



A GUIDE TO THE SCRUM BODY OF KNOWLEDGE

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

Troisième 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

© 2019 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 – Troisième 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™, veillez 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 troisième 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 OF CONTENTS

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
1.5	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	24
2.4	Auto-organisation.....	26
2.4.1	Avantages de l'auto-organisation	27
2.5	Collaboration.....	28
2.5.1	Avantages de la collaboration dans les projets Scrum	29
2.5.2	Importance du regroupement dans la collaboration.....	31
2.6	Priorisation basée sur la valeur.....	31
2.7	Time-boxing	33
2.7.1	Time-boxes Scrum	33
2.8	Développement itératif	35
2.9	L'approche Scrum comparée à la gestion de projet traditionnelle.....	38
3.	ORGANISATION.....	39

3.1	Introduction	39
3.2	Guide des rôles.....	40
3.3	Rôles des projets Scrum.....	40
3.3.1	Rôles principaux.....	40
3.3.2	Rôles non essentiels	42
3.4	Product Owner	44
3.4.1	Voix du client (VOC)	45
3.4.2	Chief Product Owner	45
3.4.3	Product Owner du programme	46
3.4.4	Product Owner du portefeuille	47
3.5	Scrum Master.....	47
3.5.1	Chief Scrum Master.....	48
3.5.2	Scrum Master du programme	50
3.5.3	Scrum Master du portefeuille.....	50
3.6	Équipe Scrum	50
3.6.1	Sélection du personnel	52
3.6.2	Taille de l'équipe Scrum	52
3.7	Projets, programmes et portefeuilles Scrum	53
3.7.1	Définition de projet, programme et portefeuille	53
3.7.2	Maintenir la participation des parties prenantes	55
3.8	Résumé des responsabilités	56
3.9	L'approche Scrum comparée à la gestion de projet traditionnelle.....	57
3.10	Théories RH populaires et leur importance dans Scrum.....	58
3.10.1	Le modèle de Tuckman sur la dynamique des groupes	58
3.10.2	Gestion des conflits	59
3.10.3	Techniques de gestion des conflits.....	59
3.10.4	Styles de leadership	61
3.10.5	La pyramide des besoins de Maslow.....	63
3.10.6	Théorie X et théorie Y.....	64
4.	JUSTIFICATION COMMERCIALE	65
4.1	Introduction	65
4.2	Guide des rôles.....	66

4.3	Livraison de la valeur ajoutée	66
4.3.1	Responsabilités du Product Owner concernant la justification commerciale	68
4.3.2	Responsabilités des autres rôles Scrum concernant la justification commerciale	68
4.4	Importance de la justification commerciale	69
4.4.1	Paramètres utilisés pour déterminer la justification commerciale	70
4.4.2	Justification commerciale et cycle du projet.....	70
4.5	Techniques de justification commerciale	72
4.5.1	Estimation de la valeur du projet	73
4.5.2	Planification de la valeur.....	74
4.5.3	Classement relatif des priorités	77
4.5.4	Représentation des stories	77
4.6	Justification de valeur continue	77
4.6.1	Analyse de valeur acquise.....	78
4.6.2	Diagramme de flux cumulatif (DFC)	81
4.7	Confirmer la réalisation des bénéfices	82
4.7.1	Prototypes, simulations et démonstrations	82
4.8	Résumé des responsabilités	82
4.9	L'approche Scrum comparée à la gestion de projet traditionnelle.....	83
5.	QUALITÉ	85
5.1	Introduction	85
5.2	Guide des rôles.....	86
5.3	Définir la qualité	86
5.3.1	Qualité et portée	86
5.3.2	Qualité et valeur commerciale	87
5.4	Critères d'acceptation et backlog de produit priorisé	88
5.4.1	Écriture des critères d'acceptation.....	89
5.4.2	Définition de <i>terminé</i>	90
5.4.3	Définition minimum de <i>terminé</i>	91
5.4.4	Acceptation ou rejet des éléments du backlog de produit priorisé.....	92
5.5	Gestion de la qualité dans Scrum	92
5.5.1	Planification de la qualité	93
5.5.2	Contrôle et assurance qualité	94

