certaines considérations supplémentaires. Les tableaux 13-1 à 13.5 présentent un résumé des impacts d'un grand projet sur les processus fondamentaux Scrum pour chaque phase du projet.

13.1.1 Lancement

La phase de lancement d'un grand projet a les mêmes objectifs et suit le même déroulement que la phase de lancement d'un petit projet Scrum classique.

Par rapport à un projet Scrum classique, des rôles supplémentaires doivent être identifiés et des activités supplémentaires doivent être réalisées pour parvenir à un accord sur la manière dont les multiples Product Owners, Scrum Masters et équipes Scrum collaboreront entre eux et avec les parties prenantes de l'entreprise.

| Processus | Résumé des impacts d'un grand projet |
|-------------------------------------|---|
| 8.1 Création de la vision de projet | <p>Un Chief Product Owner et des Product Owners supplémentaires sont identifiés dans ce processus. Dans un grand projet, le Chief Product Owner a la responsabilité globale du projet et travaille avec le sponsor et d'autres parties prenantes de l'entreprise pour créer la vision du projet. En outre, le plan de collaboration des Product Owners est créé.</p> <p>Output supplémentaire : Identification du Chief Product Owner* (voir section 3.7.2.1) Dans le cadre d'un grand projet, le Chief Product Owner assume la responsabilité globale du projet et travaille avec le sponsor et d'autres parties prenantes de l'entreprise pour créer la vision du projet. Dans le cadre de ce processus, le Chief Product Owner est identifié.</p> <p>Output mis à jour : Product Owner identifié* (voir section 8.1.3.1) Comme il y a plusieurs Product Owners dans un grand projet, ils doivent également être identifiés dans ce processus.</p> <p>Output supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2) Pour les grands projets, il sera essentiel que l'ensemble de l'équipe des Product Owners adopte Scrum et collabore pour mener à bien les projets Scrum.</p> |

| Processus | Résumé des impacts d'un grand projet |
|---|---|
| 8.2 Identification du Scrum Master et de la/des partie(s) prenante(s) de l'entreprise | <p>Un Chief Scrum Master et des Scrum Masters supplémentaires sont identifiés dans ce processus. Le Chief Scrum Master se concentre sur l'interaction et la synchronisation de plusieurs équipes. Certains outputs supplémentaires ou mis à jour pour ce processus sont les suivants.</p> <p>Output supplémentaire : Chief Scrum Master identifié* (voir section 3.7.2.2) Tout comme le Chief Product Owner, le Chief Scrum Master doit également être identifié pour un grand projet. Il/elle se concentre sur l'interaction et la synchronisation de plusieurs équipes.</p> <p>Output mis à jour : Scrum Masters identifiés* En effet, dans un grand projet, il y a plusieurs Scrum Masters identifiés dans ce processus.</p> <p>Output supplémentaire : Organisation Scrum pour grands projets* (voir section 13.2.1) Il sera essentiel que l'ensemble de l'équipe d'un grand projet adopte une compréhension commune de Scrum et des accords de travail pour mener à bien le projet Scrum.</p> <p>Output supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3) Le plan de collaboration des Scrum Masters définit la manière dont les Scrum Masters et les équipes participent au raffinement du Product Backlog priorisé. Ce plan définirait également les représentants de l'équipe qui participeraient au processus de raffinement, et la manière dont ils sont sélectionnés.</p> <p>Output supplémentaire : Ressources partagées* (voir section 13.2.4) La connaissance de toutes les ressources partagées disponibles pour les équipes Scrum serait un input nécessaire à la formation des équipes Scrum individuelles.</p> <p>Output mis à jour : Services de soutien identifiés (voir section 3.3.2) En plus de l'identification des services de soutien, pour un grand projet, certains services de soutien supplémentaires peuvent être nécessaires pour coordonner les activités entre tous les Product Owners, les Scrum Masters et les équipes Scrum.</p> |

8.3 Formation de l'équipe Scrum

Dans un grand projet, plusieurs équipes Scrum sont formées, impliquant plusieurs Product Owners et Scrum Masters, ainsi qu'un Chief Product Owner et un Chief Scrum Master. Le Chief Product Owner et le Chief Scrum Master sont impliqués dans la détermination de la formation des équipes Scrum et ils apportent également leur input lors de la sélection des membres de l'équipe. Le Chief Product Owner et le Chief Scrum Master servent les intérêts du projet dans son ensemble, tandis que les Product Owners et les Scrum Masters se concentrent davantage sur leurs équipes Scrum respectives.

Input supplémentaire : Chief Product Owner* (voir section 3.7.2.1)

Dans le cas d'un grand projet, le Chief Product Owner participera à la détermination de la formation des équipes Scrum et fournira son input sur les membres de ces équipes. Le Chief Product Owner sert les intérêts du grand projet dans son ensemble, tandis que les Product Owners se concentrent sur les équipes respectives.

Input supplémentaire : Chief Scrum Master* (voir section 3.7.2.2)

Pour un grand projet, le Chief Scrum Master participe à la détermination de la formation des équipes Scrum et contribue à la sélection des membres de ces équipes. Le Chief Scrum Master sert les intérêts du grand projet dans son ensemble, tandis que les Scrum Masters se concentrent sur les équipes respectives.

Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3)

Input supplémentaire : Spécialisation de l'équipe* (voir section 13.2.5)

Certaines équipes Scrum et certains membres de l'équipe Scrum peuvent avoir besoin de compétences spécialisées pour travailler sur des questions spécifiques liées à de grands projets.

Outil supplémentaire : Plan de communication pour les grands projets* (voir section 13.3.1)

Ce plan met en évidence la manière de gérer une communication efficace entre toutes les personnes impliquées dans un grand projet.

Outil supplémentaire : Planification des ressources pour les grands projets*

Elle permet de gérer la complexité de l'affectation de divers types de ressources aux nombreuses équipes Scrum travaillant en parallèle.

Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)

Puisque les grands projets impliquent un grand nombre de personnes, ils sont complexes et requièrent une interaction importantes entre les équipes Scrum. Il est donc recommandé d'utiliser un outil de projet Scrum pour automatiser les processus, gérer la complexité, générer des rapports, gérer la communication entre les parties prenantes de l'entreprise, etc.

| | |
|--|---|
| <p>8.3 Formation de l'équipe Scrum (suite)</p> | <p>Outil supplémentaire : Identification de l'environnement*</p> <p>Dans les grands projets, il est important d'identifier le nombre et les types d'environnements nécessaires car de nombreuses équipes Scrum travailleront simultanément pour réaliser les travaux de leurs sprints respectifs.</p> <p>Output supplémentaire : Environnement et calendrier de l'environnement* (voir section 13.2.6)</p> <p>Une fois les environnements identifiés, un calendrier d'environnement est créé, qui est utilisé pour la coordination des activités du sprint entre les équipes.</p> <p>Output supplémentaire : Plan de collaboration des Scrum Masters mis à jour* (voir section 13.2.3)</p> <p>Au fur et à mesure que les équipes Scrum sont formées, les inputs des équipes et les considérations supplémentaires généreront des mises à jour du plan de collaboration des Scrum Masters.</p> <p>Outil supplémentaire : Identification de l'environnement*</p> <p>Dans les grands projets, il est important d'identifier le nombre et les types d'environnements nécessaires car de nombreuses équipes Scrum travailleront simultanément pour réaliser les travaux de leurs sprints respectifs.</p> |
|--|---|

| | |
|--|---|
| 8.4 Développement d'une ou plusieurs Epic(s) | <p>Toute interaction du Product Owner avec le client et les autres parties prenantes de l'entreprise est gérée par le Chief Product Owner et les multiples Product Owners, plutôt que par un seul Product Owner. La manière dont cette interaction est divisée est définie dans le plan de collaboration du Product Owner. Toute interaction avec les Scrum Masters et/ou les membres de l'équipe Scrum et toute participation de ces derniers se déroule conformément au plan de collaboration des Scrum Masters. Sinon, la création des Epics se fait de la même manière que dans un petit projet Scrum classique.</p> <p>Input supplémentaire : Plan de collaboration des Product Masters* (voir section 13.2.2)</p> <p>Il définit comment les multiples Product Owners travaillent ensemble et avec le Chief Product Owner. Il traite de la façon dont ils travaillent avec les parties prenantes de l'entreprise pour collecter les exigences, mettre à jour le Product Backlog priorisé et travailler avec les différentes équipes Scrum. Il n'y aura qu'un seul Product Owner en interaction directe avec chaque équipe Scrum. Cependant, des décisions doivent être prises concernant la répartition des équipes Scrum entre les Product Owners, et le nombre d'équipes Scrum avec lesquelles chaque Product Owner travaillera.</p> <p>Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3)</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |
| 8.5 Création du Product Backlog priorisé | <p>En règle générale, les différents Product Owners contribuent à la création du Product Backlog priorisé, comme convenu dans le plan de collaboration des Product Owners. Les décisions finales de priorisation et la résolution des conflits entre les Product Owners sont traitées par le Chief Product Owner. Sinon, le processus est géré de la même manière que dans un petit projet Scrum classique.</p> <p>Input supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2)</p> <p>Comme le plan de collaboration des Product Owners définit comment les Product Owners font les mises à jour du Product Backlog priorisé, il s'agit d'un input important pour ce processus.</p> <p>Outil supplémentaire : Product Backlog priorisé Affectations* (voir section 13.3.4)</p> <p>Comme le Chief Product Owner et plusieurs Product Owners sont impliqués dans un grand projet, les affectations prioritaires du Product Backlog priorisé assurent l'attribution efficace des Epics et User Stories à tous les Product Owners.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |

| | |
|--|---|
| 8.6 Exécution de la planification de release | <p>Pour un grand projet, le calendrier de planification de release est créé par le Chief Product Owner. Comme les releases peuvent être plus compliquées dans le cas de grands projets, un plan de préparation de release (Release Readiness Plan) est créé au cours du processus (si nécessaire), qui est ensuite utilisé pour confirmer que les exigences de chaque release ont été satisfaites. Sinon, le processus est géré de la même manière que dans un projet Scrum classique</p> <p>Input supplémentaire : Chief Scrum Master* (voir section 3.7.2.2)</p> <p>Outil supplémentaire en option : Sprint de préparation de release (Release Readiness Sprint) (voir section 13.3.7) Parfois, un sprint de préparation de release doit être planifié pour la release globale du projet. Les exigences d'un tel sprint sont ensuite ajoutées au Product Backlog priorisé.</p> <p>Output supplémentaire : Plan de préparation de release* (voir section 13.2.7) Le plan de préparation de release comprend des activités spécifiques qui doivent être réalisées peu avant la planification de release.</p> |
|--|---|

Tableau 13-1: Impact des grands projets sur les processus Scrum fondamentaux - Phase d'initiation

13.1.2 Planification et estimation

Comme dans un projet Scrum classique/de petite taille, la phase de planification et d'estimation d'un grand projet a pour objectif de définir et de planifier le travail qui sera réalisé au cours du prochain sprint. Chaque équipe Scrum travaille avec son Scrum Master respectif et le Product Owner pour s'engager à effectuer le travail spécifique du sprint et pour planifier la manière dont l'équipe effectuera le travail (comme dans un projet Scrum typique).

Par rapport à un petit projet Scrum, des mesures supplémentaires doivent être prises pour répartir le travail entre les différentes équipes Scrum. Le choix des équipes et des membres d'équipe participant à la création et à l'estimation d'une User Story, et la manière dont la validation des User Stories et la création des produits livrables respectifs sont répartis entre les différentes équipes, sont basés sur la spécialisation de l'équipe et le plan de collaboration.

| Processus | Résumé des impacts d'un grand projet |
|---------------------------------|---|
| 9.1 Création des User Stories | <p>Dans un grand projet, plusieurs Product Owners et plusieurs équipes Scrum sont impliqués dans la création des User Stories. Tous les Product Owners et tous les membres d'équipe Scrum/toutes les équipes Scrum ne peuvent pas être impliqués dans la création de chaque User Story. Ainsi, la création des User Stories est répartie entre les multiples Product Owners sur la base de la stratégie de collaboration des Product Owners, ainsi qu'entre les multiples équipes Scrum sur la base de la spécialisation des équipes et du plan de collaboration. Sinon, le processus se déroule de la même manière que dans un petit projet Scrum typique.</p> <p>Input supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2)</p> <p>Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3)</p> <p>Outil supplémentaire : Identification de l'environnement* (voir section 13.3.3) Dans un grand projet, il est important d'identifier le nombre et les types d'environnements nécessaires pour que les équipes Scrum puissent travailler efficacement.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |
| 9.2 Estimation des User Stories | <p>Dans un grand projet, tous les membres de l'équipe Scrum ne peuvent pas être impliqués dans l'estimation de chaque User Story. Par conséquent, seules certaines équipes/certains membres de l'équipe participent généralement à l'estimation des User Stories. L'équipe ou les membres de l'équipe qui estiment les User Stories sont choisis en fonction de la spécialisation de l'équipe et du plan de collaboration.</p> <p>Le Product Owner qui travaille avec les équipes Scrum respectives dépend généralement de celui qui a créé les User Stories respectives et est le mieux placé pour répondre à toutes les questions liées aux User Stories en cours d'estimation. Cela est également basé sur le plan de collaboration des Product Owners.</p> <p>Sinon, l'estimation des User Stories est faite de la même manière que dans un projet Scrum typique.</p> <p>Input supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2)</p> <p>Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3)</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |

| Processus | Résumé des impacts d'un grand projet |
|-----------------------------------|---|
| 9.3 Validation des User Stories | <p>Pour chaque sprint d'un grand projet, il est demandé à chaque équipe de valider un sous-ensemble spécifique de User Stories pour le sprint et ensuite de créer les produits livrables respectifs. La décision de savoir quelle équipe est invitée à valider et à implémenter quelles User Stories, dépend des compétences spécifiques de chaque équipe et est basée sur la spécialisation de l'équipe.</p> <p>Sinon, la validation des User Stories est gérée de la même manière que dans un petit projet Scrum typique, sur la base des priorités, des estimations et de la vélocité propre à l'équipe.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |
| 9.4 Identification des tâches | <p>L'identification des tâches est faite par chaque équipe pour décomposer les User Stories que l'équipe a validées. Chaque équipe Scrum identifie les tâches de la même manière que dans un petit projet Scrum classique. Cependant, l'identification correcte des dépendances aide les équipes Scrum à déterminer lesquelles de leurs décisions et activités peuvent avoir un impact sur les autres équipes et peut également influencer l'ordre relatif dans lequel une équipe Scrum exécute ses tâches respectives pour créer les produits livrables du sprint.</p> <p>Outil mis à jour : Détermination des dépendances*</p> <p>Description dans la section 9.4.2.3. Dans les grands projets, l'identification correcte des dépendances aide les équipes Scrum à déterminer lesquelles de leurs décisions et activités peuvent avoir un impact sur les autres équipes. Elle peut également influencer l'ordre relatif dans lequel une même équipe Scrum exécute ses tâches respectives pour créer les produits livrables du sprint.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |
| 9.5 Estimation des tâches | <p>Les tâches identifiées dans le processus précédent seront éventuellement estimées, exactement comme pour un petit projet Scrum classique.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |
| 9.6 Mise à jour du Sprint Backlog | <p>Chaque équipe met à jour son Sprint Backlog spécifique en fonction de ses User Stories et des tâches associées. Chaque équipe met à jour le Sprint Backlog exactement comme cela se fait dans un petit projet Scrum classique.</p> <p>Input supplémentaire : Environnement et calendrier de l'environnement (voir section 13.2.6)</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |

Tableau 13-2: Impact des grands projets sur les processus Scrum fondamentaux - Phase de planification et d'estimation

13.1.3 Implémentation

Dans la phase d'implémentation d'un grand projet, chaque équipe Scrum, sous la direction de son Scrum Master, crée les produits livrables qui sont associés aux User Stories qui ont été validées en poursuivant et achevant les tâches que l'équipe a identifiées dans la phase de planification et d'estimation.

Par rapport à un petit projet Scrum, des mesures supplémentaires doivent être prises pour garantir une communication efficace entre les différentes équipes (comme défini dans le plan de communication) et la synchronisation du travail (comme détaillé dans le plan de collaboration).

Comme dans un projet Scrum classique/de petite taille, pendant que les équipes Scrum créent les produits livrables du sprint, le Chief Product Owner et les autres Product Owners raffinent le Product Backlog priorisé pour le maintenir à jour en fonction des modifications apportées aux exigences et/ou aux priorités. Ils s'assurent également que l'ensemble des User Stories qu'ils souhaitent que les équipes Scrum s'engagent à livrer au cours du prochain Sprint seront prêtes pour l'estimation et la validation.

Sur un grand projet, les Product Owners se répartissent le travail, sur la base du plan de collaboration des Product Owners. Les interactions avec les équipes Scrum et/ou les membres de l'équipe Scrum sont basées sur la spécialisation de l'équipe Scrum et le plan de collaboration.

| Processus | Résumé des impacts d'un grand projet |
|--------------------------------------|--|
| 10.1 Création des produits livrables | <p>Chaque équipe crée les produits livrables associés aux User Stories qu'elle a validées.</p> <p>Contrairement à un petit projet Scrum classique, l'équipe Scrum d'un grand projet est confrontée à des limites dans sa liberté d'organiser son travail. Ces limitations sont dues à la nécessité de partager les ressources avec d'autres équipes (comme spécifié et convenu dans les ressources partagées) ainsi que l'environnement et le calendrier environnemental, et également en raison des impacts potentiels des décisions que les autres équipes peuvent prendre.</p> <p>En outre, des activités supplémentaires sont nécessaires pour assurer une bonne communication et une bonne synchronisation entre les équipes Scrum.</p> <p>Input supplémentaire : Plan de préparation de release* (voir section 13.2.7)</p> <p>Outil supplémentaire : Réunions Scrum of Scrums* (voir section 13.3.5)</p> <p>Il s'agit de réunions ciblées où les représentants de l'équipe Scrum se rencontrent pour partager le statut de leurs équipes respectives.</p> <p>Outil supplémentaire : Méthodes de préparation de release* (voir section 13.3.6)</p> <p>Ce sont les méthodes utilisées afin d'exécuter les tâches identifiées dans le plan de préparation de release de façon à ce que les produits livrables soient prêts pour l'expédition/la release.</p> |
| 10.2 Mêlée quotidienne | <p>Chaque équipe conduit sa mêlée quotidienne exactement comme elle le fait dans un projet Scrum classique. Cependant, comme chaque Scrum Master peut travailler avec plusieurs Scrum Teams, un effort de coordination est nécessaire pour éviter les conflits de calendrier entre les mêlées quotidiennes de leurs Scrum Teams respectives.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |

| Processus | Résumé des impacts d'un grand projet |
|--|--|
| 10.3 Raffinage du Product Backlog priorisé | <p>Les différences par rapport à un petit projet Scrum classique sont les mêmes différences que dans les processus de <i>développement des Epics</i> et de <i>création de Product Backlog priorisé</i>. Plus précisément, toute interaction du Product Owner avec le client et les autres parties prenantes de l'entreprise est gérée par le Chief Product Owner et/ou les multiples Product Owners plutôt que par un seul Product Owner. La façon dont cette interaction est divisée est définie dans le plan de collaboration des Product Owners. En outre, les décisions finales de priorisation sont prises par le Chief Product Owner.</p> <p>Toute interaction avec les Scrum Masters et/ou les membres de l'équipe Scrum et toute participation de ces derniers se déroule conformément au plan de collaboration des Scrum Masters. Autrement, le raffinage du Product Backlog priorisé est traité comme dans un projet Scrum classique.</p> <p>Input supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2)</p> <p>Le plan de collaboration des Product Owners définit la manière dont les Product Owners mettent à jour le Product Backlog priorisé.</p> <p>Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3)</p> <p>Input supplémentaire : Spécialisation de l'équipe* (voir section 13.2.5)</p> <p>Outil supplémentaire : Outil de projet Scrum (voir section 13.3.8)</p> <p>Outil supplémentaire en option : Sprint de préparation de release* (voir section 13.3.7)</p> <p>Outputs supplémentaires : Mise à jour du plan de préparation de release* (voir section 13.2.7)</p> <p>Les modifications du Product Backlog priorisé apportées au cours du raffinage du Product Backlog priorisé peuvent avoir un impact sur le plan de préparation de release.</p> <p>Outil supplémentaire en option : Outil de projet Scrum (voir section 13.3.8)</p> |

Tableau 13-3: Impact des grands projets sur les processus fondamentaux Scrum—phase d'implémentation

13.1.4 Révision et rétrospective

Dans la phase de révision et de rétrospective d'un grand projet, chaque équipe Scrum présente les produits livrables qu'elle a créés au cours d'un sprint à son Product Owner respectif pour approbation et commentaires, puis se réunit pour déterminer les moyens d'améliorer continuellement son travail. Cela se fait de la même façon que dans un projet Scrum classique.

En outre, les Product Owners se réunissent pour déterminer les moyens d'améliorer leur travail et, comme étape finale d'un sprint, le Chief Product Owner et le Chief Scrum Master rencontrent les Product Owners, les Scrum Masters et les membres de l'équipe de Scrum concernés pour partager les résultats de leurs réunions de rétrospective de sprint respectives. Étant donné que chaque Scrum Master et chaque Product Owner peut travailler avec plusieurs équipes Scrum, un effort de coordination est nécessaire pour éviter les conflits de calendrier entre les réunions de révision de sprint et/ou les réunions de rétrospective de sprint de différentes équipes Scrum.

| Processus | Résumé des impacts d'un grand projet |
|---|--|
| 11.1 Présentation et validation du sprint | <p>Ce processus est réalisé individuellement par chaque équipe Scrum. Pour chaque équipe, le Product Owner respectif approuve les User Stories terminées. Cependant, cela peut être quelque peu complexe en raison des interdépendances. Il peut arriver que le Chief Product Owner assiste à des réunions de révision de sprint pour certaines équipes Scrum qui ont des produits livrables interdépendants.</p> <p>Input supplémentaire : Chief Product Owner* (voir section 3.7.2.1)</p> |

| Processus | Résumé des impacts d'un grand projet |
|------------------------------|--|
| 11.2 Rétrospective de sprint | <p>Chaque équipe Scrum du projet se réunit avec son Scrum Master pour une réunion de rétrospective de sprint qui se déroule de la même manière que dans un petit projet Scrum classique. Parce qu'un seul Scrum Master et un seul Product Owner peuvent travailler avec plusieurs Scrum Teams, un effort de coordination est nécessaire pour éviter les conflits de calendrier entre les réunions de rétrospective de sprint des différentes équipes Scrum. En outre, le Chief Product Owner et les Product Owners se réunissent pour une réunion de rétrospective de sprint afin de discuter de leur collaboration et d'autres aspects du sprint.</p> <p>En outre, les Scrum Masters et/ou d'autres représentants de chaque équipe Scrum se réunissent pour une réunion Scrum of Scrums (SoS) spéciale afin de partager les meilleures pratiques et d'autres résultats des réunions de rétrospective de sprint des différentes équipes ; par exemple, les problèmes de collaboration inter-équipes. Parce que dans un grand projet, très souvent les meilleures pratiques et les problèmes proviennent de la collaboration entre les multiples équipes Scrum et l'équipe des Product Owners, il est courant que le Chief Product Owner et les autres Product Owners participent à cette réunion.</p> <p>Input supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2)</p> <p>Le raffinement de l'arrière de produits classés par ordre de priorité peut être particulièrement difficile lorsque de grands projets sont concernés. S'il n'est pas effectué efficacement, le raffinement peut causer des problèmes et gaspiller les efforts des équipes. Par conséquent, il est recommandé de discuter du raffinement dans le cadre de la rétrospective, en se concentrant spécifiquement sur la façon dont les multiples Product Owners interagissent entre eux et avec les équipes Scrum pour mener efficacement le raffinement du Backlog. En outre, il est préférable de regrouper des Epics et des User Stories ayant beaucoup de dépendances.</p> <p>Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3).</p> <p>Pour les grands projets, le raffinement du Product Backlog priorisé peut s'avérer particulièrement difficile. S'il n'est pas effectué efficacement, le raffinement peut créer des problèmes et gaspiller les efforts des équipes. Par conséquent, il est recommandé de discuter du raffinement dans le cadre de la rétrospective, en mettant l'accent sur l'interaction entre les différents Scrum Masters, les équipes Scrum et la façon dont ils interagissent avec les Product Owners pour les activités de raffinement.</p> |

Tableau 13-4: Impact des grands projets sur les processus Scrum fondamentaux - Phase de révision et de rétrospection

13.1.5 Release

Dans la phase de validation d'un grand projet, les Product Owners de chaque équipe Scrum coordonnent avec le Chief Product Owner pour s'assurer que les produits livrables de toutes les équipes Scrum sont synchronisés, intégrés et validés comme l'exigent le client et les autres parties prenantes de l'entreprise.

En outre, le plan de collaboration des Product Owners et des Scrum Masters est raffiné pour assurer une meilleure coordination entre toutes les équipes d'un grand projet.

| Processus | Résumé des impacts d'un grand projet |
|---------------------------------------|---|
| 12.1 Livraison des produits livrables | <p>Dans un grand projet, tous les produits livrables acceptés des sprints précédents sont généralement livrés ou transmis aux parties prenantes de l'entreprise de la même manière que dans un petit projet Scrum classique. Toutefois, en fonction de considérations commerciales et en raison de la complexité des grands projets, des étapes de préparation supplémentaires peuvent être nécessaires pour se préparer à la release.</p> <p>Input supplémentaire : Chief Product Owner (voir section 3.7.2.1)</p> <p>Input supplémentaire : Chief Scrum Master (voir section 3.7.2.2)</p> <p>Input supplémentaire : Plan de préparation de release (voir section 13.2.7)</p> |

| Processus | Résumé des impacts d'un grand projet |
|-------------------------------|--|
| 13.2 Rétrospective de release | <p>Input supplémentaire : Chief Product Owner* (voir section 3.7.2.1)</p> <p>Input supplémentaire : Chief Scrum Master* (voir section 3.7.2.2)</p> <p>Input supplémentaire : Plan de collaboration des Product Owners* (voir section 13.2.2)</p> <p>Pour les grands projets, le raffinement du Product Backlog priorisé peut s'avérer particulièrement difficile. S'il n'est pas effectué efficacement, le raffinement peut causer des problèmes et gaspiller les efforts des équipes. Par conséquent, il est recommandé de discuter du raffinement dans le cadre de la rétrospective, en se concentrant spécifiquement sur la façon dont les multiples Product Owners interagissent entre eux et avec les équipes Scrum pour mener efficacement le raffinement du Backlog. En outre, il est préférable de regrouper des Epics et des User Stories ayant beaucoup de dépendances.</p> <p>Input supplémentaire : Plan de collaboration des Scrum Masters* (voir section 13.2.3)</p> <p>Pour les grands projets, la coordination et la gestion des activités et des interdépendances entre plusieurs Scrum Masters et Scrum Teams peuvent s'avérer très difficiles. Il est donc recommandé de disposer d'un plan de collaboration des Scrum Masters et de le raffiner en permanence pour garantir une meilleure collaboration entre tous les Scrum Masters et les équipes Scrum.</p> |

Tableau 13-5: Impact des grands projets sur les processus Scrum fondamentaux - Phase de Release

13.2 Inputs et outputs supplémentaires pour les grands projets

13.2.1 Organisation de Scrum pour les grands projets*

La structure organisationnelle appropriée nécessaire pour implémenter et soutenir les grands projets est définie en consultation avec le Chief Product Owner, le Chief Scrum Master, la direction générale, le Scrum Guidance Body et d'autres experts concernés.

Les organisations qui prévoient d'utiliser Scrum pour implémenter de grands projets devraient adopter pleinement le cadre de Scrum. L'organisation doit être en mesure de soutenir les efforts à fournir en engageant les ressources nécessaires. Si l'organisation ne peut pas engager les ressources requises, il faut prévoir de se procurer les ressources nécessaires telles que les personnes, les outils et l'espace de travail. Il est impératif qu'une organisation qui envisage d'utiliser Scrum soit prête à changer radicalement sa culture et ses habitudes de travail afin d'en tirer véritablement profit.

Dans un environnement Scrum où de grands projets sont développés, il y aura de nombreuses équipes Scrum en compétition pour les ressources. Il est donc important de gérer les ressources organisationnelles de manière optimale pour atteindre les objectifs globaux du projet. L'organisation Scrum d'un grand projet doit prendre en compte les éléments qui seront développés, les compétences, les coûts et les autres ressources nécessaires à leur développement, la vitesse actuelle des équipes Scrum (ou la vitesse estimée/supposée) pour fournir une estimation de haut niveau de la durée du projet, les exigences en matière de communication et les autres interfaces que les équipes Scrum doivent maintenir.

13.2.2 Plan de collaboration des Product Owners*

Le Chief Product Owner travaille avec les principaux Product Owners pour créer le plan de collaboration des Product Owners. Le plan de collaboration des Product Owners définit spécifiquement comment les Product Owners doivent collaborer avec le Chief Product Owner. Au minimum, ce plan doit définir le nombre d'équipes Scrum auxquelles un Product Owner peut être affecté (généralement en fonction de facteurs tels que l'expérience, le temps et la connaissance du domaine), la manière dont le travail de collecte des exigences auprès des parties prenantes de l'entreprise sera réparti entre les Product Owners, comment le Product Backlog priorisé sera mis à jour avec de nouvelles exigences ou des changements dans les exigences, et comment les Product Owners collaboreront avec plusieurs équipes Scrum. Il convient de noter que chaque équipe de Scrum ne collaborera toujours qu'avec un seul Product Owner ; toutefois, un seul Product Owner peut collaborer avec plus d'une équipe Scrum, si nécessaire.

13.2.3 Plan de collaboration des Scrum Masters*

Le Chief Scrum Master travaille avec les principaux Scrum Masters, le Scrum Guidance Body et parfois certains membres déjà identifiés des équipes Scrum pour créer le plan de collaboration des Scrum Masters. Le plan de collaboration de l'équipe Scrum définit comment les différentes équipes Scrum collaborent ensemble de façon à offrir la meilleure valeur ajoutée dans le délai le plus court possible.

Dans le cadre d'un projet de grande envergure, plusieurs Scrum Masters sont généralement affectés à ce projet, chacun d'entre eux facilitant et assurant un environnement de travail productif pour son équipe Scrum respective. Il est possible qu'un Scrum Master puisse animer plus d'une équipe Scrum en même temps. Les Scrum Masters doivent collaborer avec d'autres Scrum Masters ainsi qu'avec le Chief Scrum Master, le Chief Product Owner et les Product Owners pour établir la liste des éléments et des ressources nécessaires en commun à toutes les équipes tout au long du projet. Ils contribuent également à la création du plan de préparation de release.

Le plan doit comprendre des informations sur les domaines spécialisés attribués aux équipes qualifiées, sur la manière dont les équipes soutiendront le raffinage et l'estimation du Product Backlog priorisé (c.-à-d., quels membres de l'équipe participeront aux sessions d'affinage et aux exercices d'estimation de haut niveau), et comment les équipes organiseront les réunions Scrum of Scrums (SoS). Il peut également être nécessaire d'utiliser un outil de projet Scrum pour faciliter l'utilisation de Scrum sur de grands projets.

Le plan de collaboration des Scrum Masters peut également inclure des informations sur la manière dont chaque équipe Scrum sera encadrée. Par exemple, il peut inclure des informations sur la question de savoir s'il y aura un coach distinct en plus du Scrum Master, s'il y aura un Scrum Master à chaque endroit dans le cas d'équipes distribuées, comment les membres de l'équipe collaboreront avec les Scrum Masters situés au même endroit et avec les Scrum Masters qui ne sont pas situés au même endroit, etc.

Bien que les équipes Scrum travaillant sur un grand projet doivent interagir les unes avec les autres lors de la création du Product Backlog priorisé, les Product Owners peuvent s'assurer que les User Stories ayant beaucoup de dépendances soient regroupées et appartiennent à un seul Product Owner afin que les produits livrables soient réalisés par une seule ou quelques équipes Scrum. Cela permet de minimiser les dépendances des tâches entre les différentes équipes Scrum, et à chaque équipe Scrum de travailler efficacement pour créer ses produits livrables.