5.5.3	Cycle Plan-Do-Check-Act (PDCA).....	95
5.6	Résumé des responsabilités	96
5.7	L'approche Scrum comparée à la gestion de projet traditionnelle.....	97
6.	CHANGEMENT.....	99
6.1	Introduction	99
6.2	Guide des rôles.....	100
6.3	Aperçu	100
6.3.1	Demandes de changement rejetées et approuvées	101
6.4	Le changement dans Scrum	103
6.4.1	Équilibrer flexibilité et stabilité.....	103
6.4.2	Atteindre la flexibilité.....	103
6.5	Intégration des changements.....	109
6.5.1	Changements apportés à un sprint.....	109
6.6	Changements apportés aux portefeuilles et aux programmes.....	115
6.6.1	Dans un portefeuille.....	115
6.6.2	Dans un programme	115
6.7	Résumé des responsabilités	117
6.8	L'approche Scrum comparée à la gestion de projet traditionnelle.....	118
7.	RISQUES	119
7.1	Introduction	119
7.2	Guide des rôles.....	120
7.3	Qu'est-ce que le risque ?	120
7.3.1	Différences entre les risques et les problèmes	120
7.3.2	Attitude face aux risques	121
7.4	Procédure de gestion des risques.....	122
7.4.1	Identification des risques	122
7.4.2	Évaluation des risques.....	123
7.4.3	Priorisation des risques	127
7.4.4	Atténuation des risques	128
7.4.5	Communication des risques	129
7.5	Minimiser les risques avec l'approche Scrum	131
7.6	Les risques dans les portefeuilles et les programmes	132

7.6.1	Dans un portefeuille.....	132
7.6.2	Dans un programme	132
7.7	Résumé des responsabilités	134
7.8	L'approche Scrum comparée à la gestion de projet traditionnelle.....	135
8.	INITIATION	137
8.1	Création de la vision de projet.....	141
8.1.1	Contributions	143
8.1.2	Outils	145
8.1.3	Résultats.....	147
8.2	Identification du Scrum Master et des parties prenantes	148
8.2.1	Contributions	150
8.2.2	Outils	152
8.2.3	Résultats.....	153
8.3	Formation de l'équipe Scrum	154
8.3.1	Contributions	155
8.3.2	Outils	157
8.3.3	Résultats.....	158
8.4	Développement des epics.....	160
8.4.1	Contributions	161
8.4.2	Outils	164
8.4.3	Résultats.....	167
8.5	Création du backlog de produit priorisé	168
8.5.1	Contributions	169
8.5.2	Outils	171
8.5.3	Résultats.....	173
8.6	Planification de la release	175
8.6.1	Contributions	176
8.6.2	Outils	178
8.6.3	Résultats.....	179
8.7	Diagramme de flux de données de la phase d'initiation.....	181
9.	PLANIFICATION ET ESTIMATION.....	183
9.1	Création des user stories	187

9.1.1	Contributions	188
9.1.2	Outils	190
9.1.3	Résultats.....	191
9.2	Estimation des user stories	193
9.2.1	Contributions	194
9.2.2	Outils	195
9.2.3	Résultats.....	198
9.3	Validation des user stories.....	199
9.3.1	Contributions	200
9.3.2	Outils	201
9.3.3	Résultats.....	201
9.4	Identification des tâches	202
9.4.1	Contributions	203
9.4.2	Outils	203
9.4.3	Résultats.....	205
9.5	Estimation des tâches.....	206
9.5.1	Contributions	207
9.5.2	Outils	208
9.5.3	Résultats.....	209
9.6	Création du backlog de sprint	210
9.6.1	Contributions	211
9.6.2	Outils	211
9.6.3	Résultats.....	212
9.7	Diagramme de flux de données de la phase de planification et d'estimation.....	214
10.	IMPLÉMENTATION	215
10.1	Création des éléments livrables	218
10.1.1	Contributions	219
10.1.2	Outils	221
10.1.3	Résultats.....	222
10.2	Organisation des mêlées quotidiennes	224
10.2.1	Contributions	225
10.2.2	Outils	226

10.2.3	Résultats.....	227
10.3	Affinage du backlog de produit priorisé.....	229
10.3.1	Contributions	230
10.3.2	Outils	232
10.3.3	Résultats.....	233
10.4	Diagramme de flux de données de la phase d'implémentation.....	234
11.	REVUE ET RÉTROSPECTIVE.....	235
11.1	Démonstration et validation du sprint.....	237
11.1.1	Contributions	238
11.1.2	Outils	240
11.1.3	Résultats.....	241
11.2	Rétrospective de sprint	242
11.2.1	Contributions	243
11.2.2	Outils	243
11.2.3	Résultats.....	245
11.3	Diagramme de flux de données de la phase de revue et de rétrospective	247
12.	RELEASE	249
12.1	Livraison des produits	251
12.1.1	Contributions	252
12.1.2	Outils	254
12.1.3	Résultats.....	254
12.2	Rétrospective du projet	255
12.2.1	Contributions	256
12.2.2	Outils	257
12.2.3	Résultats.....	258
12.3	Diagramme de flux de données de la phase de release	259
13.	SCRUM POUR LES GROS PROJETS.....	261
13.1	Création de composants pour les gros projets.....	265
13.1.1	Contributions	266
13.1.2	Outils	269
13.1.3	Résultats.....	271
13.2	Déroulement et coordination des sprints.....	274

13.2.1	Contributions	275
13.2.2	Outils	277
13.2.3	Résultats.....	278
13.3	Préparation de la release d'un gros projet	279
13.3.1	Contributions	280
13.3.2	Outils	281
13.3.3	Résultats.....	282
13.4	Impact des gros projets sur les processus Scrum fondamentaux	282
14.	SCRUM POUR L'ENTREPRISE.....	287
14.1	Création de composants pour les programmes ou les portefeuilles.....	291
14.1.1	Contributions	292
14.1.2	Outils	293
14.1.3	Résultats.....	294
14.2	Revue et mise à jour du Scrum Guidance Body	295
14.2.1	Contributions	296
14.2.2	Outils	297
14.2.3	Résultats.....	298
14.3	Création et affinage du backlog de programme ou de portefeuille.....	299
14.3.1	Contributions	300
14.3.2	Outils	303
14.3.3	Résultats.....	304
14.4	Coordination des composants pour les programmes ou les portefeuilles	305
14.4.1	Contributions	307
14.4.2	Outils	309
14.4.3	Résultats.....	311
14.5	Réunion de rétrospective de release de programme ou de portefeuille.....	312
14.5.1	Contributions	313
14.5.2	Outils	314
14.5.3	Résultats.....	314
ANNEXE A.	UN APERÇU DU CONCEPT AGILE	315
ANNEXE B.	AUTEURS ET CONTRIBUTEURS DU GUIDE SBOK™	325
ANNEXE C.	MISES À JOUR DE LA TROISIÈME ÉDITION.....	329