13.2.4 Ressources partagées*

Les ressources partagées peuvent inclure les personnes, environnements et équipements nécessaires à toutes ou certaines des équipes Scrum travaillant sur le projet. Dans le cas d'un grand projet, les ressources partagées peuvent être limitées et requises par toutes ou certaines des équipes Scrum au même moment. Dans ce contexte, le Chief Product Owner le Chief Scrum Master, les autres Product Owners et les autres Scrum Masters doivent convenir d'une méthode d'allocation des ressources partagées.

Un exemple de méthode d'allocation des ressources partagées pourrait consister à s'assurer que les ressources sont affectées en priorité aux équipes travaillant sur les fonctionnalités les plus importantes/de plus grande valeur. Lorsque des demandes concurrentes ont des priorités très similaires, le Chief Product Owner doit décider de l'allocation des ressources en fonction des exigences commerciales actuelles, des priorités et d'autres critères définis.

13.2.5 Spécialisation de l'équipe*

Trois considérations relatives à la spécialisation des équipes doivent être prises en compte lors de l'application de Scrum à de grands projets. La première dimension est le besoin de réaliser un ensemble de tâches spécifiques. Par exemple, une équipe spécialisée peut être une équipe d'intégration qui possède une connaissance approfondie de l'intégration continue. Ces connaissances peuvent être particulièrement importantes lors de la réalisation d'un sprint de préparation de release (si des tâches spécifiques doivent être effectuées avant la release).

La deuxième dimension est le besoin de compétences spécifiques pour certains membres de l'équipe. Théoriquement, tous les membres de l'équipe Scrum sont des généralistes et des spécialistes en ce sens qu'ils ont des connaissances générales dans divers domaines et sont experts dans au moins un d'entre eux. Toutefois, cela peut ne pas être le cas dans un grand projet. Les membres de l'équipe peuvent être amenés à posséder des compétences très spécifiques – par exemple une connaissance d'un domaine particulier comme la sécurité – qui peuvent ne pas être disponibles dans toutes les équipes travaillant sur le grand projet pour lequel cette compétence est nécessaire. Dans ce cas, il serait extrêmement coûteux de former tout le monde dans tous les domaines spécialisés requis.

Des experts ayant des compétences et des connaissances spécialisées peuvent être affectés au projet Scrum et intervenir selon les besoins dans différentes équipes. Cependant, il peut parfois s'avérer nécessaire de recruter des experts auprès de sources externes lorsqu'ils sont nécessaires, en gardant à l'esprit que l'ajout d'un nouveau membre de l'équipe aura un impact sur la vélocité de l'équipe.

La troisième dimension pour l'acquisition d'experts est due aux limitations éventuelles de la flexibilité de l'équipe. Typiquement, dans un projet Scrum, chaque équipe aura un ou plusieurs domaines dans lesquels elle possède une expertise significative, quelques domaines sur lesquels elle peut travailler avec un peu d'input et de formation, et d'autres domaines dans lesquels elle n'a pas les compétences ou l'expérience. Pendant la planification du sprint, pour chaque équipe, il y aura un sous-ensemble de User Stories qui sont logiquement assignées à l'équipe en fonction de leur expertise, certaines sur lesquelles ils peuvent travailler, et d'autres sur lesquelles ils ne peuvent pas travailler parce qu'ils n'ont pas les connaissances ou les compétences requises. Dans ce cas, il peut être nécessaire de faire appel à des experts lorsque les compétences des membres de l'équipe ne sont pas suffisamment flexibles. L'accès limité aux ressources ayant des compétences spécialisées dans un grand projet Scrum entraîne un certain niveau de risque pour le projet. Certaines User Stories hautement prioritaires peuvent ne pas pouvoir être terminées en un seul sprint. Les équipes peuvent être amenées à travailler sur certaines User Stories de moindre priorité en attendant la disponibilité des membres de l'équipe ayant des compétences spécialisées.

13.2.6 Environnement et calendrier de l'environnement*

Les environnements de développement requis ne sont pas toujours disponibles pendant toute la durée d'un grand projet. Par exemple, une grue nécessaire pour une activité de construction ou un environnement d'essai spécialisé peut n'être disponible que certains jours. Une fois que tous les environnements requis sont identifiés, un calendrier des environnements est créé et utilisé pour la coordination des activités du sprint dans le processus de *mise à jour du Sprint Backlog*. Le calendrier des environnements est un calendrier détaillant comment les équipes de Scrum accéderont à des environnements spécifiques et les partageront. Le calendrier alloue des jours et plages horaires pour l'utilisation de chaque environnement par les équipes respectives.

13.2.7 Plan de préparation de release*

Le Chief Product Owner travaille avec les autres Product Owners pour créer le plan de préparation de release. Le plan de préparation de release détaille les mesures à prendre par les équipes Scrum concernées et toute autre personne pour confirmer que les exigences minimales pour la release ont été satisfaites et que le produit ou l'incrément de produit est prêt pour la release. Les décisions commerciales et leurs justifications commerciales associées pour l'exécution des tâches de préparation de la release sont également décrites dans le plan de préparation de release.

Étant donné que chaque sprint crée un produit ou un autre produit livrable potentiellement expédiable, dans un petit projet Scrum classique, une release peut avoir lieu après n'importe quel sprint lorsque cela s'avère utile pour l'entreprise. Cependant, dans un projet Scrum de grande envergure, il se peut que certaines activités liées à la préparation de la release doivent être réalisées pour certains sprints du projet. Par exemple, une équipe de projet peut avoir besoin d'exécuter une série complète de tests de performance coûteux et longs ou de réaliser une série spéciale de tests d'intégration de bout en bout juste avant une release. Ces activités sont considérées en dehors des critères « Done » (Terminé) définis pour les sprints réguliers, et dans de tels cas, un sprint de préparation de release à part sera nécessaire (voir section 13.3.7) pour terminer les tâches nécessaires à la préparation d'une release.

13.3 Outils supplémentaires pour les grands projets

13.3.1 Plan de communication pour les grands projets

Le plan de communication d'un grand projet est créé par le Chief Product Owner, le Chief Scrum Master et le Scrum Guidance Body, avec les inputs des autres Product Owners, Scrum Masters, Scrum Teams et des autres personnes concernées.

Un plan de communication pour les grands projets est essentiel dans un grand projet, car une mauvaise communication ou un manque de communication sur le projet peut être préjudiciable aux efforts de collaboration et peut finalement entraîner l'échec du projet. Un plan de communication doit comprendre des informations sur la manière dont les communications seront effectuées au sein de toutes les équipes Scrum et avec les parties prenantes de l'entreprise. Le plan de communication comprend les méthodes de communication à utiliser, les canaux ou mécanismes de communication des informations clés, les responsabilités en matière de communication, la classification des informations sensibles et les moyens de les traiter, le calendrier des activités de communication et les processus d'évaluation de l'efficacité de la communication. Le plan de communication doit également inclure le calendrier et la fréquence des réunions Scrum of Scrums (SoS) et la manière dont ces réunions seront menées.

Chaque équipe Scrum peut également avoir son propre plan de communication (voir section 12.1.3.4) qui spécifie les documents qui doivent être créés et distribués, et comment ces documents seront conservés tout au long du projet. Le plan doit également inclure les méthodes à utiliser pour transmettre les informations importantes du projet aux parties prenantes de l'entreprise et qui est responsable des différentes activités de communication.

Étant donné qu'il peut être très difficile pour tous les participants à un grand projet d'être situés au même endroit, l'utilisation d'un outil de projet Scrum peut contribuer à faciliter la communication.

13.3.2 Planification des ressources pour les grands projets*

La planification des ressources pour les grands projets est essentielle en raison de la complexité de l'affectation de divers types de ressources aux nombreuses équipes Scrum travaillant en parallèle. Il y aura des besoins concurrents pour des ressources limitées, et le Chief Product Owner et les autres Product Owners doivent planifier le processus pour fournir les meilleurs résultats dans le laps de temps le plus court possible. La planification des ressources dans un grand projet doit prendre en compte les différents coûts associés aux ressources telles que les personnes, la formation, le matériel et les logiciels, les services externes et l'espace physique.

Le Chief Product Owner et les autres Product Owners peuvent être amenés à se coordonner avec des sources externes pour acquérir des ressources et renforcer le personnel (par exemple, il peut être nécessaire d'embaucher des ressources externes pour travailler avec l'équipe existante à temps plein et d'interagir avec l'équipe de gestion des fournisseurs existante au sein de l'organisation). Lors de l'embauche de ressources externes, le Chief Product Owner et l'équipe doivent se conformer aux politiques de l'organisation en matière de relations avec les ressources externes et les vendeurs.

Dans les grands projets, le Chief Product Owner peut également être amené à envisager la planification de ressources supplémentaires pour répondre aux besoins d'équipes spécialisées et à la nécessité de mettre en place des environnements pour de nombreuses équipes Scrum travaillant en parallèle.

Le Chief Product Owner et les Product Owners peuvent collaborer avec les Scrum Masters et les équipes Scrum pour définir les compétences spécialisées requises pour le grand projet, le nombre de ressources nécessaires, les équipes Scrum qui ont besoin de compétences spécialisées et l'estimation de l'allocation.

13.3.3 Identification de l'environnement*

Le Chief Scrum Master travaille avec les autres Scrum Masters, les Product Owners concernés, les membres de l'équipe Scrum, les services de soutien, le Scrum Guidance Body et d'autres experts, le cas échéant, pour identifier les environnements nécessaires et appropriés qui seront nécessaires pour mener à bien le grand projet. Cela devrait se faire de préférence une fois au cours de la phase de lancement ou lorsque les équipes et/ou le SGB le demandent.

Dans un grand projet, il est important d'identifier le nombre et les types d'environnements nécessaires car de nombreuses équipes Scrum travailleront simultanément pour réaliser les travaux de leurs sprints respectifs et les besoins en matière d'environnement pourraient être compliqués et entrer en conflit. Parmi les exemples d'environnements, citons les zones de développement ou de test de logiciels, les zones de travail physique ou les environnements comportant des équipements spécialisés. Les limites du processus pour chaque équipe Scrum peuvent également affecter les environnements. De plus, avec des équipes distribuées travaillant dans des fuseaux horaires différents, il peut être possible d'effectuer des tests 24 heures sur 24 et de maximiser l'utilisation de différents environnements. Il est donc impératif de créer un calendrier de l'environnement qui indique les horaires des tests pour chaque équipe. Pour les projets logiciels, le calendrier des environnements peut également inclure des informations sur la manière dont le code sera promu dans chaque environnement et par qui.

13.3.4 Affectation du Product Backlog priorisé*

Dans un grand projet, le Product Backlog priorisé (avec ses Epics et User Stories correspondantes) est créé par le Chief Product Owner et les autres Product Owners de la même manière que pour les petits projets Scrum classiques.

La création d'Epics et de User Stories par les Product Owners peut dépendre de multiples facteurs tels que :

- Comment les exigences ont été recueillies auprès des parties prenantes de l'entreprise
- Connaissance et expérience/compétences des Product Owners

Le Chief Product Owner et les autres Product Owners travaillent ensemble pour passer en revue le Product Backlog priorisé et déterminer quels Epics et User Stories seront gérées par chaque Product Owner. Bien que certaines Epics et User Stories puissent être créées par un Product Owner particulier, elles peuvent être assignées à un autre Product Owner pour qu'il les gère et les implémente dans le cadre du grand projet Scrum. Les Epics et les User Stories doivent être priorisées et certaines estimations préexistantes peuvent être utilisées pour faciliter leur affectation à différents Product Owners.

L'attribution des Epics et des User Stories aux Product Owners est influencée par d'autres facteurs tels que :

- Le Product Owner qui a créé l'Epic/User Story (car il/elle sera souvent responsable de son implémentation)
- Le Product Owner qui dispose de l'équipe avec les compétences appropriées pour exécuter l'Epic/User Story
- Le client, le sponsor ou l'organisation spécifique lié à l'Epic/User Story (car il pourrait y avoir une relation passée avec un Product Owner particulier)
- Le nombre de dépendances se rapportant à chaque Epic/User Story (car le regroupement des User Stories avec beaucoup de dépendances sous un seul Product Owner pourrait permettre aux équipes de travailler de manière relativement indépendante les unes des autres, sans avoir besoin de passer trop de temps sur la coordination avec les autres équipes)

L'attribution et la hiérarchisation des Epics ou des User Stories aux Product Owners peuvent être faites sur une certaine période de temps par des réunions et si possible, par l'utilisation d'un outil de projet Scrum.

13.3.5 Réunion Scrum of Scrums (SoS)*

Une réunion Scrum of Scrums (SoS) est un élément important lors de la mise à l'échelle de Scrum pour de grands projets. L'objectif de la plupart des réunions Scrum of Scrum est la synchronisation des équipes pendant la création des produits livrables, mais elles peuvent également être utilisées pour partager les meilleures pratiques après les réunions de rétrospective de sprint et pour la planification des sprints à venir. La fréquence des réunions SoS est spécifique au projet et dépend de la taille et de la complexité du projet, des dépendances entre les différentes équipes, etc. Si les Epics ou les User Stories dans les sprints peuvent être terminés sans trop d'interaction avec d'autres Epics ou User Stories, ces réunions peuvent ne pas être nécessaires trop souvent ; en revanche, si les dépendances sont élevées, une fréquence plus élevée de réunions SoS peut être nécessaire.

En général, il y a un représentant de chaque équipe de Scrum à la réunion SoS – habituellement le Scrum Master – mais il est également courant que d'autres membres d'une équipe Scrum assistent à la réunion si cela semble pertinent. Normalement, cette réunion est facilitée par le Chief Scrum Master et est censée se concentrer sur les points de coordination et d'intégration entre les différentes équipes Scrum.

Il s'agit de préférence de courtes réunions où au moins un représentant de chaque équipe Scrum (par exemple, le Scrum Master et/ou d'autres) se réunissent pour partager l'état d'avancement du travail effectué par leurs équipes respectives, un peu à la manière de la mêlée quotidienne. Elles ne sont généralement pas limitées dans le temps pour permettre à tous les représentants de partager leurs informations, même pour les très grands projets. La réunion Scrum of Scrums (SoS) se tient à intervalles prédéterminés ou lorsque les équipes de Scrum le demandent, afin de faciliter le partage nécessaire d'informations entre les différentes équipes Scrum. Les problèmes, les dépendances et les risques ayant un impact sur plusieurs équipes Scrum peuvent être suivis de près, ce qui aide les multiples équipes travaillant sur un grand projet à mieux coordonner et intégrer leur travail.

Il incombe au Chief Scrum Master (ou à un autre Scrum Master qui anime les réunions SoS) de veiller à ce que tous les représentants disposent d'un environnement propice à un partage ouvert et honnête de l'information, y compris le retour d'information aux autres représentants de l'équipe. Pour les projets plus importants, impliquant un nombre significatif d'équipes, des réunions SoS à plusieurs niveaux peuvent être tenues pour partager le statut de chaque équipe respective.

Chaque représentant de l'équipe Scrum fournit à son tour des mises à jour de son équipe. Ces mises à jour sont généralement fournies sous la forme de réponses aux quatre questions spécifiques suivantes.

1. Sur quoi mon équipe a-t-elle travaillé depuis notre dernière réunion ?
2. Que va faire mon équipe d'ici à la prochaine réunion ?
3. Quelles sont les tâches que d'autres équipes attendent de notre équipe et qui restent à faire ?
4. Qu'est-ce que notre équipe prévoit de faire qui pourrait affecter les autres équipes ?

Les réponses à ces quatre questions fournissent des informations qui permettent à chaque équipe de comprendre clairement l'état d'avancement du travail de toutes les autres équipes et s'il y a ou pourrait y avoir des problèmes avec les livraisons à venir.

13.3.6 Méthodes de préparation de release (Release Preparation Methods)*

Ce sont les méthodes utilisées afin d'exécuter les tâches identifiées dans le plan de préparation de release de façon à ce que les produits livrables soient prêts pour l'expédition/la release. Ces méthodes peuvent être spécifiques à un projet, mais sont plus souvent valables au niveau d'un programme ou d'un portfolio. Elles peuvent être définies par le Scrum Guidance Body.

13.3.7 Sprint de préparation de release

Si des tâches spécifiques doivent être effectuées pour préparer la release et pour garantir que les exigences minimales ont été remplies, ces tâches sont effectuées lors d'un sprint de préparation de release. Dans un sprint de préparation de release, aucune User Story du Product Backlog priorisé n'est développée. Au lieu de cela, les tâches identifiées dans le plan de préparation de release (voir 13.2.7) sont exécutées. Un sprint de préparation de release n'est réalisé qu'une seule fois par release, comme première étape du processus de *livraison des produits livrables*. La durée d'un sprint de préparation de release peut être différente de celle des autres sprints.

En présence d'un sprint de préparation de release, les critères « Done » (Terminé) correspondants sont généralement uniques et différent des critères de « Done » (Terminé) des User Stories pour les autres sprints (qui doivent toujours être respectés). Les critères « Done » (Terminé) sont définis dans le but de garantir que les produits livrables du sprint sont « potentiellement expédiables ». Le sprint de préparation de release concerne toutes les activités qui ne sont réalisées qu'une seule fois par release, sur la base de décisions commerciales délibérées justifiées dans le plan de préparation de release.

Un sprint de préparation de release n'est pas obligatoire (), à moins qu'il n'existe une décision commerciale justifiée de l'intégrer au projet. En outre, seuls les membres de l'équipe concernés sont généralement impliqués dans le sprint de préparation de release. Other Scrum Team members not involved in the Release Readiness Sprint can begin to work on other regular Sprints.

13.3.8 Outil de projet Scrum

Étant donné que les grands projets impliquent de nombreuses équipes Scrum avec plusieurs centaines de personnes (ou plus) travaillant sur le projet, et comme les équipes peuvent également être distribuées, il peut y avoir une complexité et des interactions importantes entre les équipes. Il serait donc bénéfique pour les équipes d'avoir accès à un outil de projet Scrum structuré ou à un ensemble d'outils permettant d'automatiser les processus, de gérer la complexité, de partager les informations (entre les équipes et avec les parties prenantes de l'entreprise), de générer des rapports, etc.

Certaines des tâches spécifiques qui pourraient être gérées par l'outil de projet Scrum sur un grand projet incluent :

- Capacité à former des équipes avec des rôles appropriés et capacité à faire évoluer les rôles vers des projets de grande envergure
- Capacité à créer et à maintenir un Product Backlog priorisé pour chaque équipe, y compris la création, l'estimation et la gestion des Epics, User Stories et des tâches
- Capacité à prendre en charge d'autres artefacts importants du projet Scrum, tels que les Scrumboards, les Sprint Backlogs, les calendriers et les comptes rendus des réunions, etc
- Capacité à permettre une communication transparente et efficace entre tous les membres de l'équipe de projet
- Capacité à soutenir des équipes distribuées (voir section 2.5.3)
- Capacité à soutenir la création de rapports et de mesures, comme l'exigent les différents rôles de Scrum

Capacité à saisir et à partager les recommandations ou l'expertise du Scrum Guidance Body (par exemple, les leçons tirées des rétrospectives de sprint, les meilleures pratiques, les politiques organisationnelles liées à Scrum, etc.).

14. La mise à l'échelle de Scrum pour l'entreprise

Ce chapitre met l'accent sur des aspects supplémentaires de Scrum qui sont applicables aux programmes et aux portfolios. La mise à l'échelle de Scrum pour l'entreprise, telle que définie dans le *guide complet sur l'approche Scrum (guide SBOK®)*, est applicable aux éléments suivants :

- Programmes, portfolios et/ou projets dans n'importe quel secteur d'activité
- Produits, services ou tout autre résultat destinés à être livrés aux parties prenantes de l'entreprise

Dans le *guide SBOK®*, le terme « programme » fait référence à un ensemble de projets et/ou de sous-programmes connexes qui doivent être gérés de manière coordonnée pour produire et livrer les éléments du programme. Le terme « portfolio » fait référence à un ensemble de programmes et/ou de projets au sein d'une même organisation qui peuvent ou non être directement liés les uns aux autres, et qui peuvent ou non avoir besoin d'être gérés de manière coordonnée pour atteindre les objectifs du portfolio. Du point de vue du cadre Scrum, les programmes et les portfolios peuvent être traités de manière similaire, mais à des niveaux différents de l'entreprise et nécessitent donc potentiellement des niveaux de coordination différents pour les programmes et/ou projets sous-jacents.

Ce chapitre traite des impacts (sur les inputs, les outils et les outputs) qu'un programme ou un portfolio a sur les processus Scrum fondamentaux décrits dans les chapitres 8 à 12. Ce chapitre présente également des processus supplémentaires qui s'appliquent uniquement aux programmes et aux portfolios qui ne sont pas pertinents au niveau du projet.

Pour faciliter l'application du cadre Scrum, ce chapitre identifie les inputs, les outils et les outputs de chaque processus comme étant « obligatoires » ou « optionnels ». Les inputs, outils et outputs accompagnés d'un astérisque (*) sont obligatoires ou considérés comme étant essentiels à la réussite d'un projet, tandis que ceux sans astérisque sont optionnels.

Il est recommandé que les personnes qui s'initient à l'application du cadre Scrum pour l'entreprise se concentrent principalement sur les inputs, les outils et les outputs obligatoires, tandis que les Program/Portfolio Product Owners, les Chief Product Owners, les Product Owners, les Program/Portfolio Scrum Masters, les Chief Scrum Masters, les Scrum Masters et les autres praticiens de Scrum plus expérimentés s'efforcent d'acquérir une connaissance plus approfondie des informations contenues dans ce chapitre. Il est également important de réaliser que, bien que tous les processus soient définis de manière unique dans le *guide SBOK®*, ils ne sont pas nécessairement exécutés séquentiellement ou séparément. Parfois, il peut être plus approprié de réaliser certains processus en parallèle ou de manière itérative, en fonction des exigences spécifiques de chaque programme ou portfolio.

Ce chapitre est rédigé du point de vue d'une seule équipe de programme ou de portfolio qui coordonne et hiérarchise les activités de plusieurs projets et/ou programmes Scrum sous-jacents. Des informations supplémentaires concernant l'application de Scrum au niveau du projet sont disponibles dans les chapitres 2 à 7, qui couvrent les principes de Scrum et les aspects de Scrum.

Scrum d'entreprise vs. projet unique de Scrum

Lorsqu'il s'agit de Scrum au niveau d'une entreprise, il peut y avoir plusieurs centaines d'équipes Scrum, avec plusieurs milliers de personnes travaillant sur de multiples projets au sein de programmes et/ou portefeuilles dans l'entreprise. L'application des processus Scrum au niveau d'un programme ou d'un portefeuille aura certains impacts sur les projets sous-jacents. En général, les projets Scrum sont toujours exécutés en utilisant les processus Scrum fondamentaux discutés dans les chapitres 8 à 12 pour les petits projets classiques, avec les inputs, les outils et les outputs supplémentaires décrits dans le chapitre 13 pour les grands projets (ayant plusieurs Product Owners et/ou Scrum Masters).

Les impacts des programmes et des portefeuilles sur les processus de niveau projet de Scrum décrits dans les chapitres 8 à 12 sont décrits dans la section 14.1 du présent chapitre comme des inputs, des outils et des outputs supplémentaires. Les processus et considérations supplémentaires qui ne sont pertinents qu'au niveau du programme ou du portefeuille sont traités dans les sections 14.2 à 14.8.

Certaines des questions qui se posent au niveau du programme ou du portefeuille sont similaires à celles qui se posent sur un grand projet Scrum. La synchronisation entre les équipes et la collaboration globale sont les plus grands défis rencontrés dans un grand projet Scrum, et ces défis existent également au niveau du programme ou du portefeuille. Les plus grands défis pour un programme ou un portefeuille, cependant, peuvent se produire du côté commercial, parce que les priorités commerciales de différents projets peuvent entrer en conflit les unes avec les autres et peuvent parfois aussi entrer en conflit avec les objectifs généraux du programme ou du portefeuille et doivent être alignées.

Comme dans un grand projet Scrum, des inputs, des outils et des outputs supplémentaires sont nécessaires pour faire face aux efforts supplémentaires de priorisation, d'alignement et de coordination. Les raisons pour lesquelles les programmes et/ou les portefeuilles ont besoin d'inputs, d'outils et d'outputs supplémentaires sont les suivantes :

Product Owners

- Programme et portefeuille—nécessité d'aligner des objectifs commerciaux contradictoires
- Programme—nécessité d'une collaboration entre le Program Product Owner et les Product Owners des projets dans le programme, tels que :
 - le raffinement du Program Backlog priorisé,
 - en interaction avec les parties prenantes de l'entreprise pour synchroniser les messages, et
 - éviter la duplication des travaux au sein du programme (c'est-à-dire, la synergie)
- Portefeuille—nécessité d'une collaboration entre le Portfolio Product Owner, les Program Product Owners, et les Product Owners des programmes et des projets dans le portefeuille, tels que :
 - le raffinement du Portfolio Backlog priorisé,
 - en interaction avec les parties prenantes de l'entreprise pour synchroniser les messages, et
 - éviter la duplication des travaux dans le portefeuille (c'est-à-dire la synergie)

Scrum Masters

- Programme et portfolio—nécessité d'une collaboration entre les Scrum Masters lors de la résolution des problèmes
- Programme—synchroniser le travail des équipes Scrum de plusieurs projets, si nécessaire
- Portfolio—synchronisation du travail des équipes Scrum de plusieurs programmes et projets, si nécessaire

Équipes Scrum

- Programme et portfolio—nécessité de gérer les dépendances entre les équipes Scrum
- Programme et portfolio—nécessité de gérer les ressources partagées et les éventuels conflits de ressources entre les équipes Scrum
- Programme et portfolio—nécessité de définir certaines lignes directrices et normes qui doivent être respectées par les équipes Scrum pour tous les projets du programme ou du portfolio (par exemple, les normes de sécurité au sein de l'organisation ou les réglementations légales et/ou gouvernementales pour des industries spécifiques)—ces éléments peuvent devoir être documentés par le Scrum Guidance Body.
- Programme et portfolio—nécessité de mettre en place et de maintenir un environnement qui sera utilisé par plusieurs équipes Scrum

14.1 Impact des programmes ou des portfolios sur les processus fondamentaux Scrum

Les tableaux 14-1 à 14-4 présentent un résumé des impacts des programmes et des portfolios sur les processus fondamentaux Scrum et des portfolios sur les processus fondamentaux Scrum pour chaque phase du projet.

14.1.1 Lancement

Les données supplémentaires provenant du niveau du programme/portfolio qui doivent être prises en compte pour la phase de lancement sont les suivantes :

| Processus | Résumé des impacts d'un programme ou d'un portfolio |
|-------------------------------------|---|
| 8.1 Création de la vision de projet | <p>La vision du projet est créée avec l'input supplémentaire du programme/portfolio auquel le projet appartient. En dehors de ce point, il n'y a pas d'autre changement dans ce processus.</p> <p>Input supplémentaire : Program/Portfolio Product Owner Les rôles du Program/Portfolio Product Owner sont décrits dans les sections 3.7.4.1 et 3.7.4.2.</p> <p>Input supplémentaire : Program/Portfolio Scrum Master Les rôles du Program/Portfolio Scrum Master sont décrits dans les sections 3.7.4.3 et 3.7.4.4.</p> <p>Input supplémentaire : Parties prenantes de l'entreprise du programme/portfolio Les parties prenantes de l'entreprise du programme/portfolio sont décrites dans la section 14.3.3.5. Elles influencent tous les projets dans le programme/portfolio.</p> <p>Input supplémentaire : Program/Portfolio Backlog priorisé Le Program/Portfolio Backlog priorisé est décrit dans la section 14.6.1.2. Le Program/Portfolio Backlog priorisé contient les exigences du programme ou du portfolio qui peuvent avoir un impact sur la vision du projet.</p> |

| Processus | Résumé des impacts d'un programme ou d'un portfolio |
|---|--|
| 8.2 Identification du Scrum Master et de la/des partie(s) prenante(s) de l'entreprise | <p>Le Scrum Master et la/les partie(s) prenante(s) de l'entreprise du projet sont identifiés avec l'input supplémentaire du programme/portfolio auquel le projet appartient. En dehors de ce point, il n'y a pas d'autre changement dans ce processus.</p> <p>Input supplémentaire : Program/Portfolio Product Owner Les rôles du Program/Portfolio Product Owner sont décrits dans les sections 3.7.4.1 et 3.7.4.2.</p> <p>Input supplémentaire : Program/Portfolio Scrum Master Les rôles du Program/Portfolio Scrum Master sont décrits dans les sections 3.7.4.3 et 3.7.4.4.</p> <p>Input supplémentaire : Parties prenantes de l'entreprise du programme/portfolio Les parties prenantes de l'entreprise du programme/portfolio sont décrites dans la section 14.3.3.5. Elles influencent tous les projets du programme ou du portfolio.</p> |
| 8.3 Formation de l'équipe Scrum | Il n'y a pas d'impact supplémentaire à ce processus pour un projet dans un programme ou un portfolio. |

| Processus | Résumé des impacts d'un programme ou d'un portfolio |
|--|--|
| 8.4 Développement d'une ou plusieurs Epic(s) | <p>Les Epics sont développés avec la contribution supplémentaire du programme/portfolio auquel le projet appartient. En dehors de ce point, il n'y a pas d'autre changement dans ce processus.</p> <p>Input supplémentaire : Program/Portfolio Backlog priorisé Le Program/Portfolio Backlog priorisé est décrit dans la section 14.6.1.2. Le Program/Portfolio Backlog priorisé contient les exigences du programme ou du portfolio qui peuvent avoir un impact sur le développement d'Epics.</p> <p>Input supplémentaire : Risques du programme/portfolio Les risques du programme et du portfolio sont décrits dans les sections 7.6.1 et 7.6.2.</p> <p>Les risques liés à un portfolio ou programme ont également un impact sur les projets faisant partie du programme ou du portfolio en question. Pendant l'évaluation des risques du programme ou du portfolio, s'il est déterminé qu'un risque peut affecter un projet individuel, les informations pertinentes sur le risque doivent être communiquées au Product Owner et à l'équipe Scrum. Les risques du programme et du portfolio deviennent des inputs du processus de <i>développement d'une ou plusieurs Epic(s)</i> et ces risques peuvent avoir un impact sur la façon dont ce processus est mené.</p> |
| 8.5 Création du Product Backlog priorisé | <p>Le Product Backlog priorisé est créé avec des données supplémentaires provenant du Program/Portfolio Backlog priorisé. En dehors de ce point, il n'y a pas d'autre changement dans ce processus.</p> <p>Input supplémentaire : Program/Portfolio Backlog priorisé Le Program/Portfolio Backlog priorisé est décrit dans la section 14.6.1.2. Le Program/Portfolio Backlog priorisé contient les exigences du programme/portfolio qui peuvent avoir un impact sur la création du Product Backlog priorisé.</p> |

| Processus | Résumé des impacts d'un programme ou d'un portfolio |
|--|---|
| 8.6 Exécution de la planification de release | <p>La planification de release est réalisée avec l'input supplémentaire du programme/portfolio auquel le projet appartient. En dehors de ce point, il n'y a pas d'autre changement dans ce processus.</p> <p>Input supplémentaire : Program/Portfolio Product Owner Les rôles du Program/Portfolio Product Owner sont décrits dans les sections 3.7.4.1 et 3.7.4.2.</p> <p>Input supplémentaire : Program/Portfolio Scrum Master Les rôles du Program/Portfolio Scrum Master sont décrits dans les sections 3.7.4.3 et 3.7.4.4.</p> <p>Input supplémentaire : Program/Portfolio Backlog priorisé Le Program/Portfolio Backlog priorisé est décrit dans la section 14.6.1.2. Le Program/Portfolio Backlog priorisé peut contenir des dates clés de coordination et/ou des délais pour certaines exigences que le projet doit respecter.</p> |

Tableau 14-1: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase d'initiation

14.1.2 Planification et estimation

L'utilisation de Scrum au niveau d'un programme/portfolio n'a aucun impact sur la phase de planification et d'estimation des projets respectifs dans le programme/portfolio.

14.1.3 Implémentation

Les données supplémentaires provenant du niveau du programme/portfolio qui doivent être prises en compte pour la phase d'implémentation sont les suivantes:

| | |
|--|---|
| 10.1 Création des produits livrables | La création des produits livrables n'est pas affectée par l'utilisation de Scrum au niveau du programme ou du portfolio. |
| 10.2 Mêlée quotidienne | La tenue de la mêlée quotidienne n'est pas affectée par l'utilisation de Scrum au niveau du programme ou du portfolio. |
| 10.3 Raffinage du Product Backlog priorisé | <p>Lors du raffinage du Product Backlog priorisé, les exigences nouvelles ou modifiées au niveau du programme ou du portfolio doivent être priorisées et incorporées dans le Product Backlog de manière appropriée.</p> <p>Input supplémentaire : Program/Portfolio Product Owner Les rôles du Program/Portfolio Product Owner sont décrits dans les chapitres 3.7.4.1 et 3.7.4.2. Dans ce processus, le Program/Portfolio Product Owner communique au projet toutes les exigences mises à jour depuis le niveau du programme ou du portfolio.</p> <p>Input supplémentaire : Program/Portfolio Backlog priorisé Le Program/Portfolio Backlog priorisé est décrit dans la section 14.6.1.2. Tout changement dans le Program/Portfolio Backlog priorisé doit être incorporé dans le Product Backlog priorisé du projet.</p> |

Tableau 14-2: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase de Implémentation

14.1.4 Révision et rétrospective

Les représentants du programme ou du portfolio peuvent fournir un feedback lors des révisions ou rétrospectives de sprint. En dehors de cela, il n'y a pas d'autre changement à la phase de révision et de rétrospective d'un projet.

| | |
|---|--|
| 11.1 Présentation et validation du sprint | <p>Les représentants du programme/portfolio peuvent fournir leur feedback. En dehors de ce point, il n'y a pas d'autre changement dans ce processus.</p> <p>Input supplémentaire : Program/Portfolio Product Owner Les rôles du Program/Portfolio Product Owner sont décrits dans les sections 3.7.4.1 et 3.7.4.2.</p> <p>Input supplémentaire : Parties prenantes de l'entreprise du programme/portfolio Les parties prenantes de l'entreprise du programme/portfolio sont décrites dans la section 14.3.3.5.</p> |
| 11.2 Rétrospective de sprint | Le processus de <i>rétrospective de sprint</i> n'est pas affecté par l'utilisation de Scrum au niveau du programme ou du portfolio. |

Tableau 14-3: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase de révision et de rétrospective

14.1.5 Release

L'application de Scrum au niveau du programme/portfolio peut avoir un impact sur les releases de projet individuelles, car il peut y avoir des dépendances entre les différentes releases de projet. Par exemple, si les produits livrables de deux projets, A et B, doivent idéalement être publiés ensemble, mais que les produits livrables du projet A sont retardés, cela peut avoir un impact sur la release des produits livrables du projet B, même si les produits livrables du projet B sont terminés à temps.

| | |
|---------------------------------------|---|
| 12.1 Livraison des produits livrables | <p>Les produits livrables d'un programme ou d'un portfolio sont créés de la même manière que pour les projets individuels. Cependant, il peut y avoir certaines dépendances avec des produits livrables d'autres projets qui doivent être coordonnées par le Program/Portfolio Product Owner ou le Program/Portfolio Scrum Master.</p> <p>Input mis à jour : Program/Portfolio Product Owner Description dans les sections 3.7.4.1 et 3.7.4.2.</p> <p>Input mis à jour : Program/Portfolio Scrum Master Description dans les sections 3.7.4.3 et 3.7.4.4.</p> |
| 12.2 Rétrospective de release | Le processus de <i>rétrospective de release</i> n'est pas affecté par l'utilisation de Scrum au niveau du programme ou du portfolio. |

Tableau 14-4: Impact d'un programme ou d'un portefeuille sur les processus Scrum fondamentaux - Phase de release

14.2 Processus supplémentaires pour mettre Scrum à l'échelle de l'entreprise (programme/portfolio)

Tous les projets Scrum qui font partie d'un programme ou d'un portfolio plus large peuvent appliquer les processus Scrum tels que décrits dans les chapitres 8 à 13, avec les impacts supplémentaires décrits dans la section 14.1.

Toutefois, lors de la mise à l'échelle de Scrum pour l'entreprise, certains processus supplémentaires peuvent être nécessaires pour gérer la complexité accrue de plusieurs centaines ou milliers de personnes travaillant sur les projets associés et les exigences de coordination supplémentaires au niveau du programme ou du portfolio. Tous ces processus ne sont pas nécessairement séquentiels et peuvent être appliqués en parallèle et de manière itérative selon les besoins de l'entreprise.

La figure 14-1 donne un aperçu des processus supplémentaires requis pour la mise à l'échelle de Scrum pour l'entreprise, qui sont les suivants :

14.3 Création/mise à jour d'équipes pour les programmes ou portfolios—dans ce processus, des rôles supplémentaires sont créés ou identifiés pour gérer les programmes et les portfolios. Ces rôles comprennent le Program Product Owner, le Portfolio Product Owner, le Program Scrum Master, le Portfolio Scrum Master, les parties prenantes de l'entreprise, et les services de soutien.

14.4 Création/mise à jour des éléments de programme ou de portfolio—dans ce processus, les Program ou Portfolio Product Owners, les Scrum Masters, et les parties prenantes de l'entreprise identifient et créent les éléments communs et les ressources nécessaires au programme ou au portfolio. Les critères « Done » (Terminé) minimaux sont établis et toutes les parties prenantes de l'entreprise pertinentes sont identifiées. Les dépendances entre les projets sont abordées, les obstacles communs sont examinés et les meilleures pratiques sont partagées. Des recommandations pour les améliorations du Scrum Guidance Body sont parfois formulées.

14.5 Révision et mise à jour du Scrum Guidance Body—dans ce processus, les recommandations du Scrum Guidance Body sont régulièrement passées en revue par les membres du Scrum Guidance Body et sont mises à jour le cas échéant. Les changements des membres du Scrum Guidance Body sont aussi abordés durant ce processus. L'objectif principal de ce processus est de surveiller et d'améliorer en permanence la productivité des projets, programmes et des programmes Scrum, et des portfolios au sein de l'organisation.

14.6 Création/raffinage du Program ou Portfolio Backlog priorisé—dans ce processus, le Program ou Portfolio Backlog est d'abord créé sur la base des exigences du programme ou du portfolio. Sur une base continue, le Program ou Portfolio Backlog priorisé est révisé pour ajouter ou mettre à jour les exigences, les risques et les priorités.

14.7 Création/mise à jour de releases de programme ou portfolio—dans ce processus, les releases du programme ou du portfolio sont planifiées, en tenant compte des éventuelles dépendances entre les versions. La planification des releases de programme ou de portfolio aura un impact sur la planification des releases au niveau du projet. Le calendrier de release du programme ou du portfolio est créé et doit être révisé régulièrement en fonction de l'avancement des produits livrables du projet, des exigences nouvelles ou modifiées ou de leurs priorités, et d'autres facteurs.

14.8 Rétrospective de release de programme ou de portfolio—dans ce processus, le the Program ou Portfolio Product Owner et les parties prenantes de l'entreprise se réunissent pour faire une rétrospective d'une release de programme ou de portfolio, ainsi que pour discuter et internaliser les enseignements appris. Généralement, les enseignements mènent à l'implémentation d'améliorations convenues réalisables (agreed actionable improvements) lors des futurs projets. Parfois, des améliorations du Scrum Guidance Body peuvent être recommandées.

| 14.3 Création/mise à jour d'équipes pour les programmes ou portfolios | Création des éléments de programme ou de portfolio | 14.5 Révision et mise à jour du Scrum Guidance Body |
|--|---|--|
| <p>INPUTS</p> <ol style="list-style-type: none"> 1. Vision et mission de l'entreprise* 2. Direction générale* 3. Organisation matricielle des ressources (Organizational Resource Matrix) 4. Consultants <p>OUTILS</p> <ol style="list-style-type: none"> 1. Stratégie de ressources humaines de l'entreprise* 2. Analyse des parties prenantes* <p>OUTPUTS</p> <ol style="list-style-type: none"> 1. Portfolio Product Owner* 2. Program Product Owner* 3. Portfolio Scrum Master* 4. Program Scrum Master* 5. Parties prenantes de l'entreprise * 6. Services de soutien* | <p>INPUTS</p> <ol style="list-style-type: none"> 1. Vision et mission de l'entreprise* 2. Portfolio Product Owner* 3. Portfolio Scrum Master* 4. Program Product Owner* 5. Program Scrum Master* 6. Organisation matricielle des ressources (Organizational Resource Matrix) 7. Recommandations du Scrum Guidance Body 8. Parties prenantes de l'entreprise <p>OUTILS</p> <ol style="list-style-type: none"> 1. Plan(s) de communication* 2. Stratégie de ressources humaines de l'entreprise* 3. Analyse des parties prenantes* 4. Réunion Scrum of Scrums* 5. Réunion Scrum of Scrums of Scrums (SoSoS) 6. Techniques de communication 7. Outil de projet Scrum <p>OUTPUTS</p> <ol style="list-style-type: none"> 1. Critères « Done » (Terminé) minimaux * 2. Ressources partagées* 3. Parties prenantes de l'entreprise identifiées* 4. Mise à jour des Impediment Logs* 5. Dépendances mises à jour* 6. Plan de collaboration des Product Owners* 7. Plan de collaboration des Scrum Masters* 8. Améliorations recommandées du Scrum Guidance Body | <p>INPUTS</p> <ol style="list-style-type: none"> 5. Règlements* 6. Améliorations recommandées du Scrum Guidance Body* 7. Membres du Scrum Guidance Body* <p>OUTILS</p> <ol style="list-style-type: none"> 1. Critères de sélection des membres* 2. Réunions du Scrum Guidance Body* 3. Rapports de performance 4. Benchmarking <p>OUTPUTS</p> <ol style="list-style-type: none"> 1. Mise à jour des recommandations du Scrum Guidance Body* 2. Actionable Escalations (actions exploitables) 3. Mise à jour des membres du Scrum Guidance Body 4. Mises à jour rejetées des recommandations du Scrum Guidance Body |

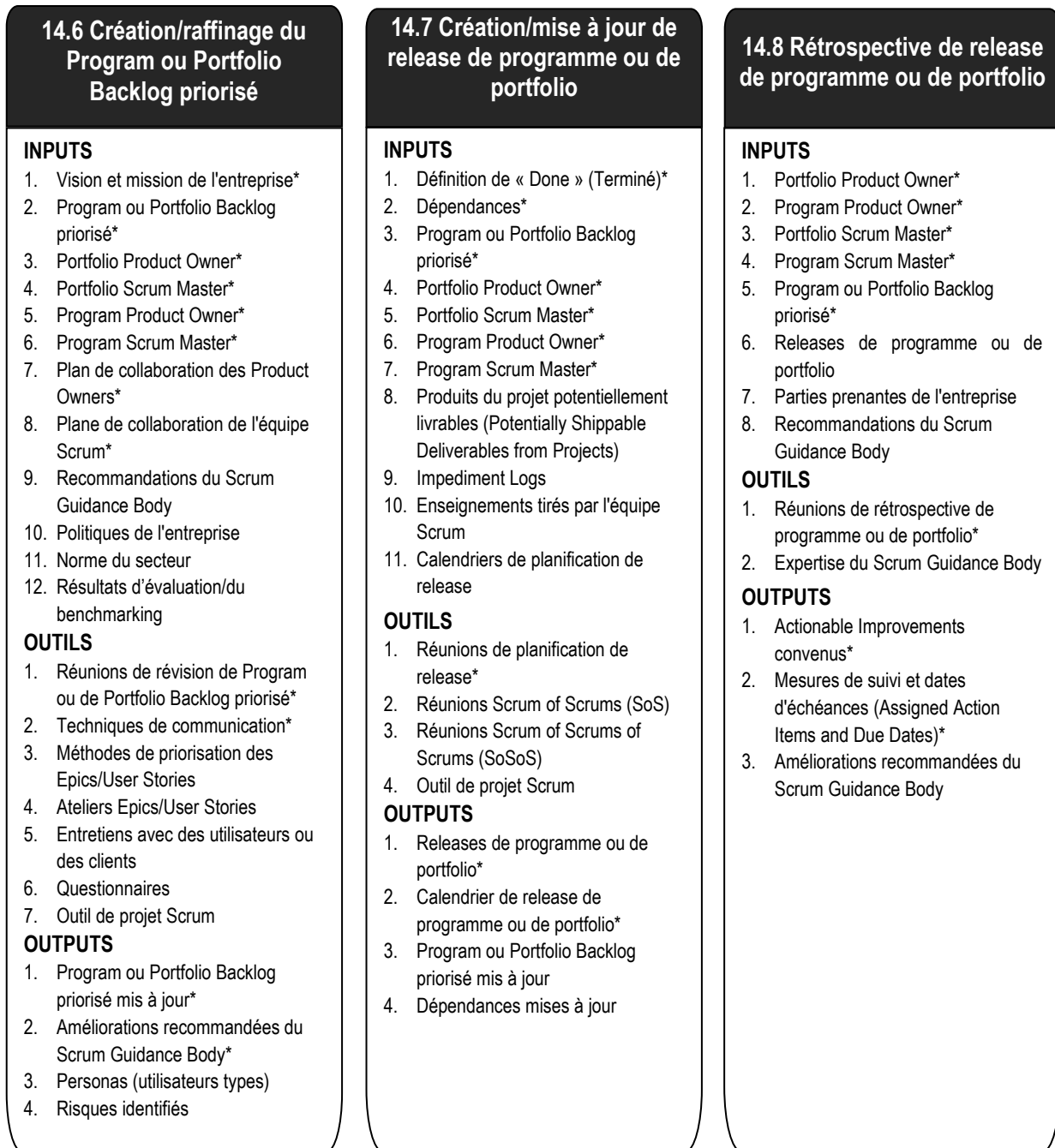


Figure 14-1: La mise à l'échelle de Scrum pour l'entreprise

14.3 Création/mise à jour d'équipes pour les programmes ou portefeuilles

Des équipes de programme ou de portefeuille doivent être créées et/ou identifiées avant que les processus Scrum puissent être appliqués dans un environnement d'entreprise. Parmi les rôles importants, citons le Program Product Owner, le Portfolio Product Owner, le Program Scrum Master, le Portfolio Scrum Master, les parties prenantes de l'entreprise, et les services de soutien.

Il est important de noter que toutes ces personnes ne doivent pas nécessairement être identifiées et assignées au début du programme ou du portefeuille. Certaines personnes clés peuvent être identifiées dès le début, tandis que d'autres peuvent être assignées au fil du temps en fonction des exigences spécifiques du programme ou du portefeuille.

Les résultats du processus de *création/mise à jour d'équipes pour les programmes ou portefeuilles* deviennent des inputs dans le processus après la création de l'équipe initiale. Par exemple, une fois que le Program Product Owner ou le Program Scrum Master sont identifiés, ces personnes seront impliquées dans tous les futurs changements ou mises à jour de l'équipe du programme.

La figure 14-2 montre tous les inputs, les outils et les outputs du processus de *création/mise à jour d'équipes pour les programmes ou portefeuilles*.

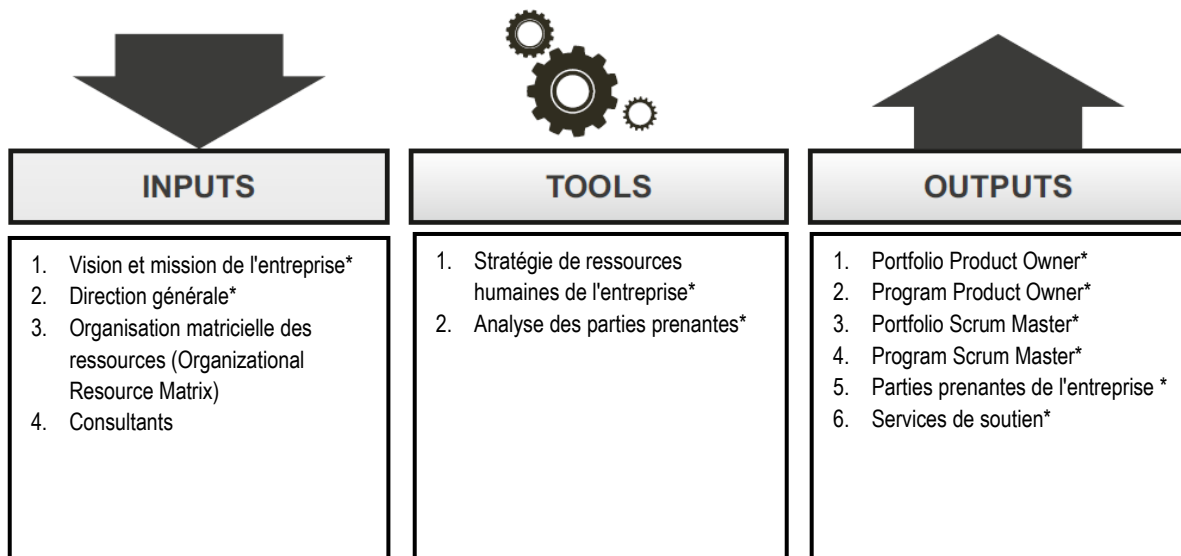


Figure 14-2: Create/Update Program or Portfolio Teams—Inputs, Tools, and Outputs

14.3.1 Inputs

14.3.1.1 Vision et mission de l'entreprise*

La vision et la mission de l'entreprise sont des éléments importants pour tout projet, mais elles sont encore plus importantes pour la planification des programmes et surtout des portefeuilles. Les programmes et les portefeuilles doivent être guidés par la mission et la vision globales de l'entreprise, car cela garantit un effort uniforme dans toute l'organisation. Comprendre la vision de l'entreprise aide les équipes chargées du portefeuille ou du programme à rester concentrées sur les objectifs de l'organisation et le potentiel futur de l'entreprise. Les Program or Portfolio Product Owners peuvent s'inspirer des conseils et directives de la vision de l'entreprise afin de créer l'énoncé de la vision du projet.

La mission de l'entreprise offre un cadre pour la formulation des stratégies de l'entreprise et guide le processus de décision global. L'énoncé de la vision du projet doit être formulé de telle sorte que sa réalisation aide l'organisation à remplir sa mission.

14.3.1.2 Direction générale*

La direction générale comprend les représentants seniors de l'organisation interne de l'entreprise qui bénéficient des produits livrables du programme ou du portefeuille, ou qui en sont responsables. Les cadres supérieurs de la société peuvent comprendre le directeur général, le directeur de la technologie, le directeur financier, les vice-présidents, les directeurs et d'autres cadres supérieurs de différentes divisions de la société organization.

14.3.1.3 Organisation matricielle des ressources (Organizational Resource Matrix)

La matrice des ressources organisationnelles au niveau d'un programme ou d'un portefeuille devrait inclure les employés de l'organisation qui ont les compétences et la disponibilité nécessaires pour assumer des rôles supérieurs liés aux projets Scrum. Pour plus d'informations sur la matrice des ressources de l'organisation, voir la section 8.2.1.5.

14.3.1.4 Consultants

Si toutes les compétences requises pour mettre en place et gérer des programmes ou des portefeuilles Scrum ou des portefeuilles au sein de l'organisation ne sont pas disponibles en interne, il est possible de faire appel à des consultants externes pour guider la direction générale dans la mise en place des équipes de programme ou de portefeuille nécessaires.

14.3.2 Outils

14.3.2.1 Stratégie de ressources humaines de l'entreprise*

Le plan de ressources humaines de l'entreprise fournit des informations générales sur le moment où le personnel sera disponible pour divers projets, programmes et portfolios. Le plan fournit également des informations sur les compétences et les capacités disponibles au sein de l'entreprise, ainsi que sur les plans de recrutement du personnel nécessaire aux futurs efforts à fournir.

14.3.2.2 Analyse des parties prenantes

Diverses techniques d'analyse des parties prenantes peuvent être utilisées pour identifier et analyser les parties prenantes de l'entreprise et toutes les autres parties prenantes concernées au niveau du programme et du portfolio. Les techniques d'analyse des parties prenantes peuvent également être utilisées pour évaluer les intérêts, la participation et l'impact potentiel de chaque partie prenante identifiée sur le programme ou le portfolio. L'analyse des parties prenantes est également utile pour comprendre les exigences de communication et d'implication nécessaires au bénéfice du programme ou du portfolio.

14.3.3 Outputs

14.3.3.1 Portfolio Product Owner*

Description dans les sections 3.7.4.2.

14.3.3.2 Program Product Owner*

Description dans la section 3.7.4.1.

14.3.3.3 Portfolio Scrum Master*

Description dans les sections 3.7.4.4.

14.3.3.4 Program Scrum Master*

Description dans la section 3.7.4.3.

14.3.3.5 Parties prenantes de l'entreprise *

Dans ce processus, les parties prenantes de l'entreprise qui joueront des rôles clés dans le programme ou le portfolio sont identifiées, y compris les clients, les utilisateurs et les sponsors du programme ou du portfolio. Les parties prenantes de l'entreprise influencent le programme ou le portfolio lui-même, ainsi que les projets associés (au sein du programme ou du portfolio) tout au long de leur développement (au sein du programme ou du portfolio) tout au long du développement de chaque projet. Les parties prenantes du programme ou du portfolio peuvent également aider à définir la vision du projet/programme/portfolio et fournir des conseils concernant la valeur commerciale.

Les parties prenantes de l'entreprise du programme interagissent avec les parties prenantes de l'entreprise du portfolio pour garantir l'alignement du programme sur les buts et les objectifs du portfolio. Les parties prenantes de l'entreprise du programme et/ou du portfolio sont également impliquées dans l'identification des parties prenantes de l'entreprise pour les projets individuels et garantissent que la vision, les objectifs, les résultats et les releases de ces projets associés au sein du programme/portfolio s'alignent sur ceux du programme/portfolio.

Au niveau du portfolio, les parties prenantes de l'entreprise peuvent être les membres du conseil d'administration d'une entreprise ou d'une organisation gouvernementale. Au niveau du programme, les parties prenantes de l'entreprise peuvent être des cadres supérieurs et le(s) sponsor(s) du programme et des projets associés.

14.3.3.6 Services de soutien*

Au niveau d'un programme ou d'un portfolio, les services de soutien devraient inclure des personnes ou des groupes responsables de la gestion de la formation, de la logistique, du marketing, des finances, de l'infrastructure, de l'architecture et d'autres services de soutien nécessaires au bon fonctionnement du programme ou du portfolio. Certaines de ces personnes peuvent également travailler à plein temps pour le programme ou le portfolio. Pour plus d'informations sur les services de soutien, voir la section 3.3.2.

14.4 Création/mise à jour des éléments de programme ou de portfolio

Dans ce processus, les Program ou Portfolio Product Owners, les Scrum Masters, et les parties prenantes de l'entreprise identifient et créent les éléments communs et les ressources nécessaires au programme ou au portfolio. Les critères « Done » (Terminé) minimaux sont établis et toutes les parties prenantes de l'entreprise pertinentes sont identifiées. Les dépendances entre les projets sont abordées, les obstacles communs sont examinés et les meilleures pratiques sont partagées. Des recommandations pour les améliorations du Scrum Guidance Body sont parfois formulées.

Il est important de noter que tous les éléments du programme ou du portfolio ne doivent pas nécessairement être créés au départ. Il s'agit généralement d'un processus itératif, car certains éléments clés du programme ou du portfolio sont créés dès le début, et d'autres peuvent être créés ou mis à jour ultérieurement, à mesure que de nouvelles informations sont disponibles.

Les outputs du processus de *création des éléments de programme ou de portfolio* deviennent des inputs du processus après leur création initiale. Par exemple, une fois que les critères « Done » (Terminé) minimaux sont identifiés, ils deviendront des inputs lorsque les éléments du programme ou du portfolio seront mis à jour.

La figure 14-3 montre tous les inputs, les outils et les outputs du processus de *création/mise à jour des éléments de programme ou de portfolio*.

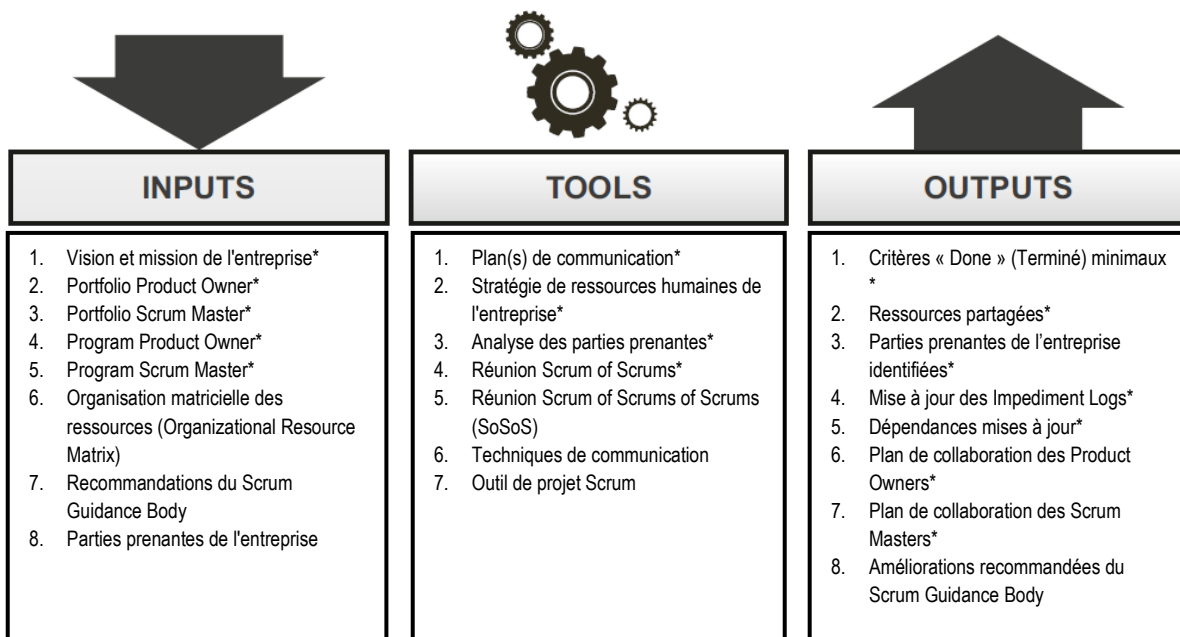


Figure 14-3 : Création/mise à jour des éléments de programme ou de portfolio—inputs, outils et outputs

Note : Les astérisques (*) indiquent un input, un outil ou un output « obligatoire » pour le processus correspondant.

14.4.1 Inputs

14.4.1.1 Vision et mission de l'entreprise*

La vision et la mission de l'entreprise sont des éléments importants pour tout projet, mais encore plus pour les programmes et les portfolios. Les programmes et les portfolios doivent être guidés par la mission et la vision globales de l'entreprise, car cela garantit un effort uniforme dans toute l'organisation. Pour plus d'informations sur la vision et la mission de l'entreprise, voir la section 14.3.1.1.

14.4.1.2 Portfolio Product Owner*

Description dans la section 3.7.4.2.

14.4.1.3 Portfolio Scrum Master*

Description dans la section 3.7.4.4.

14.4.1.4 Program Product Owner*

Description dans la section 3.7.4.1.

14.4.1.5 Program Scrum Master*

Description dans la section 3.7.4.3.

14.4.1.6 Organisation matricielle des ressources (Organizational Resource Matrix)

Description dans les sections 8.2.1.5.

14.4.1.7 Recommandations du Scrum Guidance Body

Les recommandations du Scrum Guidance Body sont particulièrement importantes au niveau des programmes et des portfolios, car des orientations appropriées sont nécessaires pour un nombre potentiellement important de projets connexes. Pour plus d'informations sur les recommandations du Scrum Guidance Body, voir la section 8.1.1.7.

14.4.1.8 **Parties prenantes de l'entreprise**

Description dans les sections 3.3.2 et 14.3.3.5.

14.4.2 **Outils**

14.4.2.1 **Plan(s) de communication***

Le(s) plan(s) de communication définit/issent la manière dont l'information doit être diffusée aux parties prenantes de l'entreprise et à travers les programmes, le portfolio et l'organisation dans son ensemble. Il/ils doit/doivent également définir comment et quand communiquer et quel mode de communication doit être utilisé. Les rôles du portfolio fournissent une orientation et un input au plan de communication pour les programmes associés au sein du portfolio. De même, les rôles du programme fournissent des conseils et des inputs au plan de communication pour les projets au sein du programme. Pour plus d'informations sur le plan de communication, voir la section 12.1.3.4.

14.4.2.2 **Stratégie de ressources humaines de l'entreprise***

Description dans la section 14.3.2.1.

14.4.2.3 **Analyse des parties prenantes**

Description dans la section 14.3.2.2.

14.4.2.4 **Réunion Scrum of Scrums (SoS)***

L'objectif de la réunion Scrum of Scrums (SoS) est similaire à son utilisation dans les grands projets. Au niveau du programme, les représentants de chaque projet sous-jacent du programme se réunissent à intervalles réguliers pour des réunions de type « Scrum of Scrums » (SoS). Pour plus d'informations sur les réunions SoS, voir la section 13.3.5.

14.4.2.5 **Réunion Scrum of Scrums of Scrums (SoSoS)**

Au niveau du programme et surtout du portfolio, il est logique d'avoir une autre session de réunions. Les représentants de programmes et projets pertinents ou interdépendants dans le programme ou le portfolio se réunissent à intervalles réguliers, ou selon les besoins. Des représentants de chacune des réunions Scrum of Scrums seront présents.

Ce niveau supplémentaire de réunions est appelé « Scrum of Scrums of Scrums » (SoSoS). La figure 14-4 illustre le concept des réunions « Scrum of Scrums » (SoS) et des réunions « Scrum of Scrums of Scrums (SoSoS) ».

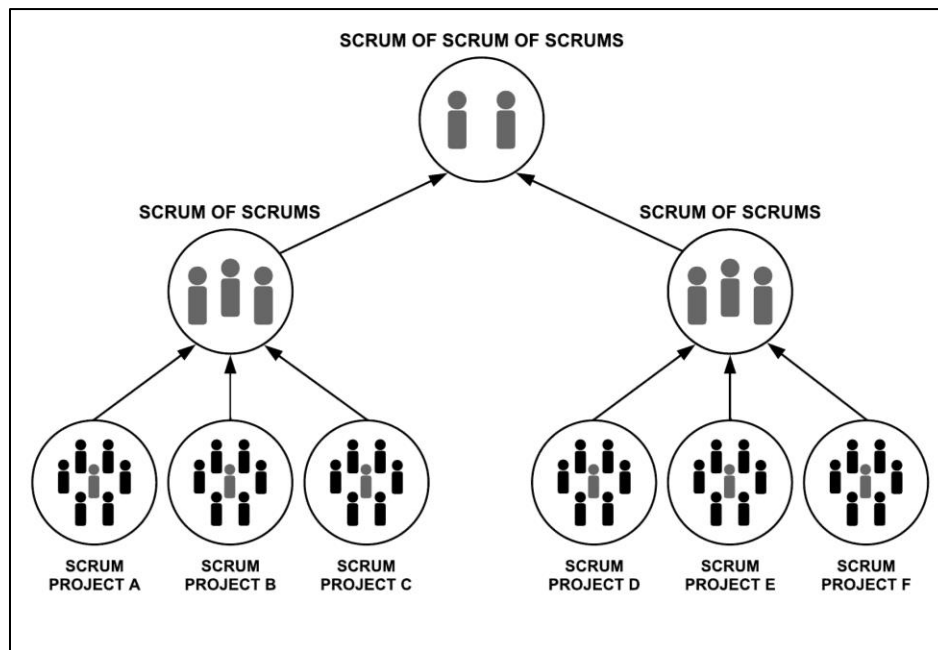


Figure 14-4 : Réunion Scrum of Scrums of Scrums (SoSoS)

Dans cet exemple, il y a six projets Scrum en cours simultanément. Les projets Scrum A, B et C font partie d'un programme tandis que les projets Scrum D, E et F font partie d'un autre programme. Une réunion Scrum of Scrums est organisée pour coordonner les interdépendances au sein de chacun des deux programmes. Une réunion Scrum of Scrums of Scrums peut alors être organisée pour coordonner et gérer les dépendances entre les deux programmes du portfolio.