TABLE DES MATIÈRES

GLOSSAIRE.....	333
INDEX.....	377

LIST OF FIGURES

Figure 1-1 : Flux de l'approche Scrum pour un sprint	2
Figure 1-2 : Cadre du <i>Guide SBOK™</i>	7
Figure 1-3 : Principes Scrum.....	9
Figure 1-4 : Organisation de Scrum	12
Figure 2-1 : La transparence dans Scrum.....	23
Figure 2-2 : L'inspection dans Scrum.....	24
Figure 2-3 : L'adaptation dans Scrum	25
Figure 2-4 : Les défis relatifs à la gestion de projet traditionnelle	26
Figure 2-5 : Les objectifs d'une équipe autoorganisée.....	28
Figure 2-6 : Les avantages de la collaboration dans les projets Scrum	30
Figure 2-7 : Priorisation basée sur la valeur.....	33
Figure 2-8 : Durées des réunions Scrum	35
Figure 2-9 : L'approche Scrum comparée à l'approche en cascade traditionnelle	37
Figure 3-1 : Aperçu des rôles Scrum	42
Figure 3-2 : Questions posées lors d'une réunion Scrum of Scrums	49
Figure 3-3 : Traits désirables pour les rôles Scrum principaux	52
Figure 3-4 : Scrum dans les organisations pour les projets, programmes ou portefeuilles	54
Figure 3-5 : Les phases du modèle de Tuckman sur le développement des groupes	58
Figure 3-6 : La pyramide des besoins de Maslow.....	63
Figure 4-1 : Livraison de la valeur ajoutée dans Scrum comparée aux projets traditionnels.....	67
Figure 4-2 : Hiérarchie des responsabilités concernant la justification commerciale.....	68
Figure 4-3 : Justification commerciale et cycle du projet.....	72
Figure 4-4 : Analyse Kano.....	76
Figure 4-5 : Exemple de diagramme de flux cumulatif (DFC)	81
Figure 5-1 : Diagramme de flux des incrémentés du projet.....	89
Figure 5-2 : Définition de <i>terminé</i> en cascade.....	91
Figure 5-3 : Le cycle PDCA dans Scrum.....	96
Figure 6-1 : Exemple de processus d'approbation des changements.....	102
Figure 6-2 : Mise à jour du backlog de produit priorisé avec les changements approuvés	102
Figure 6-3 : Caractéristiques Scrum pour atteindre la flexibilité	104
Figure 6-4 : Motivations des parties prenantes pour demander des changements.....	105
Figure 6-5 : Motivations de l'équipe Scrum principale pour effectuer des demandes de changement....	106
Figure 6-6 : Intégration des changements dans Scrum.....	110
Figure 6-7 : Impact des changements prévus sur la durée du sprint	112
Figure 6-8 : Intégrer les changements dans les portefeuilles et les programmes	116
Figure 7-1 : Exemple d'arbre de probabilité	124
Figure 7-2 : Exemple de diagramme de Pareto	125
Figure 7-3 : Exemple de grille de probabilité et d'impact.....	126