14.4.2.6 Techniques de communication

Au niveau du programme ou du portfolio, les méthodes et techniques de communication doivent être adaptées à un plus grand nombre de participants et au fait que tous ne se trouvent pas sur le même lieu de travail. D'autres considérations doivent être prises en compte pour les types de communication « push » et « pull », par exemple les tableaux de bord ou les rapports publiés en ligne pour les parties prenantes de l'entreprise à consulter à la demande (pull) ou envoyés directement (push) à intervalles réguliers. La communication au sein de groupes plus importants peut utiliser davantage d'outils, tels que les réunions en ligne, le courrier électronique, la messagerie instantanée et les tableaux de messages, lorsque la communication en face à face n'est pas possible. Utilisation d'un outil de projet Scrum peut également faciliter la communication au niveau d'un programme ou d'un portfolio. Pour plus d'informations sur les techniques de communication, voir la section 10.3.2.2.

14.4.2.7 Outil de projet Scrum

Au niveau d'un programme ou d'un portfolio, l'outil de projet Scrum offre la possibilité d'évoluer afin de gérer les rôles supplémentaires, la coordination, les rapports, les communications et les autres exigences pertinentes. Pour plus d'informations sur l'outil de projet Scrum, voir les sections 2.5.3.1 et 13.3.8.

14.4.3 Outputs

14.4.3.1 Critères « Done » (Terminé) minimaux*

Les critères « Done » (Terminé) minimaux définis au niveau d'un portfolio s'appliquent à tous les programmes et projets sous-jacents. De même, les critères « Done » (Terminé) minimaux au niveau du programme s'appliquent à tous les projets sous-jacents. L'ensemble en cascade des critères « Done » (Terminé) garantit que tous les critères achevés sous-jacents répondent aux exigences minimales établies aux niveaux supérieurs. Le Scrum Guidance Body peut être impliqué dans la définition des critères « Done » (Terminé) minimaux au niveau du portfolio. Pour plus d'informations sur les critères « Done » (Terminé) minimaux, voir la section 5.4.4.

14.4.3.2 Ressources partagées*

Description dans la section 13.2.4.

14.4.3.3 Parties prenantes de l'entreprise identifiées*

Les parties prenantes de l'entreprise au niveau du portfolio ou du programme constituent un input à ce processus. D'autres parties prenantes de l'entreprise sont identifiées au cours de ce processus. Pour plus d'informations sur les parties prenantes de l'entreprise pertinentes au niveau du projet, voir la section 8.2.3.2.

14.4.3.4 Mise à jour des Impediment Logs*

Les obstacles rencontrés par les projets individuels peuvent être pertinents pour d'autres projets au sein du programme ou du portfolio. Par conséquent, les Impediment Logs au niveau du projet peuvent devoir être partagés avec les autres projets et/ou programmes. À la suite des réunions Scrum of Scrums (SoS) ou de la Scrum of Scrums of Scrums (SoSoS), il peut être nécessaire de mettre à jour les Impediment Logs au niveau du projet. Il pourrait également y avoir des Impediment Logs au niveau du programme ou du portfolio. Pour plus d'informations sur les Impediment Logs, voir la section 10.1.1.4.

14.4.3.5 Dépendances mises à jour*

Il peut y avoir des dépendances entre des projets liés entre eux et même entre des programmes au sein de l'entreprise qui doivent être identifiées. Par conséquent, une coordination entre les projets associés doit être assurée pour gérer ces dépendances.

Exemples de dépendances :

- Dates de release partagées pour les projets interdépendants
- Dépendances entre les releases
- Dépendances relatives à des fonctionnalités interdépendantes

Suite à la coordination des éléments du programme ou du portfolio, il peut être nécessaire de mettre à jour les dépendances connues avec de nouvelles dépendances ou des modifications des dépendances existantes. Par exemple, il peut y avoir des dépendances entre les projets au sein d'un programme ou d'un portfolio. Si l'on considère deux projets A et B dans un programme, il se peut que ces deux projets doivent avoir la même date de release, ou il se peut que le projet A ne puisse être lancé qu'après la release du projet B. Dans les deux cas, si le projet B est retardé, le projet A le sera aussi, même si ses produits livrables sont prêts à temps. Pour plus d'informations sur les dépendances au niveau du projet, voir les sections 8.5.2.6 et 9.4.2.3.

14.4.3.6 Plan de collaboration des Product Owners*

Description dans la section 13.2.2.

14.4.3.7 Plan de collaboration des Scrum Masters*

Description dans la section 13.2.3.

14.4.3.8 Améliorations recommandées du Scrum Guidance Body

À la suite du processus de *création/mise à jour des éléments de programme ou de portfolio*, des suggestions ou des commentaires peuvent être fournis pour des améliorations potentielles de la documentation du Scrum Guidance Body. Ces améliorations recommandées seront discutées et acceptées ou rejetées par le Scrum Guidance Body (voir la section 14.5, *Révision et mise à jour du Scrum Guidance Body*). Si les suggestions sont acceptées, celles-ci seront ajoutées en tant que mises à jour dans la documentation du Scrum Guidance Body.

14.5 Révision et mise à jour du Scrum Guidance Body

Dans ce processus, les recommandations du Scrum Guidance Body sont régulièrement passées en revue par les membres du Scrum Guidance Body et mises à jour le cas échéant. Les changements des membres du Scrum Guidance Body sont aussi abordés durant ce processus. L'objectif principal est de surveiller constamment et de travailler à l'amélioration de la productivité des projets, des programmes et des portfolios Scrum au sein de l'organisation.

Il est important de noter que le processus de *révision et de mise à jour du Scrum Guidance Body* est généralement un processus itératif, puisque les projets Scrum s'engagent continuellement dans des rétrospectives où des opportunités d'amélioration sont régulièrement identifiées et transmises aux niveaux du programme et du portfolio. Au niveau du programme ou du portfolio, le Scrum Guidance Body examine les inputs des rétrospectives au niveau du programme et du portfolio, identifie les possibilités d'amélioration et contribue à la diffusion des bonnes pratiques dans l'entreprise.

La figure 14-5 montre tous les inputs, outils et outputs du processus de *révision et de mise à jour du Scrum Guidance Body*.

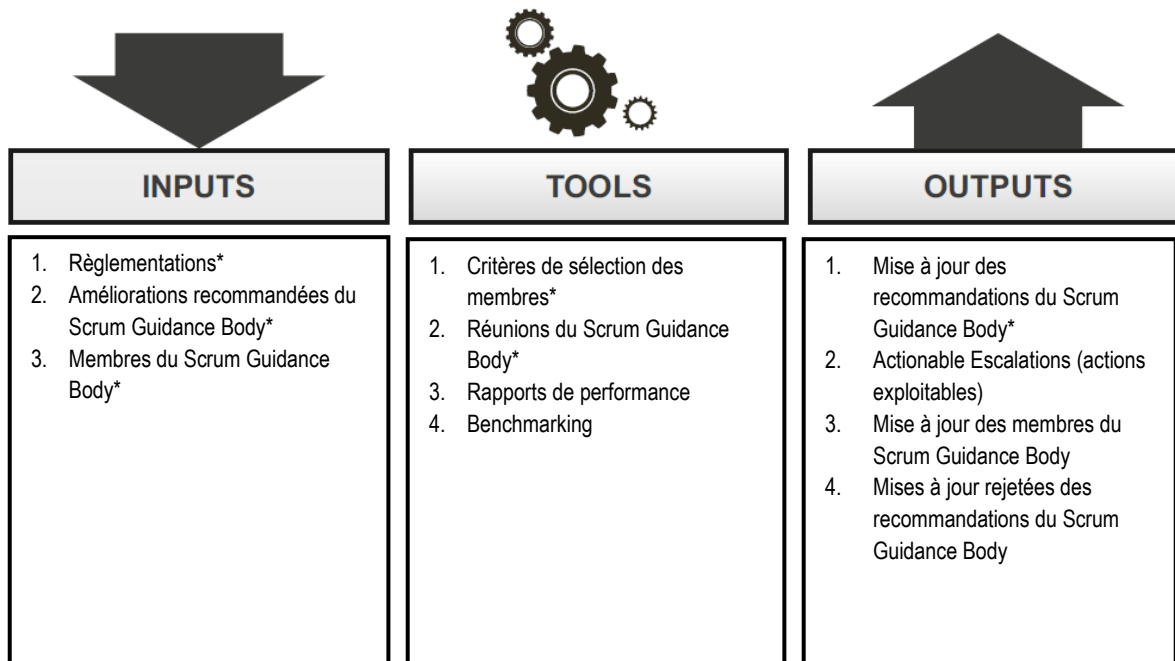


Figure 14-5 : Révision et mise à jour du Scrum Guidance Body—inputs, outils et outputs

Note : Les astérisques (*) indiquent un input, un outil ou un output « obligatoire » pour le processus correspondant.

14.5.1 Inputs

14.5.1.1 Règlements*

Les réglementations comprennent toutes les réglementations fédérales, étatiques, locales ou industrielles auxquelles le programme ou le portfolio doit se conformer. Les User Stories créées pour répondre aux réglementations gouvernementales dans un délai donné sont incluses dans le Portfolio ou le Program Product Backlog. Parfois, les recommandations du Scrum Guidance Body doivent être mises à jour afin de prendre en compte de nouvelles réglementations.

14.5.1.2 Améliorations recommandées du Scrum Guidance Body*

À la suite des rétrospectives Scrum et d'autres processus, des suggestions et des retours d'information visant à réviser ou à améliorer le Scrum Guidance Body, des lignes directrices, des modèles et autres documents peuvent être formulés. Si le Scrum Guidance Body est en accord avec les suggestions ou le retour d'information, les changements pertinents seront intégrés en tant que mises à jour du matériel du Scrum Guidance Body et fournis sous forme de recommandations aux équipes de projet, de programme et de portfolio.

14.5.1.3 Membres du Scrum Guidance Body

Les membres du Scrum Guidance Body (SGB) peuvent inclure des experts Scrum, des coachs Scrum, des consultants externes, des Scrum Masters désignés, des Product Owners et des membres d'équipes (à tous les niveaux). Toutefois, il convient de limiter le nombre de membres que le SGB peut avoir afin de garantir que celui-ci reste pertinent et ne devienne pas prescriptif par nature.

14.5.2 Outils

14.5.2.1 Critères de sélection des membres*

Les critères de sélection des membres sont déterminés pour définir les membres du Scrum Guidance Body, leurs rôles et responsabilités, le nombre de membres ainsi que leur expertise et les compétences requises. Chaque organisation peut avoir ses propres critères de sélection pour les membres du Scrum Guidance Body ; cependant, il est recommandé que chaque membre ait une expertise de Scrum et puisse contribuer efficacement au Scrum Guidance Body.

14.5.2.2 Réunions du Scrum Guidance Body*

Le Scrum Guidance Body se réunit régulièrement afin d'aborder le besoin potentiel pour la mise à jour des recommandations du Scrum Guidance Body (p. ex. améliorations recommandées à partir des rétrospectives et autres processus, mise à jour de la réglementation, etc.). La fréquence de ces réunions est décidée par le Scrum Guidance Body en fonction des besoins spécifiques de l'entreprise.

14.5.2.3 Rapports de performance

Des rapports sur la performance des projets, des programmes et des portefeuilles Scrum peuvent être disponibles. Ces rapports de performance peuvent inclure des informations relatives à la vitesse de l'équipe, aux fonctionnalités livrées, à l'état d'avancement, etc. Ces informations peuvent être prises en compte par le Scrum Guidance Body lorsqu'il détermine les possibilités d'amélioration.

14.5.2.4 Benchmarking

Le benchmarking consiste à comparer les procédures commerciales et les mesures de performance d'une organisation avec celles d'entreprises dominantes dans le même ou dans un autre secteur. Une entreprise doit régulièrement comparer ses propres pratiques à celles d'organisations qui ont réussi (afin de ne pas se laisser distancer par la concurrence) et aux normes et pratiques actuelles et futures du secteur.

14.5.3 Outputs

14.5.3.1 Mise à jour des recommandations du Scrum Guidance Body*

Après avoir examiné et pris en compte le Scrum Guidance Body, les suggestions d'amélioration, les rapports de performance et les données d'analyse comparative, il peut s'avérer nécessaire de modifier la documentation existante. Toute modification approuvée entraînera une mise à jour du matériel du Scrum Guidance Body et sera fournie sous forme de recommandations aux projets, programmes et portefeuilles Scrum actuels ou futurs.

14.5.3.2 Actionable Escalations (actions exploitables)

Dans certains cas, le Scrum Guidance Body peut décider de faire en sorte que certaines politiques d'entreprise empêchent l'équipe de profiter au mieux des avantages associés à l'application de l'approche Scrum. Dans ce cas, un processus d'« escalade » doit être déclenché afin d'obtenir l'approbation d'un changement de politique ou autre.

14.5.3.3 Mise à jour des membres du Scrum Guidance Body

À la suite de l'évaluation des membres du Scrum Guidance Body, de nouveaux membres peuvent être ajoutés et des membres existants peuvent quitter le Scrum Guidance Body.

14.5.3.4 Mises à jour rejetées des recommandations du Scrum Guidance Body

Les améliorations recommandées du Scrum Guidance Body ne sont pas toujours acceptées. Si les améliorations recommandées ne sont pas acceptées par les membres du Scrum Guidance Body, un feedback expliquant la ou les raisons du rejet est fournies aux parties concernées.

14.6 Création/raffinage du Program ou Portfolio Backlog priorisé

Dans ce processus, le Program ou Portfolio Backlog est d'abord créé sur la base des exigences du programme ou du portfolio. De façon continue, le Program ou Portfolio Backlog priorisé est mis à jour et maintenu avec les exigences, les risques et les priorités nouveaux ou mis à jour.

Tous les résultats du processus de *création/raffinage du Program ou Portfolio Backlog priorisé* deviennent des inputs du processus après leur création initiale. Par exemple, le Program ou Portfolio Backlog priorisé est créé pour la première fois au cours de ce processus, mais il devient un input obligatoire pour les raffinages/mises à jour ultérieurs.

La figure 14-6 montre tous les inputs, outils et outputs du processus de *création/raffinage du Program ou Portfolio Backlog priorisé*.

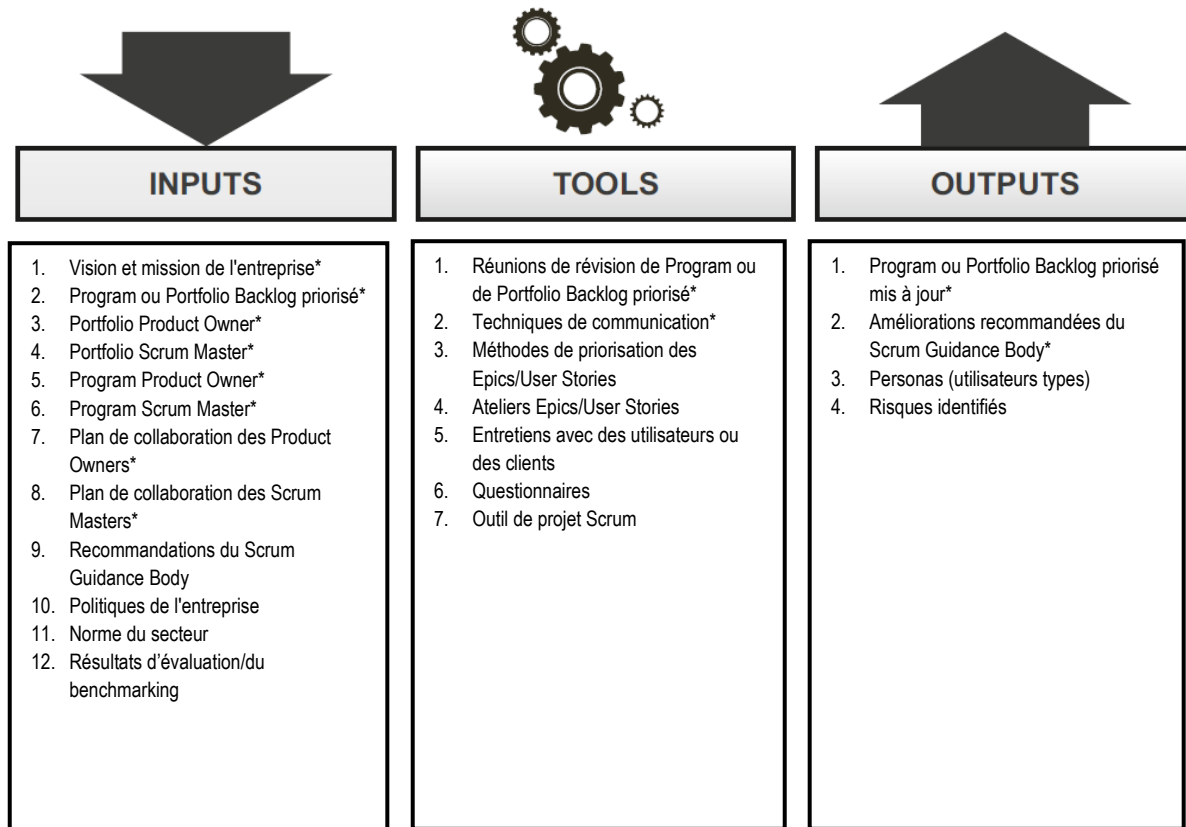


Figure 14-6 : Création/raffinage du Program ou Portfolio Backlog priorisé—inputs, outils et outputs

Note : Les astérisques (*) indiquent un input, un outil ou un output « obligatoire » pour le processus correspondant.

14.6.1 Inputs

14.6.1.1 Vision et mission de l'entreprise*

Description dans la section 14.3.1.1.

14.6.1.2 Program ou Portfolio Backlog priorisé*

Le Program ou Portfolio Product Owner élabore le Program ou Portfolio Backlog priorisé, qui contient une liste priorisée d'exigences commerciales et de projet de haut niveau, de préférence rédigées sous la forme de grandes Epics au niveau du programme ou du portfolio. Ils sont ensuite raffinés par les Product Owners des projets individuels lorsqu'ils créent et hiérarchisent les backlogs de produits pour leurs propres projets. Ces Product Backlogs priorisés ont des User Stories beaucoup plus petites mais plus détaillées qui peuvent être estimées et validées par les équipes Scrum individuelles. Ces détails sont partagés entre les différents projets afin d'éviter toute duplication inutile des efforts.

Le Program Backlog priorisé joue un rôle très similaire au niveau du programme à celui du Product Backlog par priorisé au niveau du projet. Il saisit les exigences du programme et leurs priorités. Il existe toutefois quelques différences. La création des produits livrables respectifs et leur acceptation sont traitées au sein des projets du programme. Les critères « Done » (Terminé) ou d'acceptation pour chaque élément du Product Backlog/de la User Story peuvent être définis au niveau du programme. Les équipes doivent se conformer à ces critères, mais de nouveaux critères peuvent être ajoutés, le cas échéant.

La durée d'un sprint est spécifique au projet et peut donc varier d'un projet à l'autre au sein d'un programme. En outre, la vélocité varie d'une équipe à l'autre. Par conséquent, il n'est pas nécessaire d'avoir des User Stories très granulaires au niveau du programme. Généralement, les programmes ont des exigences de haut niveau sous forme d'Epics, et le raffinement des Epics au niveau du programme suffit à garantir que chaque Epic respective est clairement comprise et que des critères d'acceptation tangibles pour le programme peuvent être définis.

Le Portfolio Backlog priorisé joue le même rôle au niveau du portfolio que le Program Backlog priorisé au niveau du programme. Les éléments du Portfolio Backlog priorisé alimentent les divers Program Backlogs priorisés et, en fin de compte, les Product Backlogs priorisés de chacun des projets correspondants. Comme dans le cas du carnet de commandes des programmes classés par ordre de priorité, le raffinement des Epics est minime, voire inexistant, à ce niveau, car il est effectué dans le cadre des projets associés au niveau de chacun des carnets de commandes des produits classés par ordre de priorité.

Le Program ou Portfolio Backlog priorisé est continuellement raffiné par le Program ou Portfolio Product Owner pour s'assurer que de nouvelles exigences commerciales sont ajoutées, et que les exigences existantes sont correctement documentées et priorisées. Cela permet de s'assurer que les exigences les plus utiles pour atteindre les objectifs du portfolio ou du programme sont priorisées en priorité élevée et que les autres sont priorisées en priorité inférieure.

Le Program ou Portfolio Backlog priorisé présente une image plus large de tous les projets qui font partie du programme ou du portfolio. Par conséquent, il peut fournir des indications importantes concernant les buts, la portée et les objectifs du projet, ainsi que les avantages commerciaux attendus.

14.6.1.3 **Portfolio Product Owner***

Le Portfolio Product Owner est responsable de la création et du raffinage du Portfolio Backlog priorisé. Pour plus d'informations sur le rôle du Portfolio Product Owner, voir la section 3.7.4.2.

14.6.1.4 **Portfolio Scrum Master***

Au niveau du portfolio, le Program Scrum Master joue un rôle similaire à celui que le Program Scrum Master joue pour un programme. Il ou elle est un médiateur, qui résout les problèmes et élimine les obstacles au niveau du portfolio. Pour plus d'informations sur le rôle du Portfolio Scrum Master, voir la section 3.7.4.4.

14.6.1.5 **Program Product Owner***

Au niveau du programme, le Program Scrum Owner est responsable de la création et du raffinage du Program Product Backlog priorisé. Pour plus d'informations sur le rôle du Program Product Owner, voir la section 3.7.4.1.

14.6.1.6 **Program Scrum Master***

Au niveau du programme, le Program Scrum Master joue un rôle similaire à celui du Scrum Master dans un projet. Il ou elle est un médiateur, qui résout les problèmes et élimine les obstacles au niveau du programme. Pour plus d'informations sur le rôle du Program Scrum Master, voir la section 3.7.4.3.

14.6.1.7 **Plan de collaboration des Product Owners**

Description dans la section 13.2.2.

14.6.1.8 **Plan de collaboration des Scrum Masters**

Description dans la section 13.2.3.

14.6.1.9 **Recommandations du Scrum Guidance Body**

Lors de la création et du raffinage du Program ou Portfolio Backlog priorisé, les recommandations du Scrum Guidance Body indiquent les meilleures pratiques qui devraient être prises en considération au niveau du programme ou du portfolio. Pour plus d'informations sur les recommandations du Scrum Guidance Body, voir les sections 8.1.1.7 et 10.3.1.11.

14.6.1.10 **Politiques de l'entreprise**

Les politiques de l'entreprise sont une série de principes, règles et directives formulés ou adoptés par une organisation. La modification des politiques de l'entreprise pourrait affecter les Epics ou les User Stories existantes qui ont été créées sur la base des politiques existantes.

14.6.1.11 **Norme du secteur**

Les nouvelles normes de secteur, ou des modifications des normes existantes doivent être implémentées de façon à maintenir un produit ou service viable. Par conséquent, les User Stories liées au respect de ces normes doivent être incluses dans le Program ou Portfolio Backlog priorisé et priorisées en conséquence. Parfois, les recommandations du Scrum Guidance Body doivent également être modifiées pour refléter des normes industrielles nouvelles ou modifiées.

14

14.6.1.12 **Résultats d'évaluation/du benchmarking**

Avant tout, les résultats de l'évaluation ou du benchmarking nécessiteront une mise à jour du Scrum Guidance Body pour les meilleures pratiques. Les résultats peuvent également aider à définir une norme minimale lors de la création d'un produit ou d'un service et peuvent conduire à des modifications des critères « Done » (Terminé). Parfois, de nouveaux résultats d'évaluation ou de benchmarking peuvent également inciter un Program ou un Portfolio Product Owner à développer de nouvelles Epics pour implémenter les meilleures pratiques supplémentaires ou mises à jour.

14.6.2 **Outils**

14.6.2.1 **Réunions de révision de Program ou de Portfolio Backlog priorisé***

La participation aux réunions de révision de Program ou de Portfolio Backlog est très différente de la participation aux réunions de révision du Product Backlog au niveau du projet. Les équipes Scrum participent aux sessions de raffinage au niveau du projet.

Au niveau du programme ou du portfolio, il y a une représentation de chaque projet au sein du programme ou de chaque programme et/ou projets autonomes dans le portfolio. Cependant, pour rationaliser les réunions, il est généralement recommandé de ne faire participer qu'un ou quelques représentants de chaque projet ou programme au niveau du programme ou du portfolio. Reportez-vous aux sections connexes 6.5.1.2 et 10.3.2.1 pour plus d'informations.

14.6.2.2 **Techniques de communication***

Description dans la section 10.3.2.2.

14.6.2.3 **Méthodes de priorisation des Epics/User Stories**

Au niveau du programme ou du portfolio, il y a normalement un plus petit nombre d'exigences/Epics/User Stories qu'au niveau du projet. De plus, ces exigences seront d'un niveau très élevé et la priorisation sera déterminée principalement par les exigences commerciales (telles que déterminées par les parties prenantes de l'entreprise), le Portfolio Product Owner et le Program Product Owner. For more information on prioritization methods, see section 8.5.2.1.

14.6.2.4 **Ateliers Epics/User Stories**

Par rapport aux projets, les ateliers User Stories pour les programmes et les portfolios visent à produire uniquement des Epics/User Stories de niveau supérieur, ce qui signifie qu'il y aura beaucoup moins d'Epics/User Stories à ce stade. Cependant, les réunions ont toujours de la valeur car elles sont suivies par des représentants des projets du programme ou des programmes du portfolio et ces personnes peuvent transmettre des informations pertinentes à leurs équipes respectives. Ces ateliers sont généralement coordonnés par le Program or Portfolio Scrum Master. Cela garantit que les exigences sont bien définies et comprises dans l'ensemble du programme ou du portfolio. For more information on User Story Workshops, see section 8.4.2.2.

14.6.2.5 **Entretiens avec des utilisateurs ou des clients**

Description dans la section 8.4.2.4.

14.6.2.6 **Questionnaires**

Description dans la section 8.4.2.5.

14.6.2.7 Outil de projet Scrum

Un outil de projet Scrum conçu de manière appropriée fournit une vue facile à comprendre du Program ou Portfolio Backlog priorisé et aide également le Program ou le Portfolio Product Owner à visualiser et à prioriser les exigences/Epics/User Stories. Pour plus d'informations sur l'outil de projet Scrum, voir les sections 2.5.3.1 et 13.3.8.

14.6.3 Outputs

14.6.3.1 Program ou Portfolio Backlog priorisé mis à jour*

Le Program ou Portfolio Backlog priorisé peut être mis à jour avec de nouvelles Epics/User Stories, le travail lié aux nouvelles demandes de changement ou aux risques identifiés ; et/ou pour refléter la redéfinition des priorités des Epics/User Stories existantes. Le raffinement du Program ou Portfolio Backlog peut aboutir à la décision de lancer un nouveau projet (par exemple, pour créer un cadre ou une interface web commune à utiliser par tous les projets).

14.6.3.2 Améliorations recommandées du Scrum Guidance Body*

À la suite du processus de *création/raffinage du Program ou Portfolio Backlog priorisé*, des suggestions ou des commentaires peuvent être fournis pour des améliorations potentielles de la documentation du Scrum Guidance Body. Ces améliorations recommandées seront discutées et acceptées ou rejetées par le Scrum Guidance Body. Si l'une des suggestions est acceptée, elle sera intégrée en tant que mise à jour de la documentation du Scrum Guidance Body. Pour plus d'informations, voir le processus de *révision et de mise à jour du Scrum Guidance Body*.

14.6.3.3 Personas

Description dans la section 8.4.3.2.

14.6.3.4 Risques identifiés

Les risques liés à un programme ou à un portfolio auront également un impact sur les projets qui font partie de ce programme ou de ce portfolio. Pendant l'évaluation des risques du programme ou du portfolio, s'il est déterminé qu'un risque peut affecter un programme ou un projet de niveau inférieur, les informations pertinentes concernant ce risque doivent être communiquées au Product Owner et à l'équipe Scrum respectifs. Les risques liés aux programmes et aux portfolios sont pris en compte dans le processus de *développement des Epics* pour le(s) projet(s) concerné(s) et peuvent avoir un impact global sur la façon dont ce processus est mené. Pour plus d'informations sur l'identification des risques du projet, voir les sections 7.4.1 et 8.4.3.4. Les risques liés au programme et au portfolio sont abordés à la section 7.6.

14.7 Création/mise à jour de release de programme ou de portfolio

Dans ce processus, les releases de programme ou de portfolio sont planifiées, en tenant compte des éventuelles dépendances entre les releases. La planification des releases de programme ou de portfolio aura un impact sur la planification des releases au niveau du projet. Le calendrier de release du programme ou du portfolio est créé et doit être révisé régulièrement en fonction de l'avancement des produits livrables du projet, des exigences nouvelles ou modifiées ou de leurs priorités, et d'autres facteurs.

La réunion de planification des releases sert à examiner les releases existantes et à planifier les nouvelles releases.

Les outputs du processus de *création/mise à jour de release de programme ou de portfolio* deviennent des inputs de processus après leur création initiale. Par exemple, les releases initiales d'un programme ou d'un portfolio peuvent être créées pour la première fois au cours de ce processus, mais elles deviennent des inputs obligatoires pour les futures releases de programme ou de portfolio.

La figure 14-7 présente l'ensemble des inputs, des outils et des outputs du processus de *création/mise à jour de release de programme ou de portfolio*.

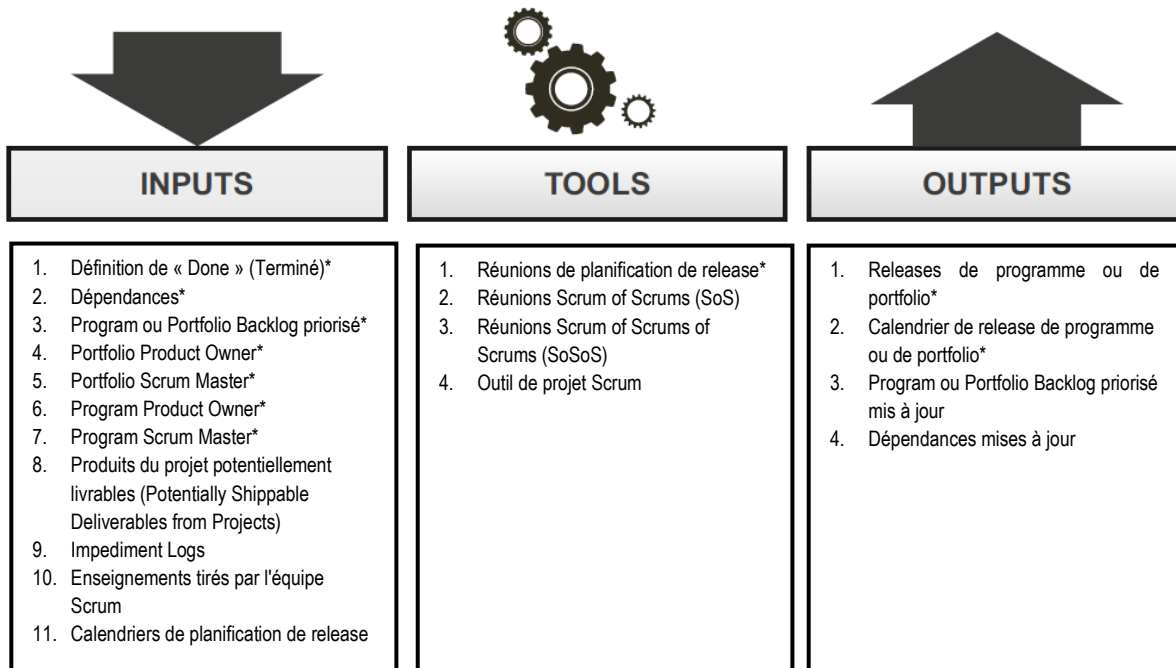


Figure 14-7: Création/mise à jour de release de programme ou de portfolio—inputs, outils et outputs

Note : Les astérisques (*) indiquent un input, un outil ou un output « obligatoire » pour le processus correspondant.

14.7.1 Inputs

14.7.1.1 Définition de « Done » (Terminé)*

Description dans la section 5.4.3.

La définition de « Done » (Terminé) (ou critères « Done ») définie au niveau du programme ou du portfolio peut être utilisée comme critères « Done » (Terminé) minimaux pour les projets dans toute l'entreprise. Pour plus d'informations sur les critères « Done » (Terminé), voir les sections 5.4.3 et 8.5.3.2.

14.7.1.2 Dépendances*

Description dans les sections 8.5.2.6 et 8.5.3.5.

14.7.1.3 Program ou Portfolio Backlog priorisé*

Description dans la section 14.6.1.2.

14.7.1.4 Portfolio Product Owner*

Description dans la section 3.7.4.2.

14.7.1.5 Portfolio Scrum Master*

Description dans la section 3.7.4.4.

14.7.1.6 Program Product Owner*

Description dans la section 3.7.4.1.

14.7.1.7 Program Scrum Master*

Description dans la section 3.7.4.3.

14.7.1.8 **Produits du projet potentiellement livrables (Potentially Shippable Deliverables from Projects)**

Potentially shippable deliverables from projects are valuable inputs for coordination at the program or portfolio level. At the end of each Sprint in a project, product increments or deliverables are completed. The User Stories included in these increments meet the Done Criteria as well as their respective Acceptance Criteria.

14.7.1.9 **Impediment Logs**

Description dans les sections 10.1.1.4 et 14.4.3.4.

14.7.1.10 **Enseignements tirés par les équipes Scrum**

Description dans la section 11.2.3.5.

14.7.1.11 **Calendriers de planification de release**

Ces calendriers, bien que provisoires et susceptibles d'être modifiés, sont essentiels pour évaluer si les projets respectifs sont susceptibles de respecter les délais requis, et sont particulièrement cruciaux en ce qui concerne les dépendances. Pour plus d'informations sur le calendrier de planification de release au niveau du projet, voir la section 8.6.3.1.

14.7.2 **Outils**

14.7.2.1 **Réunions de planification de release***

Les réunions de planification des releases ont lieu entre le Portfolio Product Owner, le Program Product Owner, le Portfolio Scrum Master, le Program Scrum Master, et les autres parties prenantes pertinentes de l'entreprise de l'équipe commerciale et de l'équipe de projet pour s'assurer que toutes les releases de programme et de portfolio sont planifiées correctement. Les releases de programme et de portfolio fournissent à leur tour des informations précieuses pour les releases du plan au niveau du projet.

14.7.2.2 **Réunions Scrum of Scrums (SoS) ***

Description dans la section 13.3.5.

14.7.2.3 Réunions Scrum of Scrums of Scrum (SoSoS) *

Description dans la section 14.4.2.5.

14.7.2.4 Outil de projet Scrum

L'outil de projet Scrum aide les équipes à visualiser facilement les releases de programme et de portfolio, à apporter les modifications appropriées si nécessaire, et à planifier des releases supplémentaires. Pour plus d'informations sur l'outil de projet Scrum, voir les sections 2.5.3.1 et 13.3.8.

14.7.3 Outputs

14.7.3.1 Releases de programme ou de portfolio*

Une release de programme ou de portfolio comprend les releases de tous les projets sous-jacents dans le programme ou le portfolio. Il y a généralement deux types de releases de programme ou de portfolio :

1. Tous les produits livrables des projets sous-jacents sont terminés mais conservés en statut « prêts » pour la release de la manière dont le programme ou le portfolio le déterminent. Il peut y avoir une date de release spécifique à laquelle l'ensemble du programme ou du portfolio est mis à la disposition du client final.
2. Les produits livrables sont mis à disposition au niveau du projet dès qu'ils sont prêts, par exemple dans le cadre de DevOps, qui permet un développement, une implémentation et un déploiement continus.

14.7.3.2 Calendrier de release de programme ou de portfolio*

Le calendrier de release d'un programme ou d'un portfolio comprend les dates cibles des différentes releases planifiées au niveau du programme ou du portfolio.

14.7.3.3 Program ou Portfolio Backlog priorisé mis à jour

Le Program ou Portfolio Backlog priorisé est mis à jour au fur et à mesure que les releases existantes sont examinées et que des changements proposés ou de nouvelles releases sont ajoutés. La mise à jour de release de programme ou de portfolio peut légalement avoir un impact sur la priorisation des Epics/User Stories dans le Program ou Portfolio Backlog priorisé. Pour plus d'informations sur le Program ou Portfolio Backlog priorisé, voir la section 14.6.1.2.

14.7.3.4 Dépendances mises à jour

Les dépendances peuvent être mises à jour en fonction des discussions qui ont lieu lors des réunions ou des sessions de planification de release.

14.8 Rétrospective de release de programme ou de portfolio

Dans ce processus, le Program ou le Portfolio Product Owner et les parties prenantes de l'entreprise se réunissent pour faire une rétrospective d'une release de programme ou de portfolio et également pour discuter et internaliser les enseignements appris. Généralement, les enseignements mènent à l'implémentation d'améliorations convenues réalisables (agreed actionable improvements) lors des futurs projets. Parfois, des améliorations du Scrum Guidance Body peuvent être recommandées. Ces réunions peuvent être programmées après la publication de chaque programme ou portfolio.

La figure 14-8 montre tous les inputs, outils et outputs du processus de *rétrospective de release de programme ou de portfolio*.

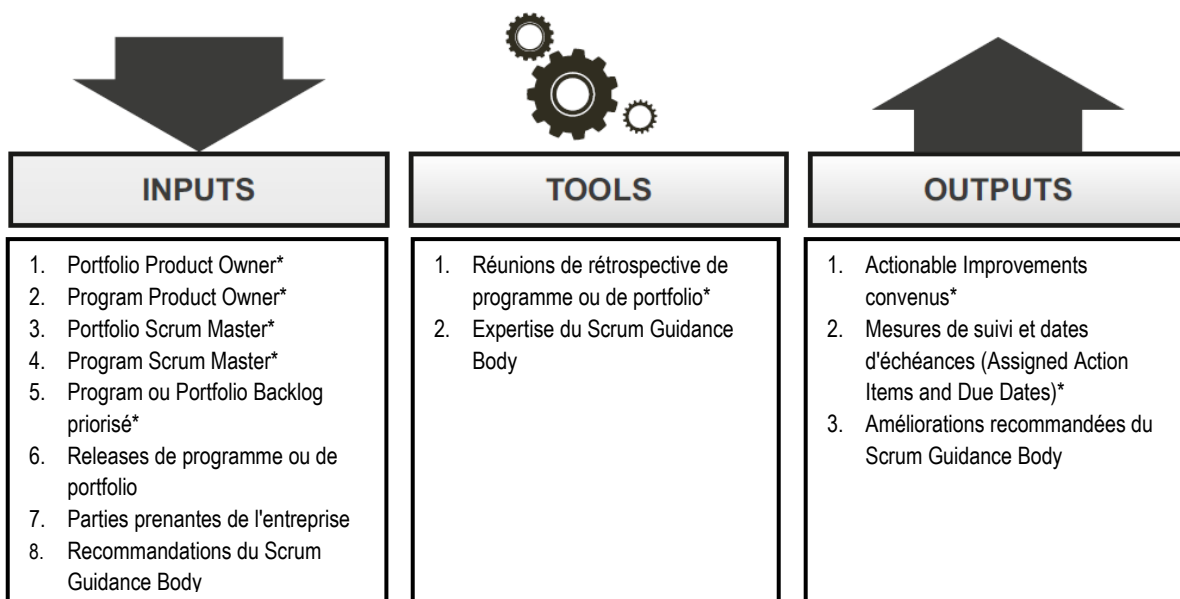


Figure 14-8 : Rétrospective de release de programme ou de portfolio—inputs, outils et outputs

14.8.1 Inputs

14.8.1.1 Portfolio Product Owner*

Description dans la section 3.7.4.2.

14.8.1.2 **Portfolio Scrum Master***

Description dans la section 3.7.4.4.

14.8.1.3 **Program Product Owner***

Description dans la section 3.7.4.1.

14.8.1.4 **Program Scrum Master***

Description dans la section 3.7.4.3.

14.8.1.5 **Program ou Portfolio Backlog priorisé***

Description dans la section 14.6.1.2.

14.8.1.6 **Releases de programme ou de portfolio***

Description dans la section 14.7.3.1.

14.8.1.7 **Parties prenantes de l'entreprise**

Description dans les sections 3.3.2 et 14.3.3.5.

14.8.1.8 **Recommandations du Scrum Guidance Body**

Lors d'une rétrospective de releases de programme ou de portfolio, les recommandations du Scrum Guidance Body fournissent les meilleures pratiques pertinentes, y compris des informations sur les procédures administratives, les audits, les évaluations et les critères de transition des projets. Ce rôle est similaire à celui que jouent les recommandations du Scrum Guidance Body lors des rétrospectives au niveau du projet (décrites à la section 12.2.1.5).

14.8.2 Outils

14.8.2.1 Réunions de rétrospective de programme ou de portfolio*

La réunion de rétrospective de programme ou de portfolio est similaire à la réunion de rétrospective de release tenue au niveau du projet (voir section 12.2.2.1). La principale différence réside dans le fait que les réunions de rétrospective de programme ou de portfolio sont tenues beaucoup moins fréquemment que les réunions de rétrospective de release et qu'elles incluent le Program/Portfolio Product Owner, le Program/Portfolio Scrum Master et les parties prenantes de l'entreprise pour le programme ou le portfolio.

14.8.2.2 Expertise du Scrum Guidance Body

Description dans la section 8.4.2.7.

14.8.3 Outputs

14.8.3.1 Actionable Improvements convenus*

Description dans la section 11.2.3.1.

14.8.3.2 Mesures de suivi et dates d'échéances*

Description dans la section 11.2.3.2

14.8.3.3 Améliorations recommandées du Scrum Guidance Body

À la suite du processus de *rétrospective de release de programme ou de portfolio*, des suggestions et des retours d'information peuvent être formulés afin d'améliorer la documentation du Scrum Guidance Body. Ces améliorations recommandées seront discutées et acceptées ou rejetées par le Scrum Guidance Body. Si l'une des suggestions est acceptée, elle sera intégrée en tant que mise à jour de la documentation du Scrum Guidance Body. Pour plus d'informations, voir le processus de *révision et de mise à jour du Scrum Guidance Body*.

ANNEXE A. UN APERÇU DU CONCEPT AGILE

A.1 Introduction

L'objectif de cette annexe est de permettre aux lecteurs de se familiariser avec le concept de développement agile ainsi qu'avec les différentes méthodologies agiles.

Elle inclut les sections suivantes :

A.2 Aperçu — cette section définit le concept agile et traite des facteurs à l'origine de sa démocratisation.

A.3 Manifeste Agile — cette section présente le *Manifeste Agile*, ses principes ainsi que la *Déclaration d'Interdépendance* afin de situer agile dans son contexte historique.

A.4 Méthodes agiles — cette section offre un bref aperçu de quelques méthodologies agiles spécifiques, notamment :

- Lean Kanban
- Extreme Programming
- Méthodes Crystal
- Dynamic Systems Development Methods
- Feature Driven Development
- Test Driven Development
- Adaptive Software Development
- Agile Unified Process
- Conception pilotée par le domaine

A.2 Aperçu

Le terme « agile » fait souvent référence au fait d'être capable de bouger ou réagir rapidement et facilement, habile. Quelle que soit la discipline de gestion, l'agilité est une qualité à laquelle il est raisonnable d'aspirer. Spécifiquement, la gestion de projet agile implique d'être flexible durant la création d'un produit, service ou autre résultat.

Il est important de comprendre que même si les méthodes de développement agiles sont hautement adaptatives, il est également nécessaire d'intégrer des éléments de stabilité au sein de leurs processus.

A.2.1 Démocratisation du concept agile

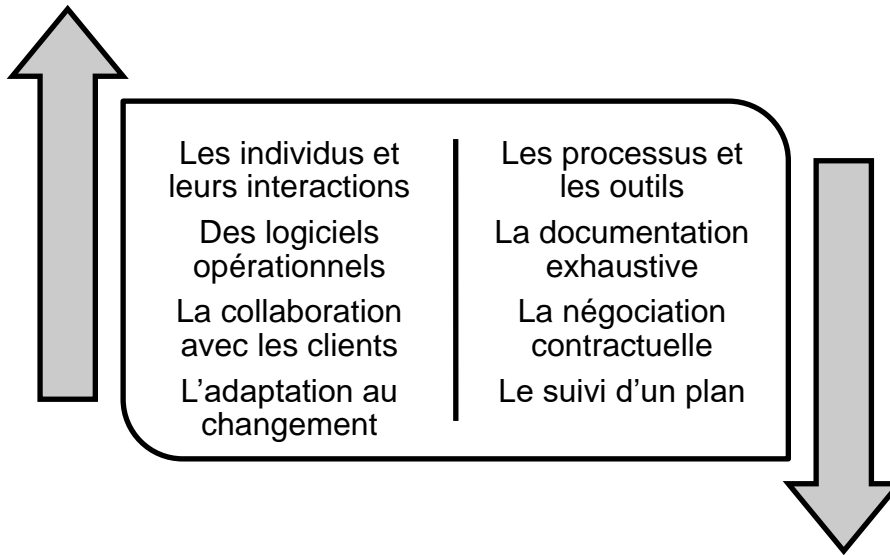
Les changements rapides en termes de technologie, de demandes de marché et d'attentes ont résulté en un nombre grandissant de défis associés au développement de produits et services en utilisant les modèles de gestion de projet traditionnels. Cela a mené à la conceptualisation ainsi qu'à l'implémentation de méthodes et valeurs agiles dans de nombreuses organisations. Les modèles de développement agiles adressent les manquements associés aux modèles de gestion de projet traditionnels. Ils tentent de répondre aux demandes et attentes environnementales en constante augmentation auxquelles les organisations doivent faire face. Puisque les modèles de gestion de projet traditionnels mettent généralement l'accent sur la planification préalable détaillée et sur le respect du plan une fois que celui-ci a été établi, de tels modèles ne réussissent pas à répondre à la réalité de changements environnementaux fréquents.

Le concept agile se fonde sur la planification adaptative ainsi que sur le développement et la livraison en itérations. Il se concentre principalement sur la valeur offerte par les personnes qui réalisent le travail de façon efficace. Bien que les méthodologies adaptatives et incrémentielles existent depuis les années 1950, seules les méthodologies qui respectent le *Manifeste Agile* sont généralement considérées comme véritablement « agiles ».

A.3 Manifeste Agile

En février 2001, un groupe de 17 gourous de l'informatique, développeurs de logiciels, et managers ont organisé une retraite afin de discuter de méthodes légères pour le développement de logiciels. Ensemble ils ont formé l'*Alliance Agile* et les discussions tenues lors de leurs réunions ont ensuite résulté en un *Manifeste pour le Développement Agile de Logiciels* ou *Manifeste Agile*. Le Manifeste a été écrit par Fowler et Highsmith (2001) puis signé par tous les participants afin d'établir les orientations fondamentales de toute méthode agile. L'objectif du *Manifeste Agile* a été établi comme suit :

*Nous découvrons comment mieux développer des logiciels
par la pratique et en aidant les autres à le faire.
Ces expériences nous ont amenés à valoriser :*



*Nous reconnaissons la valeur des seconds éléments,
mais privilégions les premiers.*

| | | |
|-------------------|----------------|------------------|
| Kent Beck | James Grenning | Robert C. Martin |
| Mike Beedle | Jim Highsmith | Steve Mellor |
| Arie van Bennekum | Andrew Hunt | Ken Schwaber |
| Alistair Cockburn | Ron Jeffries | Jeff Sutherland |
| Ward Cunningham | Jon Kern | Dave Thomas |
| Martin Fowler | Brian Marick | |

La permission de reproduire a été accordée par les auteurs ci-dessus par le biais de l'avis publié sur <https://agilemanifesto.org/iso/fr/manifesto>. Traduction originale en français par le Club Agile Rhône-Alpes, Arnaud Pierrel, Bruno Orsier, Christophe Deniaud. Révision par: Claude Aubry, François Beauregard, Laurent Bossavit, Nathalie Gilbert.

Les quatre compromis mis en avant dans le *Manifeste Agile* sont développés comme suit :

1. Les individus et leurs interactions plus que les processus et les outils

Bien que les processus et les outils aident à la réalisation d'un projet, ce sont les individus qui réalisent, participent à, implémentent un projet et déterminent quels processus et outils utiliser. Les acteurs fondamentaux de tout projet sont donc les individus et l'accent doit être placé sur eux et sur leurs interactions plutôt que sur des processus et des outils complexes.

2. Des logiciels opérationnels plus qu'une documentation exhaustive

Bien que la documentation soit nécessaire et utile à tout projet, beaucoup d'équipes se focalisent sur le recueil et l'enregistrement de descriptions qualitatives et quantitatives des incréments de produit, tandis que la réelle valeur livrée au client prend principalement la forme d'un logiciel opérationnel. L'approche agile se focalise donc sur la livraison d'un logiciel opérationnel sous forme d'incrément tout au long de la vie du produit plutôt que sur la documentation détaillée.

3. **La collaboration avec les clients plus que la négociation contractuelle**

Traditionnellement, les clients sont considérés comme des acteurs extérieurs qui participent principalement au début et à la fin du cycle de vie du produit et dont la relation est fondée sur des contrats et sur le respect de ces derniers. Le concept agile est fondé sur une approche de valeur partagée dans laquelle les clients sont perçus comme des collaborateurs. L'équipe de développement et le client travaillent ensemble afin de faire évoluer et de développer le produit.

4. **L'adaptation au changement plus que le suivi d'un plan**

Dans le marché actuel où les exigences du client, les technologies disponibles et les modèles commerciaux sont en constante évolution. Il est essentiel d'aborder le développement de produit d'une manière adaptative pour permettre l'incorporation des changements et des cycles de développement de produit rapides plutôt que de mettre l'accent sur le suivi d'un plan basé sur des données potentiellement obsolètes.

A.3.1 Principes du *Manifeste Agile*

Les 12 principes du *Manifeste Agile* par Fowler et Highsmith (2001) sont :

1. Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
2. Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
3. Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
4. Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
5. Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
6. La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
7. Un logiciel opérationnel est la principale mesure d'avancement.
8. Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.

9. Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
10. La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
11. Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.
12. À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

A.3.2 Déclaration d'interdépendance

La *Déclaration d'Interdépendance* de la gestion de projet agile a été écrite au début de l'année 2005 par un groupe de 15 chefs de projet afin de compléter le *Manifeste Agile*. Il énumère six valeurs managériales nécessaires afin de renforcer la mentalité de développement agile, en particulier lors de la gestion de projets complexes et incertains.

La déclaration souligne que les équipes d'un projet, les clients et les autres parties prenantes sont interdépendantes, connectées et doivent être conscientes de cela afin de réussir. Les valeurs en elles-mêmes sont également interdépendantes.

Nous...

augmentons le retour sur investissement en nous concentrant sur la livraison continue de valeur.

fournissons des résultats fiables en motivant les clients à avoir de fréquentes interactions et à partager la propriété du résultat.

nous attendons à l'incertitude et nous la gérons à travers les itérations, en anticipant, et en nous adaptant.

donnons libre court à la créativité et à l'innovation en reconnaissant que les individus sont la source fondamentale de valeur et en créant un environnement dans lequel ils peuvent faire la différence.

améliorons significativement les performances en engageant le groupe sur les résultats et en partageant la responsabilité sur l'efficacité de l'équipe.

améliorons l'efficacité et la fiabilité grâce à des stratégies, processus et pratiques spécifiques au contexte.

Anderson. D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., et al. 2005

A.4 Méthodes agiles

Un certain nombre de méthodologies agiles sont nées et ont gagné en popularité dans les années 1990 et au début des années 2000. Bien qu'elles diffèrent en un certain nombre de points, leur similitude découle du respect du *Manifeste Agile*.

Les méthodes agiles suivantes sont brièvement introduites ci-après :

1. Lean Kanban
2. Extreme Programming (XP)
3. Méthodes Crystal
4. Dynamic Systems Development Methods (DSDM)
5. Feature Driven Development (FDD)
6. Test Driven Development (TDD)
7. Adaptive Software Development (ASD)
8. Agile Unified Process (AUP)
9. Conception pilotée par le domaine (DDD)

A.4.1 Lean Kanban

Le concept lean (« léger ») optimise le système d'une organisation pour produire des résultats de valeur en fonction de ses ressources, besoins et alternatives tout en réduisant le gaspillage. Le gaspillage peut résulter de la fabrication du mauvais élément, de l'incapacité à apprendre ou de pratiques qui entravent les processus. Étant donné que ces facteurs sont de nature dynamique, une organisation lean évalue son système dans son ensemble et ajuste continuellement ses processus. Le fondement de la méthode lean est que la réduction de la durée de chaque cycle (c-à-d. une itération) entraîne une productivité améliorée en réduisant les délais, permet l'identification des erreurs de façon précoce et réduit donc la quantité totale d'efforts requis afin de réaliser une tâche. Les principes logiciels lean sont appliqués au développement de logiciels avec succès.

Kanban signifie littéralement « panneau » ou « affiche » et adhère à l'utilisation d'aides visuelles afin d'assister à et d'effectuer le suivi de la production. Ce concept a été présenté par Taiichi Ohno, considéré comme le père des systèmes de production Toyota (TPS pour *Toyota Production Systems*). L'utilisation d'aides visuelles est efficace et est devenue une pratique commune. Quelques exemples incluent notamment les fiches de tâches, les scrumboards et les burndown charts. Ces méthodes ont attiré l'attention en raison de leur utilisation par Toyota, un leader en termes de gestion des processus. La méthodologie Lean Kanban intègre l'utilisation de méthodes visuelles telles qu'elles sont décrites par kanban et les principes lean créant ainsi un système de gestion des processus visuel, incrémentiel et évolutif.

A.4.2 Extreme Programming

L'Extreme Programming (XP), né au sein de la Chrysler Corporation, a gagné en popularité dans les années 1990. Au fil du temps, XP permet d'éviter la forte augmentation des frais relatifs au changement de logiciel. Les principales caractéristiques de XP incluent le développement incrémentiel, la planification flexible, les codes de test automatisés, la communication verbale, le design en évolution constante, la collaboration étroite et l'association des moteurs à long terme et à court terme de tous ceux qui sont impliqués.

XP valorise la communication, le feedback, la simplicité et le courage. Les différents rôles de l'approche XP incluent le client, le développeur, le tracker et le coach. Il prescrit différentes pratiques de codage, de développement et pratiques commerciales ainsi que des événements et des artefacts afin de développer de façon efficace et performante. L'approche XP est largement employée en raison de ses pratiques techniques clairement définies.

A.4.3 Méthodes Crystal

Les méthodologies Crystal en matière de développement de logiciels ont été créées par Alistair Cockburn au début des années 1990. Les méthodes Crystal se concentrent sur les personnes, elles sont légères et faciles à adapter. Étant donné que les personnes sont fondamentales, les processus et les outils de développement ne sont pas fixés mais sont plutôt ajustés aux exigences et caractéristiques spécifiques du projet. Le spectre chromatique est utilisé afin de décider de la variante d'un projet. Les facteurs tels que le confort, les fonds facultatifs, les fonds essentiels et la vie jouent un rôle important en vue de déterminer le « poids » de la méthodologie, représenté sous différentes couleurs du spectre chromatique. La famille des méthodes Crystal est divisée en Crystal Clear, Yellow, Orange, Orange Web, Red, Maroon, Diamond et Sapphire.

Toutes les méthodes Crystal ont quatre rôles — sponsor exécutif, concepteur principal, développeurs et utilisateurs expérimentés. Les méthodes Crystal recommandent différentes stratégies et techniques afin de parvenir à l'agilité. Un cycle de projet Crystal se compose de l'affrètement, du cycle de livraison et de la clôture.

A.4.4 Dynamic Systems Development Methods (DSDM)

Le cadre des Dynamic Systems Development Methods (DSDM) a été publié en 1995 et est administré par le DSDM Consortium. DSDM définit la qualité et les efforts en termes de coûts et de temps au début et ajuste les incréments du projet afin de respecter des critères définis en priorisant les incréments dans les catégories suivantes : « vital » (must have), « essentiel » (should have), « confort » (could have) et « impossible/luxe » (won't have) en utilisant la méthode de priorisation MoSCoW. DSDM est une méthode axée sur les systèmes et comporte six phases distinctes — l'avant-projet, l'étude de faisabilité, les fondations, l'exploration et la conception, le déploiement et l'évaluation des bénéfices.

A.4.5 Feature Driven Development (FDD)

Le Feature Driven Development (FDD) a été créé par Jeff De Luca en 1997. Il se base sur la réalisation d'un projet en le divisant en petites fonctions de valeur pour le client et qui peuvent être livrées en moins de deux semaines. FDD possède deux principes fondamentaux — le développement de logiciels est une activité humaine et le développement de logiciels est une fonctionnalité de valeur pour le client.

FDD définit six rôles principaux — chef de projet, chef architecte, responsable de développement, chefs programmeurs, propriétaires de classe, et experts de secteur avec un certain nombre de rôles secondaires. Le processus FDD est itératif et consiste à développer un modèle global, à établir une liste des fonctionnalités et ensuite à planifier, concevoir et fabriquer par fonctionnalité.

A.4.6 Test Driven Development (TDD)

Aussi connu sous le nom de Test-First Development, le Test Driven Development a été introduit par Kent Beck, un des créateurs de l'Extreme Programming (XP). Test Driven Development est une méthode de développement de logiciels qui implique d'écrire le code de test automatique en premier et de développer la plus petite quantité de code possible afin de réussir le test dans un second temps. Le projet entier est divisé en petites fonctionnalités de valeur pour le client qui doivent être développées au cours d'un cycle de développement le plus court possible. Les tests sont écrits sur la base des exigences et spécifications du client. Les tests conçus lors de la phase précédente sont utilisés afin de concevoir et d'écrire le code de production.

TDD peut être catégorisé en deux niveaux : Acceptance TDD (ATDD) qui nécessite un test d'acceptation distinct et Developer TDD (DTDD) qui implique l'écriture d'un test développeur unique. TDD est devenu populaire en raison des nombreux avantages qu'il offre tels que les résultats rapides et fiables, le feedback continu et le temps de débogage réduit.

A.4.7 Adaptive Software Development (ASD)

L'Adaptive Software Development (ASD) est né du travail de développement rapide d'applications par Jim Highsmith et Sam Bayer. Les avantages de l'ASD sont l'adaptation continue des processus au travail à faire, la résolution des problèmes survenant dans les gros projets et le développement itératif en incréments avec un prototypage continu.

C'est une approche de développement axée sur les risques et tolérante aux changements. L'ASD part du principe qu'un plan ne peut pas accepter les incertitudes et les risques car ceux-ci sont révélateurs d'un plan défectueux et raté. L'ASD est une approche qui se base sur les fonctionnalités et axée sur les objectifs. La première phase de développement ASD est la phase de spéculation (par opposition à la planification) suivie par les phases de collaboration et d'apprentissage.

A.4.8 Agile Unified Process (AUP)

Agile Unified Process (AUP) est une évolution du Rational Unified Process d'IBM. Développé par Scott Ambler, AUP associe les techniques agiles éprouvées dans le secteur telles que Test Driven Development (TDD), Agile Modeling, la gestion agile des changements et le remaniement de base de données afin d'offrir un produit opérationnel de la meilleure qualité.

Les processus et techniques de l'approche AUP sont basés sur les valeurs de simplicité, d'agilité, de personnalisation, d'auto-organisation, d'indépendance des outils et se focalisent sur les activités à forte valeur. Les principes et les valeurs de l'approche AUP sont mis en œuvre lors des phases de création, d'élaboration, de construction et de transition.

A.4.9 Conception pilotée par le domaine (DDD)

La conception pilotée par le domaine ou DDD (domain-driven design) est une approche de développement agile destinée à traiter les conceptions complexes avec l'implémentation associée à un modèle évolutif. Elle a été conceptualisée par Eric Evans en 2004 et est axée sur la conception comme domaine fondamental. Le « domaine » définit un secteur d'activité auquel l'utilisateur applique un programme ou une fonctionnalité. Un certain nombre de ces secteurs sont regroupés pour concevoir un modèle. Le modèle consiste en un système d'abstraction qui peut être utilisé afin de concevoir le projet dans son ensemble et de résoudre les problèmes relatifs aux domaines regroupés. Les valeurs fondamentales de la conception DDD incluent notamment la conception axée et sur un domaine et sur un modèle, le langage omniprésent et un contexte délimité.

Dans la conception DDD le langage omniprésent est établi et le domaine est modélisé. La conception, le développement et les tests suivent ensuite. Le modèle du domaine est affiné et remanié jusqu'à ce que celui-ci soit satisfaisant.

ANNEXE B. AUTEURS ET CONTRIBUTEURS DU *GUIDE SBOK®*

Cette annexe liste les personnes qui ont contribué au développement et à la production du *GUIDE SBOK®*.

SCRUMstudy® remercie toutes ces personnes pour leur soutien constant et reconnaît leurs contributions au développement du *Guide SBOK®*.

B.1 Auteur principal

Tridibesh Satpathy

B.2 Coauteurs et comité d'experts

Winfried Hackmann

Gaynell Malone

J. Drew Nations

Buddy Peacock

Deepak R

Ruth Kim

Nikhil Kumar

B.3 Contributeurs et réviseurs

James Pruitt

Elizabeth Lynne Warren

Syed Ashraf

Melvin Wofford Jr.

Srinivas Reddy Kandi

Girish Kulkarni

Abdelnaser Dwaikat

Anu Ravi

Morris Feigel

Ian Glenister

Jo Pereira

Saurabh Gupta

Nikhil Bhargava
Simon Robertson
Dusan Kamenov
Prof. Dr. Akram Hassan
Meena Elangovan
Olumide Idowu
Vinod Kumar
Michael Rauch
Joe Schofield

B.4 Éditions précédentes

Coauteurs et comité d'experts

R-A Alves
Winfried Hackmann
Quincy D. Jordan
Gaynell Malone
J. Drew Nations
Buddy Peacock
Karen Lyncook
Jaimie M. Rush
Elizabeth Lynne Warren
Ruth Kim
Mehul Doshi
Gaurav Garg
Ajey Grandhem
Sayan Guha
Vinay Jagannath
Deepak Ramaswamy
Ahmed Touseefullah Siddiqui
Nikhil Kumar

B.5 Réviseurs et équipe d'édition

Corey T. Bailey
Sohini Banerjee
Vince Belanger
Bobbie Green
Magaline D. Harvey
Ravneet Kaur
Robert Lamb
Mimi LaRaque
Melissa Lauro
Richard Mather
Lachlan McGurk
Madhuresh Kumar Mishra
Neha Mishra
Yogaraj Mudalgi
Jose Nunez
Obi Nwaojigba
Bryan Lee Perez
James Pruitt
Charles J. Quansah
Frank Quinteros
Nadra Rafee
Tommie L. Sherrill
Barbara Siefken
Sandra A. Strech
Frances Mary Jo Tessler
Chrys Thorsen
Mike Tomaszewski
Ron Villmow

ANNEXE C. MISES À JOUR DE LA QUATRIÈME ÉDITION

Cette annexe fournit un résumé des mises à jour apportées au *Guide SBOK™* — Quatrième édition comparées à l'édition précédente.