Figure 7-4 : Processus de priorisation des risques	128
Figure 7-5 : Exemple de risk burndown chart.....	130
Figure 7-6 : Gestion des risques dans les portefeuilles et les programmes	133
Figure 8-1 : Aperçu de la phase d'initiation	139
Figure 8-2 : Aperçu de la phase d'initiation (éléments essentiels)	140
Figure 8-3 : Création de la vision de projet — Contributions, outils et résultats	141
Figure 8-4 : Création de la vision de projet — diagramme de flux de données	142
Figure 8-5 : Processus de l'analyse des lacunes	147
Figure 8-6 : Identification du Scrum Master et des parties prenantes — Contributions, outils et résultats	148
Figure 8-7 : Identification du Scrum Master et des parties prenantes — Diagramme de flux de données	149
Figure 8-8 : Formation de l'équipe Scrum — Contributions, outils et résultats.....	154
Figure 8-9 : Formation de l'équipe Scrum — Diagramme de flux de données	155
Figure 8-10 : Développement des epics — Contributions, outils et résultats	160
Figure 8-11 : Développement des epics — Diagramme de flux de données	161
Figure 8-12 : Création du backlog de produit priorisé — Contributions, outils et résultats	168
Figure 8-13 : Création du backlog de produit priorisé — Diagramme de flux de données	169
Figure 8-14 : Planification de la release — Contributions, outils et résultats.....	175
Figure 8-15 : Planification de la release — Diagramme de flux de données.....	176
Figure 8-16 : Phase d'initiation — Diagramme de flux de données.....	181
Figure 9-1 : Aperçu de la phase de planification et d'estimation	185
Figure 9-2 : Aperçu de la phase de planification et d'estimation (éléments essentiels)	186
Figure 9-3 : Création des user stories — Contributions, outils et résultats.....	187
Figure 9-4 : Création des user stories — diagramme de flux de données	188
Figure 9-5 : Estimation des user stories — Contributions, outils et résultats	193
Figure 9-6 : Estimation des user stories — diagramme de flux de données	194
Figure 9-7 : Validation des user stories — Contributions, outils et résultats	199
Figure 9-8 : Validation des user stories — diagramme de flux de données	199
Figure 9-9 : Identification des tâches — Contributions, outils et résultats	202
Figure 9-10 : Identification des tâches — Diagramme de flux de données	202
Figure 9-11 : Estimation des tâches — Contributions, outils et résultats	206
Figure 9-12 : Estimation des tâches — Diagramme de flux de données.....	206
Figure 9-13 : Création du sprint backlog — Contributions, outils et résultats	210
Figure 9-14 : Création du sprint backlog — Diagramme de flux de données	210
Figure 9-15 : Phase de planification et d'estimation — Diagramme de flux de données.....	214
Figure 10-1 : Aperçu de la phase d'implémentation	216
Figure 10-2 : Aperçu de la phase d'implémentation (éléments essentiels)	217
Figure 10-3 : Création des éléments livrables — Contributions, outils et résultats.....	218
Figure 10-4 : Création des éléments livrables — Diagramme de flux de données.....	218
Figure 10-5 : Scrumbord	219

Figure 10-6 : Mêlées quotidiennes — Contributions, outils et résultats.....	224
Figure 10-7 : Mêlées quotidiennes — Diagramme de flux de données.....	224
Figure 10-8 : Affinage du backlog de produit priorisé — Contributions, outils et résultats	229
Figure 10-9 : Affinage du backlog de produit priorisé — Diagramme de flux de données.....	230
Figure 10-10 : Phase d'implémentation — Diagramme de flux de données	234
Figure 11-1 : Aperçu de la phase de revue et de rétrospective.....	236
Figure 11-2 : Aperçu de la phase de revue et rétrospective (éléments essentiels).....	237
Figure 11-3 : Démonstration et validation du sprint — Contributions, outils et résultats	237
Figure 11-4 : Démonstration et validation du sprint — Diagramme de flux de données.....	238
Figure 11-5 : Rétrospective de sprint — Contributions, outils et résultats.....	242
Figure 11-6 : Rétrospective de sprint — Diagramme de flux de données	242
Figure 11-7 : Phase de revue et rétrospective — Diagramme de flux de données	247
Figure 12-1 : Aperçu de la phase de release	250
Figure 12-2 : Aperçu de la phase de release (éléments essentiels).....	251
Figure 12-3 : Livraison des produits — Contributions, outils et résultats.....	251
Figure 12-4 : Livraison des produits — Diagramme de flux de données.....	252
Figure 12-5 : Rétrospective de projet — Contributions, outils et résultats.....	255
Figure 12-6 : Rétrospective de projet — Diagramme de flux de données.....	256
Figure 12-7 : Phase de release — Diagramme de flux de données.....	259
Figure 13-1 : Aperçu du passage à l'échelle de Scrum pour les gros projets	263
Figure 13-2 : Passage à l'échelle de Scrum pour les gros projets (éléments essentiels).....	264
Figure 13-3 : Création de composants pour les gros projets — Contributions, outils et résultats	265
Figure 13-4 : Création de composants pour les gros projets — diagramme de flux de données	266
Figure 13-5 : Déroulement et coordination des sprints — Contributions, outils et résultats	274
Figure 13-6 : Déroulement et coordination des sprints — Diagramme de flux de données	275
Figure 13-7 : Préparation de la release d'un gros projet — Contributions, outils et résultats	279
Figure 13-8 : Préparation de la release d'un gros projet — diagramme de flux de données.....	280
Figure 14-1 : Aperçu du passage à l'échelle de Scrum pour l'entreprise	289
Figure 14-2 : Passage à l'échelle de Scrum pour l'entreprise (éléments essentiels)	290
Figure 14-3 : Création de composants pour les programmes ou les portefeuilles — Contributions, outils et résultats	291
Figure 14-4 : Création de composants pour les programmes ou les portefeuilles — diagramme de flux de données.....	291
Figure 14-5 : Revue et mise à jour du Scrum Guidance Body — Contributions, outils et résultats	295
Figure 14-6 : Revue et mise à jour du Scrum Guidance Body — Diagramme de flux de données	296
Figure 14-7 : Création et affinage du backlog de programme ou de portefeuille — Contributions, outils et résultats	299
Figure 14-8 : Création et affinage du backlog de programme ou de portefeuille — Diagramme de flux de données.....	300
Figure 14-9 : Coordination des composants pour les programmes ou les portefeuilles — Contributions, outils et résultats.....	306

Figure 14-10 : Coordination des composants pour les programmes ou les portefeuilles — diagramme de flux de données	306
Figure 14-11 : Réunion Scrum of Scrums (SoS).....	310
Figure 14-12 : Rétrospective des releases de programme ou de portefeuille — Contributions, outils et résultats	312
Figure 14-13 : Rétrospective des releases de programme ou de portefeuille — diagramme de flux de données.....	312

LIST OF TABLES

Tableau 1-1 : Résumé des processus Scrum fondamentaux.....	15
Tableau 1-2 : Résumé des processus Scrum supplémentaires	16
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	45
Tableau 3-2 : Responsabilités du Scrum Master dans les processus Scrum.....	48
Tableau 3-3 : Responsabilités de l'équipe Scrum dans les processus Scrum	51
Tableau 3-4 : Résumé des responsabilités pertinentes à l'organisation	57
Tableau 4-1 : Formules de valeur acquise	79
Tableau 4-2 : Résumé des responsabilités correspondant à la justification commerciale	83
Tableau 5-1 : Résumé des responsabilités correspondant à la qualité	97
Tableau 6-1 : Résumé des responsabilités relatives aux changements.....	117
Tableau 7-1 : Résumé des responsabilités correspondant aux risques	134
Tableau 13-1 : Résumé de l'impact des gros projets sur les processus Scrum fondamentaux	286

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
- 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. 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 forces 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.