C.1 Résumé des changements

Les mises à jour apportées au *Guide SBOK™* — Quatrième édition portent principalement sur les sujets suivants :

- une description améliorée et développée des rôles et responsabilités de l'approche Scrum, plus particulièrement en relation aux gros projets, programmes et portefeuilles
- la clarification et la rationalisation des processus identifiés pour la phase de planification et d'estimation. Cela inclut notamment la simplification des réunions impliquées dans ces processus.
- du contenu supplémentaire sur le passage à l'échelle de Scrum pour les gros projets ou au niveau de l'entreprise.

Des améliorations d'ordre général ont également été apportées au texte dans son entier afin de garantir la précision, la clarté et l'intégrité des informations fournies. Cela inclut notamment la mise à jour des tableaux et figures.

C.2 Mises à jour de la Quatrième édition par chapitre

| Chapitre | Principaux changements apportés |
|----------|--|
| 1 | <ul style="list-style-type: none"> • Amélioration de la cohérence et de la clarté. • Addition de références aux nouvelles certifications SSMC® et SSPOC® (section 1.3). • Mise à jour des processus Scrum (section 1.4.4) afin de refléter les nouveaux noms de processus de la phase de planification et d'estimation (voir chapitre 9). Ajout des processus abordés dans les chapitres 13 et 14 pour le passage à l'échelle de Scrum pour les gros projets et pour les entreprises. |
| 2 | <ul style="list-style-type: none"> • Simplification de la formulation relative aux trois questions quotidiennes et à l'organisation des « mêlées quotidiennes » en termes plus génériques concernant le moment de la journée où la réunion est organisée (section 2.7.1) • Informations supplémentaires au sujet de la réunion de planification de sprint (section 2.7.1) |

| Chapitre | Principaux changements apportés |
|----------|--|
| 3 | <ul style="list-style-type: none"> • De manière générale, ce chapitre a été restructuré afin de consolider les descriptions des rôles et responsabilités inclus dans les rôles principaux Scrum : Product Owner (section 3.4), Scrum Master (section 3.5) et équipe Scrum (section 3.6). Cela inclut notamment des définitions plus développées, particulièrement concernant les rôles relatifs aux gros projets, programmes et portefeuilles. • Résumé des responsabilités (section 3.8) mis à jour pour inclure les rôles de Chief Product Owner et Chief Scrum Master. |
| 4 | <ul style="list-style-type: none"> • Résumé des responsabilités (section 4.8) mis à jour pour inclure les rôles de Chief Product Owner et Chief Scrum Master. |
| 5 | <ul style="list-style-type: none"> • Description améliorée de la définition de <i>terminé</i> (section 5.4.2) et de la définition minimum de <i>terminé</i> (section 5.4.3) • Résumé des responsabilités (section 5.6) mis à jour pour inclure les rôles de Chief Product Owner et Chief Scrum Master. |
| 6 | <ul style="list-style-type: none"> • Résumé des responsabilités (section 6.7) mis à jour pour inclure les rôles de Chief Product Owner et Chief Scrum Master. |
| 7 | <ul style="list-style-type: none"> • Résumé des responsabilités (section 7.7) mis à jour pour inclure les rôles de Chief Product Owner et Chief Scrum Master. |
| 8 | <ul style="list-style-type: none"> • Descriptions des rôles de Product Owner de programme et Scrum Master de programme déplacées dans le chapitre 3 pour des raisons de cohérence. • Changements mineurs apportés à la terminologie et aux chiffres afin de correspondre aux mises à jour effectuées dans d'autres chapitres. |
| 9 | <ul style="list-style-type: none"> • Le processus « d'approbation, d'estimation et de validation des user stories » est remplacé par les deux processus suivants : « Estimation des user stories » (section 9.2) et « validation des user stories » (section 9.3). Cela afin d'offrir plus de clarté quant aux contributions, outils et résultats relatifs aux activités réalisées dans ces processus. • Un nouvel outil « méthodes d'estimation » a été défini afin de consolider les nombreuses techniques d'estimation individuellement mentionnées dans l'édition précédente (sections 9.2.2.3, 9.5.2.3). • Le processus de « création des tâches » a été renommé « identification des tâches » (section 9.4) afin de clarifier les tâches définies ou identifiées en fonction des user stories précédemment validées. • Les contributions, outils et résultats de tous les processus relatifs à la phase de planification et d'estimation ont été évalués et ajustés pour être plus exacts. |
| 10 | <ul style="list-style-type: none"> • Simplification de la formulation relative aux trois questions quotidiennes et à l'organisation des « mêlées quotidiennes » en termes plus génériques concernant le moment de la journée où la réunion est organisée (section 10.2.2.2) • Changements mineurs apportés à la terminologie et aux chiffres afin de correspondre aux mises à jour effectuées dans d'autres chapitres. |

| Chapitre | Principaux changements apportés |
|----------|--|
| 11 | <ul style="list-style-type: none">• Suppression du processus de « rassemblement Scrum of Scrums ». Celui-ci est maintenant abordé dans le chapitre 13, Scrum pour les gros projets.• Changements mineurs apportés à la terminologie et aux chiffres afin de correspondre aux mises à jour effectuées dans d'autres chapitres. |
| 12 | <ul style="list-style-type: none">• Changements mineurs apportés à la terminologie et aux chiffres afin de correspondre aux mises à jour effectuées dans d'autres chapitres. |
| 13 | <ul style="list-style-type: none">• Scrum pour les gros projets — le chapitre entier a été ajouté comme nouveau contenu. |
| 14 | <ul style="list-style-type: none">• Scrum pour l'entreprise — le chapitre entier a été ajouté comme nouveau contenu. |

GLOSSAIRE

Accord sur les produits finis

Les produits qui répondent aux critères d'acceptation reçoivent officiellement l'approbation commerciale et sont validés par le client ou le sponsor.

Actionable Escalations

Dans certains cas, le Scrum Guidance Body peut constater que certaines politiques d'entreprise empêchent l'équipe de profiter au mieux des avantages relatifs à la mise en œuvre de Scrum. Il est alors nécessaire de déclencher un processus d'escalade de façon à obtenir l'approbation d'un changement de politique.

Adaptation

Quand l'équipe Scrum principale et les parties prenantes apprennent grâce à la transparence ainsi qu'à l'inspection et s'adaptent en améliorant le travail effectué.

Affinage du backlog de produit priorités

Il s'agit d'un processus au cours duquel le backlog de produit priorisé est constamment mis à jour et entretenu.

Améliorations actionnables convenues

Il s'agit du principal résultat du processus de rétrospective de sprint. Elles constituent la liste des mesures à prendre imaginées par l'équipe afin de résoudre les problèmes et d'améliorer les processus de façon à optimiser leurs performances pour les sprints à venir.

Améliorations recommandées du Scrum Guidance Body

Après avoir planifié un gros projet, des suggestions peuvent être formulées afin de réviser et d'améliorer les recommandations du Scrum Guidance Body. Si le Guidance Body accepte ces suggestions, celles-ci seront ajoutées en tant que mises à jour dans la documentation du Scrum Guidance Body.

Analyse de Pareto

Cette technique d'évaluation des risques implique de classer les risques en fonction de leur magnitude. Cela aide l'équipe Scrum à gérer les risques suivant l'ordre de leur impact potentiel sur le projet.

Analyse de valeur acquise

L'analyse de valeur acquise analyse la performance réelle du projet comparée à la performance prévue à un moment donné. Elle mesure les variations actuelles dans le calendrier du projet, la performance des coûts et la prévision du coût final sur la base de la performance actuelle déterminée.

Analyse des lacunes

L'analyse des lacunes est une technique utilisée pour comparer l'état actuel avec un état désiré dans le futur et pour déterminer comment combler les lacunes entre les deux.

Analyse des parties prenantes de l'entreprise

Une analyse standard des parties prenantes de l'entreprise permet d'identifier les parties prenantes au niveau du programme ou du portefeuille. Des détails supplémentaires sur parties prenantes de l'entreprise du programme ou du portefeuille peuvent être identifiés en tant que personnages lors des processus de création et d'affinage du backlog de programme ou de portefeuille.

Analyse FFOM (SWOT analysis)

L'analyse FFOM est une approche structurée de la planification de projet qui permet d'évaluer les forces, faiblesses, opportunités et menaces relatives à un projet. Ce type d'analyse permet d'identifier à la fois les facteurs internes et externes qui pourraient affecter un projet.

Analyse Kano

L'analyse Kano a été développée par Noriaki Kano (1984) et consiste en la catégorisation des caractéristiques et exigences en quatre catégories basées sur les préférences du client.

1. Attirant/Attractif
2. Satisfaisant
3. Insatisfaisant
4. Indifférent

Arbre de probabilité

Les évènements potentiels sont représentés sous la forme d'un diagramme avec une branche pour chaque résultat possible de l'évènement. La probabilité de chaque résultat est indiquée sur la branche appropriée et ces valeurs peuvent être utilisées pour calculer l'impact global de l'occurrence des risques pour un projet.

Argent de Monopoly

Une technique qui implique de donner au client de « l'argent de Monopoly » ou « fausse monnaie » pour une somme égale à celle du budget du projet en lui demandant de distribuer cette somme entre les user stories considérées. De cette façon, le client priorise sur la base de ce qu'il est prêt à payer pour chaque user story.

Assurance Qualité

L'assurance qualité fait référence au processus d'évaluation et aux normes qui gouvernent la gestion de la qualité d'un projet afin de s'assurer qu'ils demeurent adéquats. Les activités d'assurance qualité sont effectuées avec le reste du travail.

Atelier d'estimation des tâches

L'atelier d'estimation des tâches permet à l'équipe Scrum d'estimer les efforts requis pour compléter une tâche ou une série de tâches et pour estimer l'effort humain et les autres ressources requises afin d'effectuer les tâches d'un sprint donné.

Atelier d'user stories

Des ateliers d'user stories ont lieu au cours du processus de développement des epics. Ces sessions sont facilitées par le Scrum Master. Elles impliquent toute l'équipe Scrum principale et il est aussi parfois souhaitable d'inclure d'autres parties prenantes de l'entreprise .

Atténuation des risques

L'atténuation des risques est une étape importante dans la gestion des risques qui implique de développer une stratégie adéquate afin de faire face à un risque.

Attitude face aux risques

Essentiellement, l'attitude d'une partie prenante face aux risques sert à déterminer le niveau de risque que celle-ci trouve acceptable. C'est un facteur déterminant permettant de décider quand prendre des mesures pour atténuer les risques défavorables potentiels.

Auto-organisation

L'approche Scrum repose sur le principe que les employés sont motivés et cherchent à accepter des responsabilités plus importantes. Ils offrent donc une bien meilleure valeur ajoutée lorsqu'ils sont autoorganisés.

Aversion aux risques

C'est l'une des catégories d'utility function. Elle fait référence à une partie prenante réticente à accepter un risque peu importe les bénéfices ou opportunités anticipés.

Backlog de produit de programme mis à jour

Un backlog de produit de programme qui subit un affinage régulier afin de prendre en compte les changements et nouvelles exigences.

Backlog de produit priorisés

Le backlog de produit priorisé est un document d'exigences unique qui définit la portée du projet en offrant une liste priorisée des caractéristiques du produit ou du service à développer au cours du projet.

Backlog de programme ou de portefeuille priorisé mis à jour

Le backlog de programme ou de portefeuille priorisé peut être mis à jour avec de nouvelles user stories, de nouvelles demandes de changement, de nouveaux risques identifiés, des user stories mises à jour ou une nouvelle hiérarchisation des user stories existantes.

Backlog de sprint

Le backlog de sprint est une liste des tâches qui doivent être remplies par l'équipe Scrum au cours du sprint à venir.

Benchmarking

Une entreprise se doit de régulièrement comparer ses propres pratiques à celles des entreprises concurrentes afin de rester compétitive. Le benchmarking consiste à comparer les procédures commerciales et les mesures de performance d'une organisation avec celles d'entreprises dominantes dans le même ou dans un autre secteur.

Bénéfices du projet

Ils incluent toutes les améliorations tangibles apportées à un produit, service ou résultat qui peuvent être obtenus par la réalisation d'un projet.

Besoins commerciaux

Les besoins commerciaux sont les besoins auxquels le projet est censé répondre, tels qu'ils sont documentés dans l'énoncé de vision du projet.

Brainstorming

Sessions durant lesquelles les parties prenantes de l'entreprise concernées et les membres de l'équipe Scrum principale partagent ouvertement leurs idées à travers des discussions et des sessions de partage des connaissances, normalement menées par un facilitateur.

Budget du projet

Le budget du projet est un document financier qui inclut les coûts du personnel, des matériaux et autres dépenses associées à un projet. Normalement, le budget du projet est approuvé par le sponsor afin de garantir la disponibilité de fonds suffisants.

Calendrier d'environnements

Un calendrier d'environnements définit le partage des environnements par l'équipe Scrum. Il alloue des jours et plages horaires pour l'utilisation de chaque environnement par chaque équipe.

Calendrier de planification de la release

Un calendrier de planification de la release est le résultat principal du processus de planification de la release. Le calendrier de planification de la release établit quels incréments doivent être mis à disposition des clients, ainsi que les intervalles prévus et dates de mise à disposition. Il est possible qu'une mise à disposition ne soit pas prévue à la fin de chaque sprint.

Calendrier d'équipe

Un calendrier d'équipe contient des informations au sujet de la disponibilité des membres de l'équipe, notamment les informations au sujet des vacances des employés, des congés, des événements importants et des jours fériés.

Cartes d'index (Index Cards)

Les cartes d'index, souvent appelées story cards, sont utilisées pour suivre les user stories durant le projet. Elles augmentent la visibilité et la transparence et facilitent la découverte rapide des problèmes qui pourraient survenir.

Cartographie de la chaîne de valeur

La cartographie de la chaîne de valeur utilise des organigrammes afin d'illustrer le flux des informations requises pour terminer un processus et peut être utilisée pour rationaliser un processus en identifiant les éléments qui n'apportent pas de valeur ajoutée.

Charte de projet

La charte de projet est une déclaration officielle des objectifs et résultats désirés pour le projet. Dans beaucoup d'organisations, la charte de projet est le document qui autorise officiellement et formellement le projet et donne à l'équipe l'autorité nécessaire pour commencer le travail.

Chief Product Owner

Pour les gros projets, le Chief Product Owner prépare et maintient le backlog de produit priorisé global pour le projet. Il coordonne le travail entre les Product Owners et les équipes Scrum. Les Product Owners, à leur tour, gèrent leur part respective du backlog de produit priorisé.

Chief Scrum Master

Pour les gros projets, le Chief Scrum Master est responsable de la modération des réunions Scrum of Scrums (SoS) et de l'élimination des obstacles qui affectent plusieurs équipes à la fois.

Cibler les clients pour la release

Toutes les releases ne ciblent pas tous les utilisateurs ou parties prenantes de l'entreprise . Les parties prenantes de l'entreprise peuvent choisir de limiter certaines releases à un sous-ensemble d'utilisateurs. Le calendrier des releases précise les clients cibles pour la release.

Classement relatif des priorités

Le classement relatif des priorités est une liste simple des user stories par ordre de priorité. C'est une méthode efficace pour déterminer les user stories désirées pour chaque itération ou release de produit ou de service.

Client

Le client est la personne ou l'organisation qui acquiert le produit, le service ou tout autre résultat du projet. Quelle que soit l'organisation, et en fonction du projet, il peut y avoir à la fois des clients internes (c-à-d. au sein de l'organisation même) ou des clients externes (c-à-d. en dehors de l'organisation).

Collaboration

Dans l'approche Scrum, la collaboration fait référence à l'équipe Scrum principale qui travaille avec parties prenantes de l'entreprise pour créer et valider les produits du projet répondant aux objectifs définis dans la vision du projet. La collaboration est quand une équipe travaille ensemble de façon à tirer avantage des contributions de chacun pour produire un meilleur résultat.

Communication des risques

La communication des risques implique de communiquer les résultats des quatre premières étapes de la gestion des risques aux parties prenantes de l'entreprise concernées et de déterminer leur perception des évènements incertains.

Comparaison par paire (Paired Comparison)

Une technique où la liste complète des user stories dans le backlog de produit priorisé est préparée. Puis, chaque user story est considérée individuellement et comparée à chacune des autres user stories dans la liste. Chaque fois que deux user stories sont comparées, une décision est prise pour définir laquelle des deux est la plus importante. À la fin de ce processus, il est possible de générer une liste priorisée des user stories.

Contenu de la release (Release Content)

Les informations essentielles au sujet des produits qui peuvent aider l'équipe de service client.

Contrat de « joint-venture »

Ce contrat est normalement utilisé lorsque deux ou plusieurs parties s'associent afin d'accomplir le travail d'un projet. Les parties impliquées dans le projet réaliseront toutes deux un retour sur investissement car les revenus ou bénéfices générés seront partagés entre les parties.

Contrat de développement par palier

Ce contrat libère des fonds chaque mois ou chaque trimestre après qu'une release a été réalisée avec succès. Cela fournit une incitation à la fois au client et au fournisseur tout en limitant les risques financiers pour le client sur une période de temps définie puisque les releases infructueuses ne sont pas payées.

Contrat de livraison graduelle (Incremental Delivery Contract)

Ce contrat inclut des points d'inspection à intervalles réguliers. Cela aide le client et parties prenantes de l'entreprise à prendre des décisions périodiques au sujet du développement du produit tout au long du projet à chaque point d'inspection. Le client peut soit accepter le développement du produit, décider d'interrompre le développement du produit ou demander des modifications.

Contrat d'incitation et de pénalité

Ce contrat est basé sur l'acceptation du fait que le fournisseur sera récompensé avec une incitation financière si les produits du projet sont livrés à temps, mais entraînera une pénalité financière si la livraison est en retard.

Contrôle empirique des processus

Le contrôle empirique des processus aide à prendre des décisions basées sur l'observation et l'expérimentation plutôt que sur une planification détaillée d'avance. Ce modèle repose sur trois principes fondamentaux : la transparence, l'inspection et l'adaptation.

Contrôle qualité

Le contrôle qualité fait référence à l'exécution des activités de qualité prévues par l'équipe Scrum au cours du processus de création de produits potentiellement livrables. Le contrôle qualité implique également de tirer les enseignements provenant de chaque ensemble d'activités réalisées afin de garantir la progression continue.

Coût d'opportunité

Fait référence à la valeur de la prochaine option ou du prochain projet commercial à avoir été rejeté en faveur du projet choisi.

Coûts du projet

Les coûts du projet sont les investissements et autres coûts de développement d'un projet.

Création de la vision de projet

L'étude de cas du projet est évaluée afin de créer un énoncé de vision du projet qui servira d'inspiration et permettra de recentrer le projet tout entier. Le Product Owner est également identifié au cours de ce processus.

Création des incréments de produit

L'équipe Scrum travaille sur les tâches du backlog de sprint afin de créer des incréments de produit.

Création des user stories

Processus au cours duquel les user stories et leurs critères d'acceptation sont créés. En règle générale, les user stories sont écrites par le Product Owner et sont conçues pour s'assurer que les exigences du client sont clairement expliquées et peuvent être entièrement comprises par toutes parties prenantes de l'entreprise.

Création du backlog de produit priorisé

Les epics sont redéfinis et élaborés puis priorisés afin de créer un backlog de produit priorisé pour le projet. La définition de terminé est également établie à ce moment-là.

Critères d'acceptation des user stories

Chaque user story possède des critères d'acceptation qui lui sont associés. Les user stories sont subjectives et les critères d'acceptation offrent l'objectivité requise afin de déterminer si l'user story est terminée ou non pendant la revue du sprint. Elle offre à l'équipe de la clarté sur ce qui est attendu de l'user story.

Critères d'acceptation minimums

Les critères d'acceptation minimums sont déterminés par l'unité commerciale. Ils deviennent ensuite une partie des critères d'acceptation pour n'importe quelle user story pour cette unité commerciale. Toutes les fonctionnalités définies par l'unité commerciale doivent satisfaire à ces critères d'acceptation minimums en vue d'être acceptées par le Product Owner concerné.

Critères de sélection des membres

Les critères de sélection des membres sont déterminés par parties prenantes de l'entreprise pour définir les membres du Scrum Guidance Body, leurs rôles et responsabilités, le nombre de membres ainsi que l'expertise et les compétences requises.

Critères d'estimation

L'utilisation de critères d'estimation a pour but de maintenir la taille relative des estimations et de minimiser le besoin éventuel d'une nouvelle estimation. Les critères d'estimation peuvent être exprimés de différentes façons, deux exemples courants étant la valeur en story points et la durée idéale.

Cycle PDCA/PDSA

Le cycle Plan-Do-Check-Act (Préparer-Réaliser-Vérifier-Agir) ou roue de Deming ou de Shewhart a été développé par le Dr. W. Edwards Deming (considéré comme le père du contrôle qualité moderne) et le Dr. Walter A. Shewhart. Plus tard, Deming a modifié Plan-Do-Check-Act en Plan-Do-Study-Act (Préparer-Réaliser-Étudier-Agir ou PDSA) car le terme « Study » (étudier) souligne l'analyse alors que « Check » (vérifier) privilégie simplement l'inspection. L'approche Scrum et la roue de Deming/Shewhart (ou cycle PDCA) sont deux méthodes itératives basées sur la progression continue.

Décomposition

La décomposition est un outil permettant de diviser les tâches importantes en plusieurs petites tâches plus détaillées. Les user stories sont décomposées en tâches par les membres de l'équipe Scrum. Les user stories du backlog de produit priorisé doivent être suffisamment décomposées pour permettre à l'équipe Scrum d'avoir les informations nécessaires pour créer des incréments de produits à partir des tâches mentionnées dans la liste de tâches.

Définition de *terminé*

La définition de *terminé* est une série de règles applicables à toutes les user stories. Une définition claire de *terminé* est essentielle car elle permet d'éviter toute ambiguïté au sujet des exigences et d'aider l'équipe à respecter les normes de qualité obligatoires. La définition de *terminé* est le résultat du processus de création du backlog de produit priorisé. Une user story est considérée comme *terminée* lorsque la démonstration en a été faite et qu'elle a été approuvée par le Product Owner qui la juge selon la définition de *terminé* et les critères d'acceptation des user stories.

Demandes de changement

Les requêtes de modification sont normalement présentées sous la forme de demandes de changement. Les demandes de changement restent non ratifiées jusqu'à ce qu'elles soient officiellement approuvées.

Demandes de changement approuvées

Les changements dont l'inclusion dans le backlog de produit priorisé a été approuvée. Dans certains cas, les demandes de changement approuvées peuvent provenir des responsables de programme ou de portefeuille et constituent des contributions qui doivent être intégrées à la liste des changements approuvés du projet afin d'être mises en œuvre lors des sprints à venir.

Demandes de changement non-ratifiées

Les requêtes de modification sont normalement présentées sous la forme de demandes de changement. Les demandes de changement restent non-ratifiées jusqu'à ce qu'elles soient officiellement approuvées.

Démonstration et validation du sprint

L'équipe Scrum fait la démonstration des incréments de produit au Product Owner et aux parties prenantes de l'entreprise concernées lors d'une réunion de revue de sprint.

Dépendances externes

Les dépendances externes sont les dépendances liées aux tâches, activités ou produits situés en dehors de la portée du travail à effectuer par l'équipe Scrum mais qui sont nécessaires à la réalisation d'une tâche du projet ou à la création d'un incrément de produit. Généralement, les dépendances externes sont hors du contrôle de l'équipe Scrum.

Dépendances facultatives

Les dépendances facultatives sont des dépendances qui sont ajoutées au flux de travail par choix. Généralement, les dépendances facultatives sont décidées par l'équipe Scrum sur la base des expériences précédentes et des meilleures pratiques pour un domaine ou secteur particulier.

Dépendances internes

Les dépendances entre les tâches, produits ou activités qui sont sous le contrôle de l'équipe Scrum et font partie du travail à effectuer par l'équipe Scrum.

Dépendances obligatoires

Ces dépendances sont inhérentes à la nature du travail, telles que les contraintes physiques, ou peuvent être dues à des obligations contractuelles ou légales.

Détermination des dépendances

Une fois que l'équipe Scrum a sélectionné les user stories pour un sprint spécifique, elle doit ensuite prendre en compte toutes les dépendances qui y sont associées, y compris celles relatives à la disponibilité du personnel ainsi que toutes les dépendances techniques.

La bonne documentation des dépendances aide l'équipe Scrum à déterminer l'ordre relatif dans lequel les tâches doivent être effectuées pour créer les incréments de produit. Les dépendances mettent aussi en relief les relations et interactions entre les tâches à la fois au sein de l'équipe Scrum travaillant sur un sprint spécifique et entre les différentes équipes Scrum au sein du projet.

Dettes techniques

Lorsque l'équipe travaille à la création des principaux incréments associés au produit du projet, la dette technique fait référence au travail auquel l'équipe n'accorde pas la priorité, omet ou n'effectue pas. Une dette technique s'accumule et doit finalement être remboursée.

Développement des epics

L'énoncé de vision du projet sert de base au développement des epics. Des réunions de groupes d'utilisateurs peuvent être organisées pour développer les epics.

Diagramme de flux cumulatif (DFC)

Un outil utile de suivi des performances du projet. Il offre une représentation simple et visuelle des progrès du projet à un moment donné. Il est généralement utilisé pour faire état du projet global mais pas pour les mises à jour quotidiennes de chaque sprint.

Durée du sprint

Le Product Owner et l'équipe Scrum décident de la durée des sprints pour le projet en se basant sur les informations disponibles, y compris les exigences commerciales et le calendrier de planification de la release. Une fois décidée, la durée du sprint est généralement fixe pour le projet. La durée du sprint est la durée des sprints décidée pour un projet.

Échéances mises à jour d'implémentation du projet

Les échéances d'implémentation d'un projet peuvent être mises à jour afin de refléter l'impact d'un changement ou d'une nouvelle user story qui nécessite de modifier ou d'introduire de nouvelles exigences.

Échéancier du projet

L'échéancier reflète la longueur et la durée d'un projet. L'échéancier associé à l'étude de cas inclut également la durée sur laquelle les bénéfices du projet seront réalisés.

Éléments non-fonctionnels proposés pour le backlog de produit

Les éléments non-fonctionnels peuvent ne pas être complètement définis au début du projet et peuvent apparaître pendant la réunion de revue ou de rétrospective du sprint. Ces éléments doivent être ajoutés au backlog de produit priorisé au fur et à mesure qu'ils sont découverts.

Énoncé de vision du projet

Le principal résultat du processus de création de la vision de projet est un énoncé structuré de vision de projet. Une bonne vision de projet explique les besoins commerciaux et les besoins auxquels le projet doit répondre mais pas comment ceux-ci seront remplis.

Enseignements tirés par l'équipe Scrum

Il est attendu de l'équipe Scrum autoorganisée et responsabilisée qu'elle apprenne des erreurs commises pendant le sprint. Ses enseignements permettent aux équipes d'améliorer leurs performances pour les sprints futurs.

Environnements

Fait référence à l'identification et à la documentation de tous les environnements requis pour développer et tester les produits du projet.

Epics

Les epics sont écrits durant la phase initiale du projet lorsque la majorité des user stories sont des fonctionnalités de haut niveau ou que les descriptions et exigences de produit sont relativement larges. Ce sont des user stories larges et brutes dans le backlog de produit priorisé.

Équipe principale élargie

L'équipe principale élargie comprend le Chief Product Owner, Chief Scrum Master, les Scrum Masters, Product Owners, et des membres spécifiques des équipes Scrum travaillant sur un gros projet.

Équipe Scrum

L'équipe Scrum ou équipe de développement est l'un des rôles de l'équipe Scrum principale. L'équipe Scrum travaille à la création des incréments de produit du projet et contribue à la création de la valeur commerciale pour toutes parties prenantes de l'entreprise et pour le projet.

Estimation d'affinité

Une technique utilisée afin de pouvoir rapidement estimer un grand nombre d'user stories en les catégorisant. Les catégories peuvent être petites, moyennes ou larges, ou peuvent être numérotées avec des valeurs en story points afin d'indiquer leur taille relative. Les principaux avantages de cette approche sont sa transparence, sa visibilité grand public ainsi que sa facilité de mise en œuvre.

Estimation des tâches

Lors d'un atelier d'estimation des tâches, l'équipe Scrum principale estime les efforts requis afin d'effectuer chaque tâche sur la liste des tâches. Le résultat de ce processus est une liste des tâches et efforts estimés.

Estimation des user stories

Le Product Owner clarifie les user stories de façon à permettre au Scrum Master et à l'équipe Scrum d'estimer les efforts requis pour développer la fonctionnalité décrite dans chaque user story.

Estimation en story points (Relative Sizing/Story Points)

En plus d'être utilisés pour l'estimation des coûts, les story points peuvent être utilisés pour estimer la taille globale d'une user story ou caractéristique. Cette approche attribue une valeur en story point basée sur l'évaluation globale de la taille d'une user story en tenant compte des risques, des efforts requis pour sa réalisation et de son niveau de complexité.

Étude de marché

Une étude de marché consiste en la recherche, la collecte, le rassemblement et l'analyse des données relatives aux préférences du client pour des produits. Elle inclut souvent des données importantes sur les tendances et la segmentation du marché ainsi que sur les procédés de marketing.

Évaluation des risques

Consiste à évaluer et estimer les risques identifiés.

Exigences commerciales

Les exigences commerciales définissent ce qui doit être livré de façon à répondre aux besoins commerciaux et à offrir de la valeur ajoutée aux parties prenantes de l'entreprise . La somme de toutes les informations obtenues grâce à divers outils tels que les entretiens avec les utilisateurs ou les clients, les questionnaires, les sessions JAD, l'analyse des lacunes, l'analyse FFOM et autres réunions, permet d'obtenir une meilleure perspective des exigences commerciales et aide à créer le backlog de produit priorisé.

Expérimentation des risques

L'expérimentation des risques consiste en des expériences qui impliquent de rechercher et de créer des prototypes afin de mieux comprendre les risques potentiels. Lors d'une expérimentation, un exercice intense d'une durée d'un ou deux jours est mené (de préférence au début d'un projet, avant les processus de développement des epics ou de création du backlog de produit priorisé) afin d'aider l'équipe à déterminer les incertitudes qui pourraient affecter le projet.

Expertise de l'équipe

Fait référence à l'expertise de l'équipe Scrum pour comprendre les user stories et les tâches du backlog de sprint dans le but de créer les incréments de produits finaux. L'expertise de l'équipe est utilisée pour évaluer les contributions nécessaires à l'exécution du travail prévu pour le projet.

Expertise du Scrum Guidance Body

L'expertise du Scrum Guidance Body fait référence aux règles et législations, aux principes de développement ou standards et meilleures pratiques documentés.

Expertise pour l'écriture d'user stories

Le Product Owner, en fonction de ses interactions avec parties prenantes de l'entreprise, de ses connaissances et de son expertise commerciale, et des contributions de l'équipe, développe les user stories qui forment le backlog de produit priorisé initial pour le projet.

Explorateur - Acheteur - Vacancier - Prisonnier (ESVP)

Un exercice qui peut être mené au début d'une réunion de rétrospective de sprint pour comprendre l'état d'esprit des participants et donner le ton à la réunion. Il est demandé aux personnes présentes d'indiquer de façon anonyme laquelle de ces propositions représente au mieux leur attitude vis-à-vis de la réunion.

Fist of five

Il s'agit d'une technique simple et rapide permettant d'arriver à un consensus au sein d'un groupe et de susciter la discussion. Après une discussion initiale sur une proposition donnée ou une décision à prendre, les membres de l'équipe Scrum doivent voter sur une échelle allant de 1 à 5 en utilisant leurs doigts.

Formation de l'équipe Scrum

Processus d'identification des membres de l'équipe Scrum. Généralement, c'est le Product Owner qui a la responsabilité principale de la sélection des membres de l'équipe mais il le fait souvent en collaboration avec le Scrum Master.