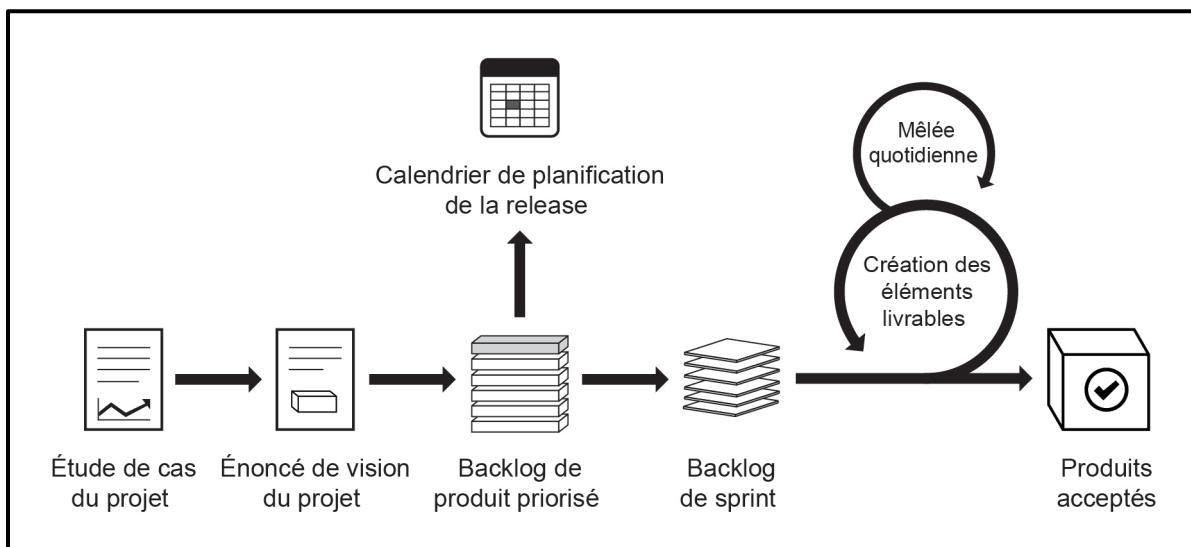


Figure 1-1 : Flux de l'approche Scrum pour un sprint

Le cycle Scrum débute par une réunion des parties prenantes 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 six semaines et engage le travail de

l'équipe Scrum afin de créer des produits potentiellement livrables ou des incrémentés 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. Vers la fin du sprint, une réunion de revue de sprint est organisée, durant laquelle le Product Owner et les parties prenantes 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 relai 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 ».

Ken Schwaber et Jeff Sutherland développent le concept de Scrum et son application au développement de logiciels lors d'une présentation à la conférence OOPSLA (*Object-Oriented Programming, Systems, Languages & Applications Conference*) organisée en 1995 à Austin au Texas. Depuis lors, de nombreux spécialistes, experts et auteurs continuent à affiner la conceptualisation et le cadre de travail Scrum. Ces dernières années, l'approche Scrum a gagné en popularité. Elle est maintenant l'approche de développement de projet privilégiée par de nombreuses organisations dans le monde entier.

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 *mélées 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 d'*affinage du backlog de produit 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 *mélées 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 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™*

Ces dernières années, il est devenu évident que les organisations qui privilégient l'approche Scrum pour leurs projets réalisent continuellement un meilleur retour sur capital investi. 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 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 les parties prenantes (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. Les chapitres 13 à 14 abordent les processus supplémentaires relatifs au passage à l'échelle de Scrum pour les gros projets et pour les entreprises.

La figure 1-2 illustre le cadre du *Guide SBOK™* et montre comment les principes, aspects et processus interagissent entre eux et revêtent tous la même importance afin de mieux comprendre l'approche Scrum.

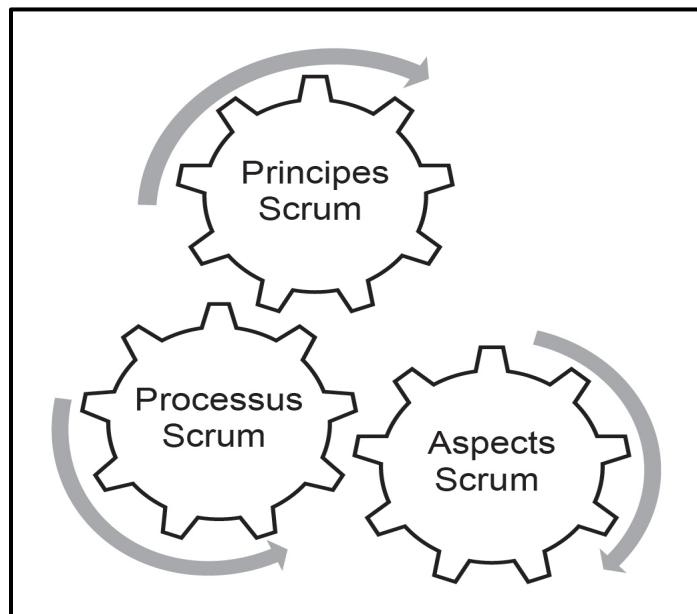


Figure 1-2 : Cadre du *Guide SBOK™*

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 processus supplémentaires s'appliquent lors du passage à l'échelle de Scrum pour les gros projets (chapitre 13) ou pour les entreprises (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.