Fourchette d'estimation

Les estimations pour les projets doivent être présentées en fourchettes. Des chiffres exacts pourraient donner l'impression d'être très précis quand ils ne le sont pas réellement. En effet, les estimations, par définition ne sont pas exactes. Les fourchettes d'estimations doivent se baser sur le niveau de confiance que l'équipe a en chaque estimation.

Fournisseur

Les fournisseurs incluent les individus ou organisations externes qui fournissent des produits ou des services qui ne font pas partie des compétences de base de l'organisation du projet.

Gestion de la qualité

Dans l'approche Scrum, la gestion de la qualité permet aux clients d'être mis au courant de n'importe quel problème survenant relativement tôt dans le projet afin de les aider à décider si le projet fonctionnera ou non pour eux. Dans l'approche Scrum, la gestion de la qualité est facilitée par trois activités complémentaires : 1. Planification de la qualité 2. Contrôle qualité 3. Assurance Qualité

Gestion des conflits

Les membres de l'équipe utilisent des techniques de gestion des conflits afin de gérer les conflits qui pourraient survenir au cours d'un projet Scrum. Les sources de conflits incluent souvent les calendriers, les priorités, les ressources, la hiérarchie, les problèmes techniques, les procédures, les personnalités et les coûts.

Identification de l'environnement

Identification du nombre et du type d'environnements nécessaires à cause des différentes équipes Scrum qui commenceront et termineront leur sprint le même jour.

Identification des risques

L'identification des risques est une étape importante dans la gestion des risques qui implique l'utilisation de différentes techniques afin d'identifier les risques potentiels.

Identification des tâches

Les user stories validées sont divisées en tâches spécifiques et rassemblées dans une liste de tâches. Ce processus est réalisé lors de la réunion de planification de sprint.

Identification du Scrum Master et des parties prenantes de l'entreprise

Le Scrum Master et parties prenantes de l'entreprise sont identifiés grâce à des critères de sélection spécifiques.

Incréments de produit

Ce sont les éléments livrables ou incréments qui sont terminés à la fin d'un sprint.

Inspection

Fait référence au suivi requis pour le contrôle empirique des processus afin de garantir que les produits du projet sont conformes aux exigences.

Justification commerciale

La justification commerciale justifie les raisons d'entreprendre un projet. Elle répond à la question « Pourquoi ce projet est-il nécessaire ? » La justification commerciale motive toutes les décisions prises en lien avec un projet.

Justification de valeur continue

La justification de valeur continue fait référence à l'évaluation régulière de la valeur commerciale de façon à déterminer si la justification ou la viabilité du projet persiste.

Justification du projet

La justification du projet inclut tous les facteurs qui justifient la nécessité du projet, qu'ils soient positifs ou négatifs, choisis ou non (p. ex. capacité inadaptée à répondre aux demandes existantes ou prévues, baisse du taux de satisfaction des clients, faibles bénéfices, exigences légales, etc.).

Leader assertif (Assertive Leader)

Un leader assertif affronte les problèmes et fait preuve de confiance pour établir l'autorité avec respect.

Leader autocratique

Les leaders autocratiques prennent leurs décisions seuls, ne permettant peu ou pas de participation ou de discussion de la part de l'équipe avant que la décision ne soit prise. Ce style de leadership ne doit être utilisé qu'en de rares occasions.

Leader déléguant

Un leader déléguant est impliqué dans la majorité des processus de décision. Cependant, il délègue une partie des responsabilités d'organisation et de décision aux membres de l'équipe, en particulier si ceux-ci sont à même de gérer ces tâches. Ce type de leadership convient aux situations où le leader est en phase avec les détails spécifiques du projets et où le temps est limité.

Leader directif

Un leader directif donne des consignes aux membres de l'équipe au sujet des tâches requises et sur la façon dont celles-ci doivent être effectuées.

Leader laissez-faire

Style de leadership où l'équipe est généralement laissée sans supervision et où le leader n'interfère pas avec les activités de travail quotidiennes. Conduit souvent à un état d'anarchie.

Leader serviteur (Servant Leader)

Les leaders de soutien utilisent l'écoute, l'empathie, l'engagement et la perspicacité tout en partageant le pouvoir et l'autorité avec les membres de l'équipe. Les leaders de soutien sont des coordonnateurs qui obtiennent des résultats en se basant sur les besoins de leur équipe. Ce style est l'illustration parfaite du rôle de Scrum Master.

Législations

Fait référence aux législations fédérales, locales et étatiques ou aux règlements de secteur auxquels le programme ou le portefeuille doivent adhérer. Parfois, les recommandations du Scrum Guidance Body doivent être mises à jour afin de prendre en compte de nouvelles législations.

Liste de contrôle des risques (Risk Checklists)

Les listes de contrôle des risques incluent les points principaux à prendre en considération lors de l'identification des risques, les risques les plus couramment rencontrés au cours des projets Scrum, ou même les catégories de risques devant être considérés par l'équipe.

Liste des tâches

C'est une liste complète qui contient toutes les tâches sur lesquelles l'équipe Scrum s'est engagée pour le sprint en cours. Elle contient une description de chaque tâche.

Liste des tâches et efforts estimés

La liste des tâches et efforts estimés est une liste des tâches relatives aux user stories validées incluses dans le Sprint. Les efforts estimés sont exprimés en termes de critères d'estimation tels qu'ils ont été décidés par l'équipe. La liste des tâches et efforts estimés est utilisée par l'équipe Scrum durant les réunions de planification de sprint pour créer le backlog de sprint et le sprint burndown chart.

Liste mise à jour des membres du Scrum Guidance Body

Après avoir évalué les membres actuels du Scrum Guidance Body, de nouveaux membres peuvent être inclus et des membres existants peuvent être retirés ou quitter le Scrum Guidance Body.

Listes de risques

Les listes des risques sont utilisées pour stimuler la réflexion concernant la source à l'origine des risques. Les listes des risques pour différents secteurs et types de projets sont disponibles publiquement.

Livraison des produits

Lors de ce processus, les produits acceptés sont livrés ou transférés aux parties prenantes de l'entreprise concernées. Un accord officiel sur les produits finis documente la réussite du sprint.

Livraison en itération

Une livraison en itération consiste en la livraison échelonnée de la valeur ajoutée au client.

Matrice d'organisation des ressources

La matrice d'organisation des ressources est une représentation hiérarchique combinant la structure d'organisation fonctionnelle et la structure d'organisation du projet. La matrice d'organisation réunit les membres de l'équipe d'un projet venus de différents départements tels que l'informatique, la finance, le marketing, les ventes, la production et d'autres départements, pour créer des équipes polyvalentes.

Matrice des compétences exigées

La matrice des compétences exigées, aussi connue sous le nom de cadre de compétences, est utilisée afin d'évaluer les lacunes de compétences et exigences de formation pour les membres de l'équipe. Une matrice des compétences établit les compétences, les capacités et le niveau d'intérêt des membres de l'équipe à utiliser ces compétences et capacités au cours d'un projet. L'utilisation de cette matrice permet à l'organisation d'évaluer toute lacune de compétence chez les membres de l'équipe et d'identifier les employés qui auront besoin d'une formation complémentaire pour une compétence ou dans un domaine particulier.

Meilleure coordination d'équipe

La réunion Scrum of Scrums facilite la coordination du travail à travers plusieurs équipes Scrum. Cela est particulièrement important lorsque des tâches impliquent des dépendances entre équipes. Les incompatibilités et différences entre le travail et les produits de différentes équipes sont rapidement exposées. Ce forum permet également aux équipes de présenter leurs progrès et d'offrir leur feedback aux autres équipes.

Mêlée quotidienne

La mêlée quotidienne est une réunion quotidienne courte, limitée à 15 minutes. Les membres de l'équipe se retrouvent pour rendre compte de leurs progrès et répondre aux trois questions suivantes :

1. Qu'ai-je fait depuis la dernière réunion ?
2. Qu'ai-je prévu de faire d'ici à la prochaine réunion ?
3. Le cas échéant, à quels blocages et obstacles fais-je face en ce moment ?

Membres du Scrum Guidance Body

Les membres du Scrum Guidance Body (SGB) peuvent inclure les experts Scrum, des Scrum Masters désignés, Product Owners et membres d'équipe (à tous les niveaux). Cependant, il doit y avoir une limite au nombre de membres du SGB afin de s'assurer que celui-ci demeure pertinent et ne devienne pas directif.

Menaces

Les menaces sont des risques qui pourraient affecter le projet de façon négative.

Mesures de suivi et dates d'échéances

Une fois que les améliorations actionnables convenues ont été élaborées et acceptées, l'équipe Scrum peut considérer les mesures à prendre pour mettre en œuvre ces améliorations. Une date d'échéance pour la réalisation de chaque mesure sera définie.

Méthode des 100 points (100-Point Method)

Cette méthode a été développée par Dean Leffingwell et Don Widrig (2003). Elle consiste à donner au client 100 points qu'il peut ensuite utiliser pour voter pour les caractéristiques qui sont les plus importantes à ses yeux.

Méthode Wideband Delphi

La méthode Delphi est une technique d'estimation basée sur un groupe pour déterminer le volume de travail requis et le temps nécessaire à sa réalisation. Chaque personne au sein de l'équipe fournit anonymement ses estimations pour chaque caractéristique et les estimations initiales sont tracées sur un diagramme. L'équipe discute ensuite des facteurs qui ont influencé leurs estimations et poursuit avec une seconde vague d'estimations. Ce processus est répété jusqu'à ce que les estimations individuelles soient proches les unes des autres et que l'équipe puisse parvenir à un consensus sur l'estimation finale.

Méthodes de déploiement organisationnelles

Les méthodes de déploiement de chaque organisation ont tendance à varier en fonction du secteur, des utilisateurs cibles et du positionnement. Selon le produit qui doit être livré, le déploiement peut se dérouler à distance ou peut impliquer une livraison physique ou un produit de transition.

Méthodes de préparation de la release (Release Preparation Methods)

Ce sont les méthodes utilisées afin d'exécuter les tâches identifiées dans le plan de préparation de la release de façon à ce que les incréments de produit soient prêts à être expédiés/mis à disposition.

Méthodes de priorisation de la release

Ces méthodes sont utilisées pour développer un plan de release. Elles sont spécifiques au secteur et à l'organisation et sont normalement déterminées par des cadres supérieurs au sein de l'organisation.

Mise à jour du Sprint Backlog

Durant ce processus, l'équipe Scrum principale tient des réunions de planification de sprint pendant lesquelles le groupe crée un backlog de sprint contenant toutes les tâches à accomplir pendant le sprint.

Mises à jour rejetées des recommandations du Scrum Guidance Body

Les améliorations recommandées du Scrum Guidance Body ne sont pas toujours acceptées. Si l'amélioration recommandée n'est pas acceptée par les membres du Scrum Guidance Body, le feedback expliquant les raisons du rejet est fourni aux parties concernées.

Mission de l'entreprise

La mission de l'entreprise offre un cadre pour la formulation des stratégies d'une entreprise ou organisation qui guide le processus de décision global.

Neutre au risque

Neutre au risque est l'une des catégories d'utilité fonction et fait référence à une partie prenante qui n'est ni opposée aux risques ni risquophile. Aucune décision n'est affectée par le degré d'incertitude du résultat. Quand deux scénarios possibles comportent les mêmes niveaux d'avantages, la partie prenante neutre au risque ne se préoccupera pas du fait qu'un scénario soit plus risqué que l'autre.

Nombre de stories

Fait référence au nombre de stories réalisées au cours d'un même sprint. Ce nombre peut être exprimé de façon simple ou pondérée.

Normes du secteur

Les nouvelles normes de secteur, ou des modifications des normes existantes doivent être prises en compte de façon à maintenir un produit ou service viable. Les user stories doivent donc être incluses au backlog de programme et/ou de portefeuille priorisé et hiérarchisées en conséquence.

Notes de release (Release Notes)

Les notes de release incluent les critères d'expédition externes ou adaptés au marché pour le produit à livrer.

Obstacle

Un obstacle est tout blocage ou barrière qui réduit la productivité de l'équipe Scrum.

Opportunités

Les risques qui auront probablement un impact positif sur le projet sont considérés comme des opportunités.

Organisation des mêlées quotidiennes

Ce processus consiste à tenir une réunion quotidienne très ciblée et limitée dans le temps. Cette réunion est appelée mêlée quotidienne et constitue un forum permettant aux membres de l'équipe Scrum de se tenir informés de leurs progrès et des obstacles auxquels ils font face.

Outils de suivi de sprint

Les outils de suivi de sprint sont utilisés pour suivre les progrès d'un sprint et pour savoir où en est l'équipe Scrum dans la réalisation des tâches contenues dans le backlog de sprint. Il existe différents outils permettant de suivre la progression d'un sprint mais le plus courant est le scrumboard, aussi connu sous le nom de tableau des tâches (task board) ou de tableau d'évolution.

Outils informatiques automatisés

Les outils informatiques automatisés sont utilisés pour la planification, la collecte d'informations et la distribution.

Parties prenantes de l'entreprise

Un terme collectif qui inclue les clients, utilisateurs et le sponsor qui interagissent régulièrement avec le Product Owner, le Scrum Master et l'équipe Scrum afin de contribuer et de faciliter la création du produit, du service ou d'un autre résultat du projet.

Personnages

Ce sont des personnages fictifs très détaillés, représentant la majorité des utilisateurs ainsi que les autres parties prenantes de l'entreprise qui ne verront pas forcément directement le produit fini. Les personnages sont créés pour identifier les besoins des principaux utilisateurs cibles.

Phase d'exécution

C'est la dernière phase de la formation d'une équipe, lorsque celle-ci est plus cohésive et opère à son plus haut niveau de performance. Les membres ont évolué en une équipe efficace de pairs professionnels uniformément productifs.

Phase d'initiation

Cette phase comprend les processus relatifs à l'initiation d'un projet : création d'une vision de projet, identification du Scrum Master et des parties prenantes de l'entreprise , formation de l'équipe Scrum, développement des epics, création d'un backlog de produit priorisé et planification de la release.

Phase de formation

La première phase de la formation d'une équipe, souvent considérée comme une étape ludique du fait que tout est nouveau et que l'équipe n'a pas encore rencontré de difficultés avec le projet.

Phase de normalisation

Troisième phase de la formation de l'équipe. Celle-ci commence à murir, est capable de résoudre les différents internes et de trouver des solutions pour travailler ensemble. C'est une période d'ajustement.

Phase de planification et d'estimation

Cette phase comporte les processus relatifs à la planification et à l'estimation des tâches, notamment la création des user stories, l'approbation, l'estimation et la validation des user stories, la création et l'estimation des tâches ainsi que la Mise à jour du Sprint Backlog.

Phase d'implémentation

La phase d'implémentation inclut tous les processus relatifs à l'exécution de tâches et des activités pour créer le produit d'un projet.

Phase d'orage

C'est la deuxième phase lors de la formation d'une équipe lorsque celle-ci commence à essayer d'effectuer le travail. Cependant, des luttes de pouvoir peuvent avoir lieu et il y a souvent du chaos ou de la confusion au sein de l'équipe.

Planification de la qualité

Fait référence à l'identification et à la définition du produit qui doit être créé à l'issue d'un sprint et à l'issue du projet ainsi qu'aux critères d'acceptation, méthodes de développement à suivre et responsabilités clés des membres de l'équipe Scrum au sujet de la qualité.

Planification de la release

Durant ce processus, l'équipe Scrum principale passe en revue les user stories de niveau élevé dans le backlog de produit priorisé afin de développer un calendrier de planification de la release. Ce dernier consiste principalement en un calendrier de déploiement échelonné qui peut être partagé avec parties prenantes de l'entreprise. La durée des sprints est également décidée au cours de ce processus.

Planification de la valeur

Planifier la valeur implique de justifier et de confirmer la valeur du projet. La responsabilité de déterminer comment la valeur ajoutée est créée incombe aux parties prenantes de l'entreprise (sponsor, clients et/ou utilisateurs), tandis que l'équipe Scrum se concentre sur ce qui doit être développé.

Planning poker

Le planning poker ou poker d'estimation est une technique d'estimation qui combine réflexion de groupe et réflexion individuelle afin d'estimer la taille relative des user stories ou les efforts requis pour les développer.

Politiques de l'entreprise

Les politiques de l'entreprise sont une série de principes, règles et directives formulés ou adoptés par une organisation. Un changement dans les politiques de l'entreprise pourrait affecter les user stories puisque celles-ci ont été créées en fonction des politiques existantes.

Portée

La portée d'un projet est la somme totale de tous les incréments de produit et du travail requis pour développer le produit fini.

Portefeuille

Un portefeuille est un groupe de programmes associés dans le but de livrer des résultats opérationnels tels qu'ils sont définis dans l'énoncé de vision du portefeuille. Le backlog de portefeuille priorisé incorpore le backlog de programme priorisé pour tous les programmes du portefeuille.

Priorisation

Prioriser consiste à déterminer l'ordre des choses et à séparer ce qui doit être fait maintenant de ce qui peut être fait plus tard.

Priorisation basée sur la valeur client

La priorisation basée sur la valeur client accorde la plus haute importance au client et cherche à réaliser les user stories ayant la plus forte valeur en premier. Ces user stories à forte valeur sont identifiées et remontées en haut du backlog de produit priorisé.

Priorisation des risques

La priorisation des risques est une étape importante de la gestion des risques. Elle implique de prioriser les risques devant être intégrés au backlog de produit priorisé pour une action spécifique.

Priorisation MoSCoW

La méthode de priorisation MoSCoW tient son nom des premières lettres des expressions suivantes : « Must have » (« doit être fait », vital) , « Should have » (« devrait être fait », essentiel), « Could have » (« devrait être fait, si possible », confort) et « Won't have » (« impossible », luxe). Ces étiquettes sont un ordre de priorité décroissant avec « Must have » représentant les caractéristiques sans lesquelles le produit n'aura aucune valeur, et « Won't have » les caractéristiques qui, bien qu'elles seraient les bienvenues, ne sont pas nécessaires.

Problèmes

Les problèmes sont généralement des certitudes spécifiques qui sont actuellement présentes dans le projet, il n'y a donc pas besoin d'effectuer une évaluation de la probabilité comme c'est le cas pour les risques.

Problèmes résolus

Lors des réunions Scrum of Scrum, les membres de l'équipe Scrum ont l'opportunité de discuter ouvertement des problèmes qui impactent leur projet. La discussion et la résolution des problèmes lors de la réunion Scrum of Scrum améliore grandement la coordination entre les différentes équipes Scrum et contribue à réduire le besoin de reconception et de reprise.

Product Owner

Le Product Owner est la personne responsable de maximiser la valeur commerciale du projet. Il est chargé d'exprimer les exigences du client et de soutenir la justification commerciale du projet.

Product Owner du portefeuille

Le Product Owner du portefeuille définit les objectifs stratégiques et les priorités du portefeuille.

Product Owner du programme

Le Product Owner du programme définit les objectifs stratégiques et les priorités du programme.

Produit

Dans le Guide SBOK®, le terme « produit » réfère à un produit, un service ou un autre bien livrable qui offre de la valeur ajoutée au client.

Produit fini

Cette version est le produit livrable final pour lequel le projet a été approuvé.

Produits acceptés

Les produits qui remplissent les critères d'acceptation des user stories sont acceptés par le Product Owner. Ceux-ci sont considérés comme des produits acceptés et peuvent être livrés au client s'il le désire.

Produits du projet potentiellement livrables

Les produits du projet potentiellement livrables sont des apports précieux pour la coordination au niveau du programme ou du portefeuille. À la fin des sprints des projets, les incréments de produit sont terminés. Les user stories incluses dans ces incréments sont conformes à la définition de terminé ainsi qu'à leurs critères d'acceptation respectifs.

Produits livrables rejetés

Sont des produits livrables qui ne remplissent pas les critères d'acceptation définis. Une liste des produits livrables rejetés est maintenue et mise à jour après chaque réunion de revue de sprint et contient un inventaire de tous les produits ayant été rejetés.

Programme

Un programme est un groupe de projets associés dans le but de livrer des résultats opérationnels tels qu'ils sont définis dans l'énoncé de vision du programme. Le backlog de programme priorisé incorpore les backlogs de produits priorisés pour tous les projets du programme.

Programme de consolidation d'équipe

Puisqu'une équipe Scrum est polyvalente, chaque membre doit participer activement à tous les aspects du projet. Le Scrum Master doit pouvoir identifier les problèmes potentiels auxquels les membres de l'équipe pourraient avoir à faire face et doit essayer de les résoudre consciencieusement dans le programme de consolidation d'équipe afin de maintenir une équipe efficace.

Progression continue

La progression continue est une approche de Scrum dans laquelle les membres de l'équipe tirent les enseignements de leur expérience et de l'implication des parties prenantes de l'entreprise afin de maintenir le backlog de produit priorisé constamment à jour en tenant compte de la modification des exigences.

Projet

Un projet est une initiative collaborative visant soit à créer de nouveaux produits ou services soit à fournir des résultats tels qu'ils ont été définis dans l'énoncé de vision du projet. En général, les projets sont affectés par des contraintes de temps, de coûts, de portée, de qualité, de personnel et de capacité organisationnelle.

Propension au risque

La propension au risque fait référence au degré d'incertitude qu'une partie prenante ou organisation est prête à accepter.

Qualité

La qualité est définie comme la capacité à terminer un produit qui réponde aux critères d'acceptation et réalise la valeur commerciale attendue par le client.

Quatre questions par équipe

Une série de quatre questions posées lors de chaque réunion Scrum of Scrums (SoS). Chaque représentant d'équipe Scrum fournit des informations sur son équipe en répondant à quatre questions spécifiques.

1. Sur quoi mon équipe a-t-elle travaillé depuis notre dernière réunion ?
2. Que va faire mon équipe d'ici à la prochaine réunion ?
3. Quelles sont les choses que d'autres équipes attendent de mon équipe et qui restent à faire ?
4. Qu'est-ce que mon équipe prévoit de faire qui pourrait affecter les autres équipes ?

Registres de rétrospective de sprint

Le registre de rétrospective de sprint contient une trace de toutes les opinions, discussions et mesures à prendre abordées lors d'une réunion de rétrospective du sprint. Le Scrum Master peut faciliter la création de ce registre avec l'aide des membres de l'équipe Scrum principale.

Regroupement (Colocation)

Le regroupement consiste à rassembler tous les membres de l'équipe Scrum principale sur un même lieu de travail afin de tirer avantage d'une meilleure coordination, et résolution de problèmes ainsi que d'un partage des connaissances et d'un apprentissage amélioré.

Remaniement

Le remaniement est un outil spécifique pour les projets de logiciels. Le but de cette technique est d'améliorer la maintenabilité du code existant, de le simplifier, de le rendre plus court et plus flexible. Remanier implique d'améliorer la conception du code existant sans changer la façon dont celui-ci se comporte. Cela implique :

1. d'éliminer les codes redondants et répétitifs
2. de diviser les méthodes et fonctions en de plus petites routines
3. de définir clairement les variables et les noms de méthodes
4. de simplifier la conception du code
5. de rendre le code plus facile à comprendre et à modifier

Représentants de l'équipe Scrum

Un représentant est nommé par l'équipe pour la représenter lors des réunions Scrum of Scrums (SoS) en fonction de qui est le mieux à même de remplir ce rôle en et des circonstances et problèmes actuels.

Représentation des stories

La représentation des stories est une technique offrant un aperçu visuel d'un produit et de ses éléments principaux. La représentation des stories, formulée pour la première fois par Jeff Patton (2005), est couramment utilisée pour illustrer les stratégies produits. La représentation des stories illustre la séquence de développement des itérations de produit et établit quelles caractéristiques doivent être incluses dans la première version, la deuxième, la troisième et dans les versions suivantes.

Ressources partagées

Les ressources partagées peuvent inclure les personnes, environnements et équipements nécessaires à toutes ou certaines des équipes Scrum travaillant sur le projet. Dans le cas d'un projet important, les ressources partagées peuvent être limitées et requises par toutes ou certaines des équipes Scrum au même moment.

Résultats des évaluations/du benchmarking

L'évaluation ou le benchmarking contribuent à l'établissement d'une norme minimale lors de la conception d'un produit ou d'un service et conduisent à une modification de la définition de terminé. Ils peuvent parfois fournir une impulsion conduisant le Product Owner du programme ou du portefeuille à développer de nouvelles user stories afin de mettre en œuvre les meilleures pratiques.

Retour sur capital investi (RCI)

Le retour sur capital investi (RCI), lorsqu'il est utilisé pour justifier un projet, évalue les bénéfices nets attendus du projet. Il est calculé en déduisant les coûts ou investissements prévus pour le projet des bénéfices attendus puis en déduisant le résultat (profit net) par les coûts prévus afin d'obtenir un taux de rendement.

Rétrospective de sprint

Le Scrum Master et l'équipe Scrum se réunissent pour discuter des enseignements tirés au cours du sprint. Les enseignements sont documentés et peuvent être appliqués à de futurs sprints.

Rétrospective de release

Ce processus conclut le projet. Parties prenantes de l'entreprise de l'organisation et l'équipe Scrum principale se rassemblent afin d'effectuer une Rétrospective de release et d'identifier, de documenter et d'assimiler les enseignements du projet. Généralement, ces leçons mènent à la documentation d'améliorations actionnables convenues à mettre en place au cours des projets à venir.

Réunion de planification de sprint

Une réunion de planification de sprint est organisée au début d'un sprint et fait partie du processus de Mise à jour du Sprint Backlog. Cette réunion est limitée à une durée de 8 heures pour un sprint d'un mois et est divisée en deux parties : la définition des objectifs et l'estimation des tâches.

Réunion de planification des environnements

Une réunion de planification des environnements sert à définir un calendrier de partage des environnements pour l'équipe Scrum.

Réunion de rétrospective de programme ou de portefeuille

La réunion de rétrospective de programme ou de portefeuille est similaire à la réunion de Rétrospective de release mais est organisée au niveau du programme ou du portefeuille. La principale différence est que la fréquence des réunions de rétrospective de programme ou de portefeuille est beaucoup plus faible que celle des réunions de Rétrospective de release.

Réunion de Rétrospective de release

La réunion de Rétrospective de release est une réunion permettant de déterminer de quelle façon la collaboration au sein de l'équipe et l'efficacité peuvent être améliorées pour les futurs projets. Les points positifs, négatifs et les opportunités d'amélioration sont également abordés. La durée de cette réunion n'est pas limitée dans le temps et celle-ci peut se dérouler en personne ou virtuellement.

Réunion de rétrospective du sprint

La réunion de rétrospective du sprint est limitée à 4 heures pour un sprint d'un mois et se tient dans le cadre du processus de rétrospective du sprint. Sa durée peut être augmentée ou réduite en fonction de la durée du sprint. Pendant cette réunion, l'équipe Scrum se rassemble afin de passer en revue et de réfléchir au sprint passé en termes des processus suivis, des outils employés, des stratégies de collaboration et de communication et d'autres aspects pertinents au projet.

Réunion de revue de sprint

La réunion de revue de sprint est limitée dans le temps, elle dure 4 heures pour un sprint d'un mois. Sa durée peut être ajustée en fonction de la durée du sprint. Durant la réunion de revue de sprint, l'équipe Scrum présente les incréments de produits du sprint en cours au Product Owner qui peut soit les accepter ou les rejeter.

Réunion de revue du backlog de produit priorisé

Une réunion de revue du backlog de produit (aussi connue sous le nom de session d'affinage des besoins du backlog de produit priorisé) est une réunion formelle tenue durant le processus d'affinage du Product Backlog priorisé qui aide l'équipe Scrum à passer en revue et à parvenir à un consensus au sujet du backlog.

Réunion de revue du backlog de programme ou de portefeuille priorisé

Au niveau du programme ou du portefeuille, chaque projet est représenté dans le programme et chaque programme est représenté dans le portefeuille. Afin de rationaliser la réunion, il est recommandé de n'avoir qu'un représentant présent pour chaque projet ou programme au niveau du programme ou du portefeuille.

Réunion de risques

Les risques peuvent être plus facilement priorisés par le Product Owner en organisant une réunion avec l'équipe Scrum principale et éventuellement en invitant parties prenantes de l'entreprise concernées à y participer.

Réunion de vision du projet

C'est une réunion entre les parties prenantes du programme, le Product Owner du programme, le Scrum Master du programme et le Chief Product Owner. Elle aide à identifier le contexte commercial, les exigences commerciales et les attentes des parties prenantes de l'entreprise de façon à développer un énoncé de vision du projet efficace.

Réunion d'environnement

Cette réunion est organisée afin d'identifier le type et le nombre d'environnements requis pour développer, gérer et tester les produits du projet. Les ressources nécessaires afin d'établir les environnements requis sont également abordées au cours de cette réunion.

Réunion Scrum of Scrums

La réunion Scrum of Scrums (SoS) est importante lorsque l'approche Scrum est mise à l'échelle de projets plus larges. Les représentants de toutes les équipes y assistent. Normalement, cette réunion est facilitée par le Chief Scrum Master et est censée se concentrer sur les points de coordination et d'intégration entre les différentes équipes Scrum. Cette réunion se déroule à intervalles prédéterminés ou lorsque les équipes Scrum en ont le besoin.

Réunions de groupe d'utilisateurs

Les réunions de groupes d'utilisateurs impliquent parties prenantes de l'entreprise concernées, les utilisateurs ou clients principaux du produit. Ils offrent à l'équipe Scrum principale des renseignements directs au sujet des attentes des utilisateurs. Cela aide à formuler les critères d'acceptation du produit et offre des informations importantes pour le développement des epics.

Réunions de groupe thématique

Les groupes thématiques rassemblent des personnes lors d'une séance guidée pour partager leurs opinions, perceptions ou notes sur un produit, service ou un résultat désiré. Les membres du groupe thématique sont libres de se poser des questions et de demander des clarifications sur certains sujets ou concepts. Les questionnements, critiques constructives et le feedback permettent aux groupes thématiques d'atteindre une meilleure qualité de produit et contribuent donc à répondre aux attentes des clients.

Réunions du Scrum Guidance Body

Le Scrum Guidance Body se réunit régulièrement afin d'aborder le besoin potentiel pour la mise à jour des recommandations du Scrum Guidance Body (p. ex. amélioration recommandées à partir des rétrospectives et autres processus, mise à jour de la législation, etc.). La fréquence des réunions est décidée par le Scrum Guidance Body en fonction des besoins spécifiques de l'entreprise.

Risk Burndown Chart

Un diagramme représentant la sévérité cumulée des risques d'un projet dans le temps. La probabilité des différents risques est représentée au-dessus les unes des autres afin de montrer les risques cumulés sur l'axe y. L'identification et l'évaluation initiale des risques ainsi que la création du risk burndown chart intervient relativement tôt au cours du projet.

Risque

Le risque est défini comme un événement ou une série d'événements incertains pouvant affecter les objectifs d'un projet et contribuer à sa réussite comme à son échec.

Risques atténués

Les risques atténués sont les risques qui sont gérés ou atténués avec succès par l'équipe Scrum au cours du projet.

Risques du programme et du portefeuille

Ce sont les risques liés à un portefeuille ou programme qui auront également un impact sur les projets faisant partie du portefeuille ou du programme en question.

Risquophile (Risk Seeking)

Risquophile est une des catégories d'utility function. Elle fait référence à une partie prenante qui est prête à accepter un risque même si celui-ci n'offre qu'une augmentation marginale en termes de rendement ou de bénéfices pour le projet.

Rôles non-essentiels

Les rôles qui ne sont pas obligatoirement requis pour le projet Scrum. Ces rôles peuvent inclure des membres de l'équipe qui sont intéressés par le projet, n'ont pas de rôle défini dans l'équipe de projet, peuvent interagir avec l'équipe mais ne sont pas responsables de la réussite du projet.

Rôles principaux

Les rôles principaux sont les rôles qui sont obligatoirement requis pour la production du produit du projet, qui sont investis dans le projet et sont fondamentalement responsables de la réussite de chaque sprint du projet et du projet dans son ensemble.

Rythme soutenable

Un rythme soutenable est le rythme auquel l'équipe peut travailler tout en étant capable de le maintenir confortablement. Il se traduit par une augmentation de la satisfaction des employés, de la stabilité et une meilleure précision des estimations, le tout conduisant finalement à une augmentation de la satisfaction client.

Schéma de conception

Les schémas de conception sont une solution formelle de suivi pour la résolution des problèmes de conception dans un domaine d'expertise spécifique. Ces schémas assurent le suivi à la fois du procédé utilisé et de la résolution effective et peuvent ensuite être réutilisés afin d'améliorer le processus de décision et de production.

Schémas simplifiés

Les schémas simplifiés impliquent de catégoriser les éléments en fonction de leur degré de priorité, c-à-d « 1 », « 2 », « 3 » ou « Forte », « Moyenne » et « Faible », etc. Bien que cette approche soit simple et directe, elle est souvent problématique car on observe souvent une tendance à organiser tous les éléments dans la catégorie de priorité « 1 » ou « Forte ».

Scrum Guidance Body

Le Scrum Guidance Body (SGB) est un rôle facultatif. Il se compose généralement d'un groupe de documents et/ou d'un groupe d'experts qui sont normalement impliqués dans la définition des objectifs relatifs à la qualité, aux législations gouvernementales, à la sécurité et à d'autres paramètres organisationnels clés.

Scrum Master

Le Scrum Master est l'un des rôles de l'équipe Scrum principale. Il facilite la création d'incrément de produits, gère les risques, changements et obstacles durant les mêlées quotidiennes, la rétrospective de sprint et les autres processus Scrum.

Scrum Master du portefeuille

Le Scrum Master du portefeuille résout les problèmes, élimine les obstacles, facilite et gère les réunions pour le portefeuille.

Scrum Master du programme

Le Scrum Master du programme résout les problèmes, élimine les obstacles, facilite et gère les réunions pour le programme.

Scrumboard

Le scrumboard est un outil utilisé par l'équipe Scrum afin de planifier et de suivre les progrès au cours de chaque sprint. Le scrumboard est composé de quatre colonnes pour indiquer les progrès des tâches estimées pour le sprint. Une colonne « à faire » pour les tâches qui ne sont pas encore commencées, une colonne « en cours » pour les tâches commencées mais pas encore finies, une colonne « à tester » pour les tâches finies mais en cours de test et une colonne « terminé » pour les tâches terminées et testées avec succès.

Sessions de planification de la release (Release Planning Sessions)

L'objectif principal des sessions de planification de la release est de créer un calendrier de planification de la release afin d'avoir un aperçu de la release et des livraisons pour le produit développé de façon à harmoniser les attentes du Product Owner et des parties prenantes de l'entreprise concernées.

Sessions JAD

Une session Joint Application Design (JAD) permet de déterminer les besoins. C'est un atelier facilité et très structuré qui accélère le processus de création de la vision de projet et permet aux parties prenantes de l'entreprise et aux autres décideurs de parvenir à un consensus sur la portée, les objectifs et autres spécifications du projet.

Seuil de risque

Le seuil de risque fait référence au niveau de risque acceptable pour la partie prenante ou l'organisation. Un risque se trouve soit au-dessus soit en dessous du seuil de risque. S'il est en dessous, la partie prenante ou l'organisation acceptera le risque plus facilement.

Spécialisation de l'équipe

Dans le cas d'un projet important, la spécialisation de l'équipe peut s'avérer nécessaire. Il existe trois dimensions à la spécialisation de l'équipe. La première est le besoin de réaliser des tâches spécifiques. La deuxième est le besoin de compétences spécifiques pour des membres uniques de l'équipe. La troisième est qu'il peut y avoir des limites à la flexibilité de l'équipe.

Speed Boat

La technique du speed boat peut être utilisée pour mener la réunion de rétrospective du sprint. Les membres de l'équipe jouent le rôle de l'équipage d'un speed boat. Le bateau doit atteindre une île qui est symbolique de la vision du projet. Des feuillets autocollants sont utilisés par les personnes présentes afin de noter les moteurs et les obstacles. Les moteurs sont les éléments qui les aident à atteindre l'île tandis que les obstacles sont les éléments qui les empêchent de l'atteindre. Cet exercice est limité dans le temps et ne doit prendre que quelques minutes.

Sponsor

Le sponsor est l'individu ou l'organisation qui offre les ressources et soutient le projet. C'est aussi la partie prenante à qui tout le monde doit rendre compte à la fin du projet.

Sprint

Un sprint est une itération limitée dans le temps (entre une et quatre semaines) durant laquelle l'équipe Scrum travaille à la création des incréments de produit.

Sprint burndown chart

Le Sprint Burndown Chart est un diagramme qui représente la quantité de travail restant à effectuer durant le sprint en cours.

Sprint de préparation de la release (Release Readiness Sprint)

Si des tâches spécifiques doivent être effectuées pour préparer la release et pour garantir que les exigences minimales ont été remplies, ces tâches sont effectuées lors d'un sprint de préparation de release. Un sprint de préparation de la release, s'il est nécessaire, n'est effectué qu'une fois par release et c'est le dernier sprint avant la livraison.

Stratégie de collaboration

La collaboration est un élément essentiel de l'approche Scrum et la stratégie de collaboration définit comment les décideurs, parties prenantes de l'entreprise et les membres de l'équipe s'impliquent et collaborent entre eux.

Stratégie de collaboration des équipes Scrum

La stratégie de collaboration des équipes Scrum définit comment les différentes équipes Scrum collaborent ensemble de façon à offrir la meilleure valeur ajoutée en un délai le plus court possible.

Stratégie de collaboration des Product Owners

La stratégie de collaboration des Product Owners définit comment plusieurs Product Owners collaborent avec le Chief Product Owner.

Stratégie de communication

Cette stratégie précise les rapports qui doivent être créés et maintenus durant le projet. Plusieurs méthodes différentes sont utilisées afin de communiquer aux parties prenantes de l'entreprise les informations importantes relatives au projet. La stratégie de communication définit ces méthodes et détermine la personne responsable des différentes activités de communication.

Stratégie de ressources humaines de l'entreprise

De façon générale, la stratégie de ressources humaines de l'entreprise fournit des informations sur la disponibilité de certains personnels pour différents projets, programmes et portefeuilles. Elles fournissent aussi des informations relatives à l'embauche du personnel requis pour les efforts à venir.

Stratégie pilote

Une stratégie pilote est utilisée pour planifier un déploiement pilote en détail. La portée et les objectifs du déploiement, les utilisateurs cibles principaux du déploiement, un calendrier de déploiement, les programmes de transition, la préparation requise des utilisateurs, les critères d'évaluation pour le déploiement et d'autres éléments clés relatifs au déploiement sont spécifiés dans la stratégie pilote et partagés avec les parties prenantes de l'entreprise.

Structure de répartition des risques

Dans cette structure les risques sont regroupés en fonction de leur catégories et de leurs points communs. Par exemple, les risques peuvent être catégorisés comme étant d'ordre financier, technique ou relatifs à la sécurité.

Supportive Leader

Un Supportive Leader donne des instructions, supporte et suit les progrès des membres de son équipe en écoutant, aidant, encourageant et présentant une attitude positive dans les moments d'incertitude.

Tableau de probabilité et d'impact (Probability Impact Grid)

Un tableau où les risques sont évalués en fonction de leur probabilité d'occurrence et de leur impact potentiel sur les objectifs du projet. Généralement, un score numérique est attribué indépendamment à la probabilité et à l'impact. Les deux valeurs sont ensuite multipliées pour obtenir un score de sévérité de risque qui peut être utilisé pour prioriser les risques.

Task-Oriented Leader

Un task-oriented leader s'assure que les tâches sont terminées tout en respectant les échéances.

Taux de rendement interne (TRI)

Le taux de rendement interne (TRI) est un taux de réduction sur un investissement où la valeur des flux entrants actuels est comparée à celle des flux sortants afin d'évaluer le taux de rendement d'un projet. Lorsque des projets sont comparés, un projet avec un TRI élevé est normalement meilleur.

Théorie X

La théorie X repose sur le principe que les employés sont naturellement peu motivés et évitent le travail lorsque c'est possible. Cette théorie induit un style de leadership autoritaire.

Théorie Y

La théorie Y repose sur le principe que les employés sont motivés et cherchent à accepter des responsabilités plus importantes. Cette théorie induit un style de leadership plus participatif.

Time-boxing

Le time-boxing fait référence à la mise en place de courtes durées de temps pour effectuer le travail. Si le travail effectué reste incomplet à la fin de la time-box, il est transféré à la time-box suivante. Les time-boxes offrent la structure nécessaire aux projets Scrum qui possèdent un élément d'incertitude, sont de nature dynamiques et sujets à des changements fréquents.

Tolérance aux risques

La tolérance aux risques indique le degré, la quantité ou le volume de risques que les parties prenantes de l'entreprise sont prêtes à supporter.

Transparence

La transparence permet à toutes les facettes de tout projet Scrum d'être observées par tous. Le partage de toutes les informations permet d'établir un environnement de confiance mutuelle.

Trois questions quotidiennes

Trois questions posées lors des mêlées quotidiennes facilitées par le Scrum Master. Chaque membre de l'équipe Scrum donne des informations en répondant à trois questions spécifiques :

1. Qu'ai-je fait depuis la dernière réunion ?
2. Qu'ai-je prévu de faire d'ici à la prochaine réunion ?
3. Le cas échéant, à quels blocages et obstacles fais-je face en ce moment ?

User stories

Les user stories répondent à une structure spécifique prédéfinie et sont une façon simplifiée de documenter les exigences et fonctionnalités souhaitées pour l'utilisateur. Les exigences exprimées dans les user stories sont des énoncés courts, simples et faciles à comprendre qui permettent une meilleure communication entre les parties prenantes de l'entreprise et une meilleure estimation par l'équipe.

Utilisateur

Les utilisateurs sont les individus ou l'organisation qui utilisent directement le produit, service ou autre résultat du projet. Comme pour les clients, pour chaque organisation, il peut y avoir à la fois des utilisateurs internes ou externes. Dans certains cas, les clients et les utilisateurs sont les mêmes.

Utility function

Utility function est un modèle utilisé pour mesurer l'attitude ou les préférences des parties prenantes de l'entreprise face aux risques. Il définit le niveau ou la propension des parties prenantes de l'entreprise à accepter les risques.

Valeur actuelle nette (VAN)

Une méthode utilisée pour déterminer la valeur actuelle nette d'un bénéfice financier à venir en tenant compte d'un taux d'inflation ou d'intérêt supposé.

Valeur monétaire attendue

Technique d'évaluation des risques où le potentiel impact financier d'un risque est déterminé sur la base de sa valeur monétaire attendue (VMA). La VMA est calculée en multipliant l'impact monétaire par la probabilité du risque, telle qu'elle a été estimée par le client.

Validation des user stories

Ce processus consiste en la validation par l'équipe Scrum des user stories approuvées par le Product Owner pour un sprint. Ce processus produit des user stories validées.

Vélocité de sprint

La vélocité de sprint mesure la capacité de l'équipe à réaliser le travail d'un sprint. Elle est généralement exprimée dans la même unité que celles utilisées pour les estimations, normalement en story points ou en durée idéale.

Vision de l'entreprise

Comprendre la vision de l'entreprise aide à maintenir l'axe du projet sur les objectifs de l'organisation et le futur potentiel de l'entreprise. Le Product Owner peut s'inspirer des conseils et directives de la vision de l'entreprise afin de créer l'énoncé de vision du projet.

Voix du client (Voice of the Customer ou VOC)

La voix du client (VOC) fait à la fois référence aux exigences explicites et implicites du client, qui doivent être comprises avant de concevoir le produit ou le service. Le Product Owner représente la voix du client.

War room

La War Room (ou salle de décision) est le terme normalement utilisé pour décrire l'endroit où sont installés tous les membres de l'équipe Scrum travaillant sur le projet. En général, le lieu est conçu de façon à ce que les membres de l'équipe puissent se déplacer librement et communiquer facilement en étant tous proches les uns des autres.

INDEX

- Accord sur les produits finis, 254
- Actual cost, 79
- Adaptability, 4
- Adaptation, 24
- Raffinage du Product Backlog priorisé, 216, 229
 - contributions, 230
 - outils, 232
 - résultats, 233
- Agile Expert Certified (AEC®), 6
- Agile Manifesto, 29
- Améliorations actionnables convenues, 245
- Analyse de Pareto, 125
- Analyse de valeur acquise (EVA), 78
- Analyse des lacunes, 146
- Analyse FFOM, 146
- Analyse Kano, 76
- Appropriation*, 28
- Approve, Estimate, and Commit User Stories*, 17
- Arbres de probabilité, 124
- Argent de Monopoly, 75
- Articulation*, 28
- Aspects, 7
- Assurance Qualité, 95
- Atelier d'user stories, 165
- Atténuation des risques, 128
- Attirant, 76
- Attitude face aux risques, 121
- Attractif, 76
- Auto-organisation, 108
- Aversion aux risques, 121
- Awareness*, 28
- Backlog de produit de programme mis à jour, 231
- Backlog de produit du programme, 144
- Backlog de produit priorisé, 88, 173
- Backlog de produit priorisé affiné, 180
- Backlog de produit priorisé mis à jour, 192, 233, 279, 305, 311
- Backlog de sprint, 212
- Bénéfices du projet, 70
- Besoins commerciaux, 70
- Besoins en personnel, 150
- Brainstorming, 123
- Budget at completion, 79
- Budget du projet, 148
- Business Justification, 12
- Calendrier de l'équipe, 211
- Calendrier de planification de la release, 179
- Calendrier de planification de la release mis à jour, 234
- Calendrier de vacances, 178
- Cartographie de la chaîne de valeur, 75
- Change, 13
- Change Requests, 14
- Changement, 99, 100
- Changements approuvés, 168
- Charte de projet, 148
- Chief Product Owner, 45
- Chief Scrum Master, 48
- Cibler les clients pour la release, 180
- Classement relatif des priorités, 77
- Client, 43
- Clients, 69
- Coaching, 61
- Collaboration, 10, 21, 28
- Collective ownership, 5
- Colocated Teams, 31
- Colocation, 30
- Communication des risques, 129
- Comparaison par paire, 171
- Conduct Daily Standup*, 17
- Conduct Release Planning*, 16

- Confirmation, estimation et validation des user stories, 184
 - contributions, 194, 200
 - outils, 195, 201
 - résultats, 198, 201
- Confirmer la réalisation des bénéfiques, 71, 82
- Conseils experts RH, 152, 157
- Continuous delivery of value, 4
- Continuous feedback, 4
- Continuous improvement, 4, 13
- Contrat de "joint-venture", 163
- Contrat de développement par palier, 163
- Contrat de livraison graduelle, 163
- Contrat d'incitation et de pénalité, 163
- Contrats applicables, 163
- Contrôle qualité, 94
- Core roles, 10
- Cost performance index, 79
- Cost variance, 79
- Coût d'opportunité, 70
- Coûts des ressources, 153, 158
- Coûts du personnel, 157
- Coûts du projet, 70
- Create Deliverables*, 17
- Create Prioritized Product Backlog*, 16
- Create Project Vision*, 16
- Create Sprint Backlog*, 17
- Create Tasks*, 17
- Create User Stories*, 17
- Création de la vision de projet, 138, 141, 148, 154, 160, 168, 175, 187, 193, 199, 202, 206, 210, 218, 224, 229, 237, 242, 251, 255, 265, 266, 274, 275, 279, 291, 295, 300, 305, 306, 312
 - contributions, 143
 - outils, 145
 - résultats, 147
- Création des éléments livrables, 216
 - contributions, 219
 - outils, 221
 - résultats, 222
- Création des tâches, 184
 - contributions, 203
 - outils, 203
 - résultats, 205
- Création des user stories, 184
 - contributions, 188, 266, 275, 280, 296, 300, 306, 313
 - outils, 190, 269, 277, 281, 297, 303, 309, 314
 - résultats, 191, 271, 278, 282, 298, 304, 311, 314
- Création du backlog de produit priorisé, 138
 - contributions, 169
 - outils, 171
 - résultats, 173
- Mise à jour du Sprint Backlog, 184
 - contributions, 211
 - outils, 211
 - résultats, 212
- Critères d'acceptation, 88
- Critères d'acceptation des user stories, 192, 207, 271, 279, 304, 311, 314
- Critères d'acceptation des User Stories, 88
- Critères d'acceptation minimums, 91
- Critères de sélection, 152
- Critères d'estimation, 208
- Customer centric, 4
- Cycle PDCA, 96
- Cycle Plan-Do-Check-Act (PDCA), 95
- Cycle Plan-Do-Study-Act, 95
- Daily Standup Meeting, 33
- Décomposition, 203
- Définition de terminé, 90
- Définition de Terminé, 174
- Demandes de changement, 101
- Demandes de changement approuvées, 101, 162
- Demandes de changement non-ratifiées, 162
- Demandes de changement rejetées, 101
- Demonstrate and Validate Sprint*, 18
- Démonstration de faisabilité, 144

- Démonstration et validation du sprint, 236
 - contributions, 238
 - outils, 240
 - résultats, 241
- Démonstrations, 82
- Dépendances, 174, 205
- Dépendances externes, 204
- Dépendances facultatives, 204
- Dépendances internes, 204
- Dépendances obligatoires, 204
- Des réunions de groupe d'utilisateurs, 164
- Détermination des dépendances, 203
- Dette technique, 94
- Develop Epic(s)*, 16
- Développement des epics, 138
 - contributions, 161
 - outils, 164
 - résultats, 167
- Développement itératif, 104
- Diagramme de flux cumulatif (DFC), 81
- Diagramme de flux de données
 - Phase d'initiation, 181
 - Phase de planification et d'estimation, 214
 - Phase de release, 259
 - Phase d'implémentation, 234
- Disponibilités et engagements du personnel, 151
- Distributed Teams, 31
- Durée du Sprint, 111, 179
- Earned value, 79
- Échéancier du projet, 70
- Effective Deliverables, 4
- Efficacité d'estimation, 245
- Efficient development process, 4
- Éléments du backlog de produit (PBIs), 114
- Éléments livrables du sprint, 222
- Éléments non-fonctionnels proposés, 246
- Empirical Process Control, 9, 21, 22
- Énoncé de vision du projet, 147
- Enseignements tirés par l'équipe Scrum, 246
- Entretiens d'utilisateurs ou de clients, 165
- Epics, 167
- Équipe Scrum, 41, 50, 69
 - Taille, 52
- Équipe Scrum identifiée, 158
- Équipe Scrum motivée, 228
- Équipe Scrum principale, 105, 161
- Estimate at completion, 79
- Estimate Tasks*, 17
- Estimate to complete, 79
- Estimation d'affinité, 197
- Estimation des tâches, 184
 - contributions, 207
 - outils, 208
 - résultats, 209
- Estimations, 174
- Étude de cas, 71
- Étude de cas du projet, 143
- Étude de marché, 145
- Évaluation des risques, 123
- Exigences commerciales, 170
- Expérience de la journée de travail précédente, 225
- Expérimentation des risques, 129
- Expert Scrum Master (ESM®), 6
- Expertise de l'équipe, 221
- Expertise du Scrum Guidance Body, 166, 173, 191, 240, 245, 257
- Expertise pour l'écriture des user stories, 190, 269, 277, 281
- Expertise pour l'écriture d'user stories, 303, 309, 314
- Expertise Scrum Guidance Body, 222
- Explorateur — Acheteur — Vacancier — Prisonnier (ESVP), 244
- Faster problem resolution, 4
- Feedback des pairs, 245
- Fist of five, 196
- Flexibilité, 103
- Flux des incréments du projet, 89
- Form Scrum Team*, 16
- Formation de l'équipe Scrum, 138

- contributions, 155
- outils, 157
- résultats, 158
- Formation et coûts de formation, 152, 157
- Fournisseurs, 43
- Gagnant-gagnant, 59
- Gagnant-perdant, 60
- Gestion de la qualité, 92
- Gestion des conflits, 59
 - Techniques, 59
- Gestion des risques, 122
- Grille de probabilité et d'impact, 125
- Groom Prioritized Product Backlog*, 17
- Haute direction, 106
- High trust environment, 5
- High velocity, 5
- Identification des risques, 122
- Identification du Scrum Master et des parties prenantes de l'entreprise, 138, 148
 - contributions, 150
 - outils, 152
 - résultats, 153
- Identify Scrum Master and Stakeholder(s)*, 16
- Implement, 17
- Implémentation, 215
- Incertitudes, 173
- Incréments de produit rejetés, 231
- Indicateurs de suivi de sprint, 212
- Indicateurs et méthodes de mesures, 245
- Indifférent, 76
- Informations relatives aux projets précédents, 164
- Initiate, 16
- Initiation, 137
- Innovative environment, 5
- Insatisfaisant, 76
- Inspection, 23
- Intégration continue, 94, 109
- Intégration des changements, 109
- Iterative Development, 10, 22, 35
- Justification commerciale, 65
- Justification de valeur continue, 71, 77
- Justification du projet, 70
- Leader serviteur, 61
- Leadership assertif, 61
- Leadership autocratique, 61
- Leadership déléguant, 61
- Leadership directif, 61
- Leadership laissez-faire, 61
- Leadership de soutien, 61
- Législations et règlements, 163
- Liste des tâches, 205
- Liste des tâches et efforts estimés, 209
- Liste des tâches mise à jour, 209
- Listes de contrôle des risques, 123
- Listes de risques, 123
- Livraison de la valeur ajoutée, 65, 66
- Livraison des produits, 250, 251, 262
 - contributions, 252
 - outils, 254
 - résultats, 254
- Logiciel, 221
- Matrice d'organisation des ressources, 151
- Matrice des compétences exigées, 151
- Mêlée quotidienne, 226
- Mêlées quotidiennes, 216
 - contributions, 225
 - outils, 226
 - résultats, 227
- Mesures de suivi et dates d'échéances, 246
- Méthode des 100 points, 171
- Méthode Wideband Delphi, 196
- Méthodes d'estimation des user stories, 172
- Méthodes de déploiement organisationnel, 254
- Méthodes de priorisation de la release, 179
- Méthodes de priorisation des user stories, 171
- Minimum Marketable Features (MMF), 77
- Modèle de Tuckman sur la dynamique des groupes, 58
- Motivation, 4
- Neutre au risque, 121
- Nombre de stories, 212

- Non-core roles, 11
- Notes du moral de l'équipe, 245
- Organisation, 39
- Organization, 10
- Outils de suivi de sprint, 212
- Participation des parties prenantes de l'entreprise , 55
- Parties prenantes de l'entreprise , 42, 104
- Parties prenantes du programme, 143
- Parties prenantes externes, 92
- Parties prenantes de l'entreprise identifiées, 154
- Parties prenantes internes, 92
- Percent complete, 79
- Perdant-gagnant, 60
- Perdant-perdant, 60
- Personnages, 167
- Personnages mis à jour ou affinis, 193, 272, 279, 282, 305
- Personnel remplaçant, 159
- Phase d'exécution, 58
- Phase de formation, 58
- Phase de normalisation, 58
- Phase de tension, 58
- Plan and Estimate, 17
- Planification de la qualité, 93
- Planification de la release, 138
 - contributions, 176
 - outils, 178
 - résultats, 179
- Planification de la valeur, 74
- Planification et estimation, 183
- Planned value, 79
- Planning Poker, 196
- Polyvalence, 108
- Portée, 86
- Portefeuille, 53
- Portefeuilles
 - Changement, 115
 - Risque, 132
- Principles, 7, 21
- Priorisation basée sur la valeur client, 75, 108
- Priorisation des risques, 127
- Priorisation MoSCoW, 75, 171
- Problèmes, 121
- Procédure de gestion des risques:, 122
- Processes, 7
- Processus d'approbation des changements, 102
- Processus Scrum, 15
- Product, 1
- Product Owner, 6, 10, 41, 44, 68
- Product Owner de programme, 143
- Product Owner du portefeuille, 68
- Product Owner du programme, 68
- Product Owner identifié, 147
- Produits acceptés, 241
- Produits finis, 255
- Produits livrables rejetés, 241
- Programme, 53
- Programme de consolidation d'équipe, 159
- Programmes
 - Changement, 115
 - Risque, 132
- Progrès vers la release/le lancement, 245
- Project, 2
- Project Vision, 2
- Project Vision Statement, 2
- Projet, 53
- Projet pilote, 144
- Propension au risque, 121
- Prototypes, 82
- Pyramide des besoins de Maslow, 63
- Qualité, 85, 86
- Quality, 13
- Questionnaires, 166
- Recommandations du Scrum Guidance Body, 145, 164, 171, 178, 190, 195, 200, 208, 220, 232, 240, 243, 253, 257, 267
- Recommandations mises à jour du Scrum Guidance Body, 247
- Registre des obstacles, 220

- Registres de rétrospective de sprint, 246
- Release, 18, 249
- Releases du produit, 255
- Remaniement, 221
- Représentation des stories, 77
- Requirements churn, 14
- Ressources requises, 156
- Retour sur capital investi (RCI), 73
- Retrospect Project*, 18
- Retrospect Sprint*, 18
- Retrospect Sprint Meeting, 34
- Rétrospective de sprint, 236
 - contributions, 243
 - outils, 243
 - résultats, 245
- Rétrospective de release, 250
 - contributions, 256
 - outils, 257
 - résultats, 258
- Réunion de planification de sprint, 211
- Réunion de Rétrospective de release, 257
- Réunion de rétrospective du sprint, 243
- Réunion de revue de sprint, 240
- Réunion de revue du backlog de produit, 114
- Réunion de risques, 124
- Réunion de vision du projet, 145
- Réunions de groupe thématique, 165, 190
- Réunions de planification de sprint, 203
- Réunions de planification de sprint*, 208
- Réunions de revue du backlog de produit
 - priorisé, 232
- Review and Retrospect, 18
- Revue des notes de feedback, 245
- Revue et rétrospective, 235
- Risk, 14
- Risk Burndown Chart, 130
- Risque, 119, 120, 173
- Risques, 70
- Risques atténués, 223
- Risques du programme et du portefeuille, 162
- Risques identifiés, 168
- Risquophile, 122
- Rôles non essentiels, 40, 42
- Rôles principaux, 40
- Rythme soutenable, 94
- Satisfaisant, 76
- SBOK® framework*, 7
- Scalability of Scrum, 5
- Schedule performance index, 79
- Schedule variance, 79
- Schéma de conception, 222
- Schémas simplifiés, 75
- Scrum, 2
- Scrum Aspects, 10
- Scrum Core Team, 6
- Scrum Developer Certified (SDC®), 6
- Scrum Guidance Body, 69
- Scrum Guidance Body (SGB), 11, 43
- Scrum Master, 6, 11, 41, 47, 69
- Scrum Master Certified (SMC®), 6
- Scrum Master du programme, 143
- Scrum Master identifié, 153
- Scrum principes, 8, 21
- Scrum Product Owner Certified (SPOC®), 6
- Scrum Team, 6, 11
- Scrum Teams, 5
- Scrumboard, 219
- Scrumboard mis à jour, 222
- Sélection de l'équipe Scrum, 157
- Sélection du personnel, 52
- Self-organization, 9, 21, 26
- Sessions de planification de la release, 178
- Sessions JAD, 146
- Seuil de risque, 121
- Ship Deliverables*, 18
- Simulations, 82
- Speed Boat, 244
- Sponsor, 43, 68
- Sprint, 2, 33
- Sprint burndown chart, 213
- Sprint Planning Meeting, 33
- Sprint Review Meeting, 34

Stabilité, 103
Stakeholder Meeting, 2
Stakeholder(s), 11
Stratégie de collaboration, 159
Stratégie de communication, 254
Stratégie pilote, 253
Structure de répartition des risques, 123
Styles de leadership, 61
Supportive, 61
Sustainable pace, 4
Task-oriented leadership, 61
Taux de rendement interne (TRI), 74
Taux de succès de terminé, 245
Techniques d'identification des risques:, 122
Techniques de communication, 233
Terminé, 90
Théorie X, 64
Théorie Y, 64
Théories RH, 58
Time-boxing, 10, 22, 32, 107
Tolérance aux risques, 121
Traditional Project Management, 20
Transparency, 4, 22
Trois questions quotidiennes, 226
User stories, 191, 194, 200, 271, 278
User stories confirmées, estimées et validées, 198, 201
User stories validées et mises à jour, 205
Utilisateurs, 43, 69
Utility function, 121
Valeur, 173
Valeur actuelle nette (VAN), 73
Valeur commerciale, 87
Valeur commerciale offerte, 212
Valeur monétaire attendue (VMA), 126
Value-based Prioritization, 10, 21, 31
Value-driven Delivery, 12
Variance at completion, 79
Vélocité, 212
Vélocité de l'équipe, 245
Vendors, 11
Vidéoconférence, 227
Vision de l'entreprise, 144
Voix du client, 45
War room, 227

Der grundlegende Leitfaden für eine erfolgreiche Projekt-Durchführung mittels Scrum.

Der *SBOK*[®] wurde entwickelt als Standardleitfaden sowohl für Organisationen und professionelle Anwender, die Scrum implementieren möchten, als auch für diejenigen, die Scrum schon jetzt anwenden, aber notwendige Verbesserungen an Ihren Prozessen vornehmen möchten.

Der *SBOK*[®] zieht sein kombiniertes Wissen und seine Einblicke aus Tausenden von Projekten, quer durch verschiedene Organisationen und Industriezweige. Die Beiträge vieler Scrum Experten sowie Praktiker der Projektimplementierung sind bei seiner Entwicklung berücksichtigt worden. Der Fokus von Scrum auf wertgetriebene Lieferung hilft Scrum Teams, in einem Projekt so früh wie möglichen Ergebnisse zu liefern. Auf diese Weise wird die Kapitalrendite jener Gesellschaften, die Scrum als ihren bevorzugten Rahmen zur Projektdurchführung verwenden, deutlich verbessert. Darüber hinaus ist es durch die Verwendung von kurzen, iterativen Produkt-Entwicklungszyklen und häufiger Interaktion zwischen den Kunden und den Scrum Teams einfach, Änderungen des Projektumfangs vorzunehmen.

Der *SBOK*[®] kann sowohl von erfahrenen Praktikern von Scrum oder anderen Methoden der Produkt- oder Dienst-Entwicklung als Referenz und Leitfaden verwendet werden, als auch von Personen, die keine vorherige Erfahrung oder Kenntnis von Scrum oder anderen Methoden der Projekt-Durchführung haben.

Das erste Kapitel beschreibt den Zweck und Rahmen des *SBOK*[®] und liefert eine Einführung in die Schlüsselkonzepte von Scrum. Es enthält eine Zusammenfassung der Scrum Prinzipien, Scrum Aspekte und Scrum Prozesse. Das zweite Kapitel vertieft die sechs Scrum Prinzipien, welche die Grundlage der Scrum Rahmenstruktur bilden. Vom dritten bis zum siebten Kapitel werden die fünf Scrum Aspekte ausgearbeitet, die bei allen Projekten angesprochen werden müssen - Organisation, geschäftliche Rechtfertigung, Qualität, Änderung und Risiko. Kapitel acht bis zwölf decken die 19 grundlegenden Scrum Prozesse ab, die bei einem Scrum Projekt zum Einsatz kommen. Diese Prozesse sind Teil der fünf Scrum Phasen: Initiierung, Planung und Abschätzung, Implementierung, Review und Rückblick, und Freigabe. Es werden sowohl Details der zugehörigen Inputs und Outputs jedes Prozesses beschrieben, als auch die verschiedenen Hilfsmittel, die jeweils verwendet werden können.

Diese dritte Auflage des *SBOK*[®] ergänzt die vorherige Ausgabe in den Kapiteln dreizehn und vierzehn mit erweitertem Inhalt zum Thema der Skalierung von Scrum, sowohl für große Projekte als auch für das Unternehmen.

Obwohl der *SBOK*[®] ein sehr umfassendes Nachschlagewerk für Scrum ist, ist sein Inhalt so organisiert, dass, unabhängig von der vorherigen Scrum Kenntnis des jeweiligen Lesers, das Nachschlagen von Themen einfach und das Lesen unterhaltsam sein sollte.