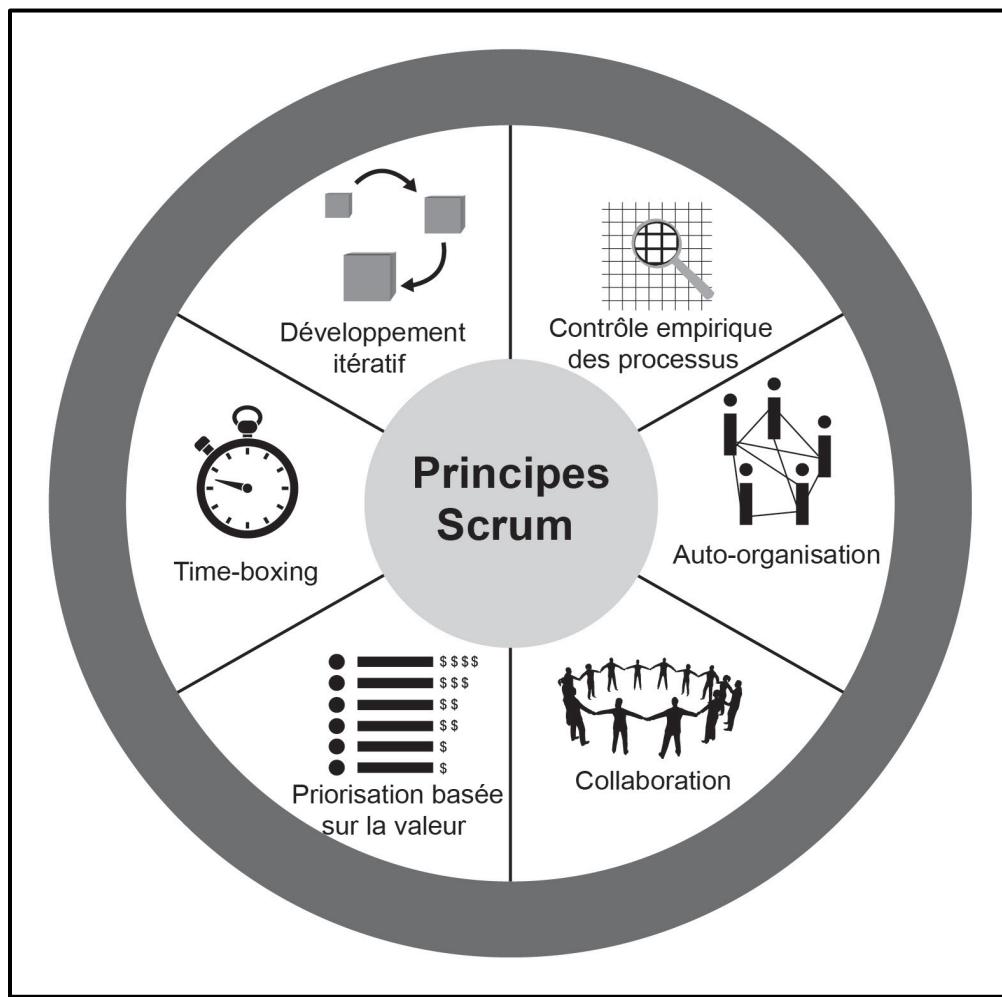


Figure 1-3 : Principes Scrum

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.
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 considérer la gestion de 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. Dans Scrum, les évènements limités dans le temps incluent les sprints, les mélées quotidiennes, les réunions de planification de sprint et les réunions de revue 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 incrémentés 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**, c'est un terme collectif qui inclut les clients, les utilisateurs et les sponsors. Les parties prenantes interagissent fréquemment avec l'équipe Scrum principale et influencent le projet au cours de son développement. Le projet produit principalement des bénéfices collaboratifs pour les parties prenantes.
- 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 1-4 illustre l'organisation de Scrum.

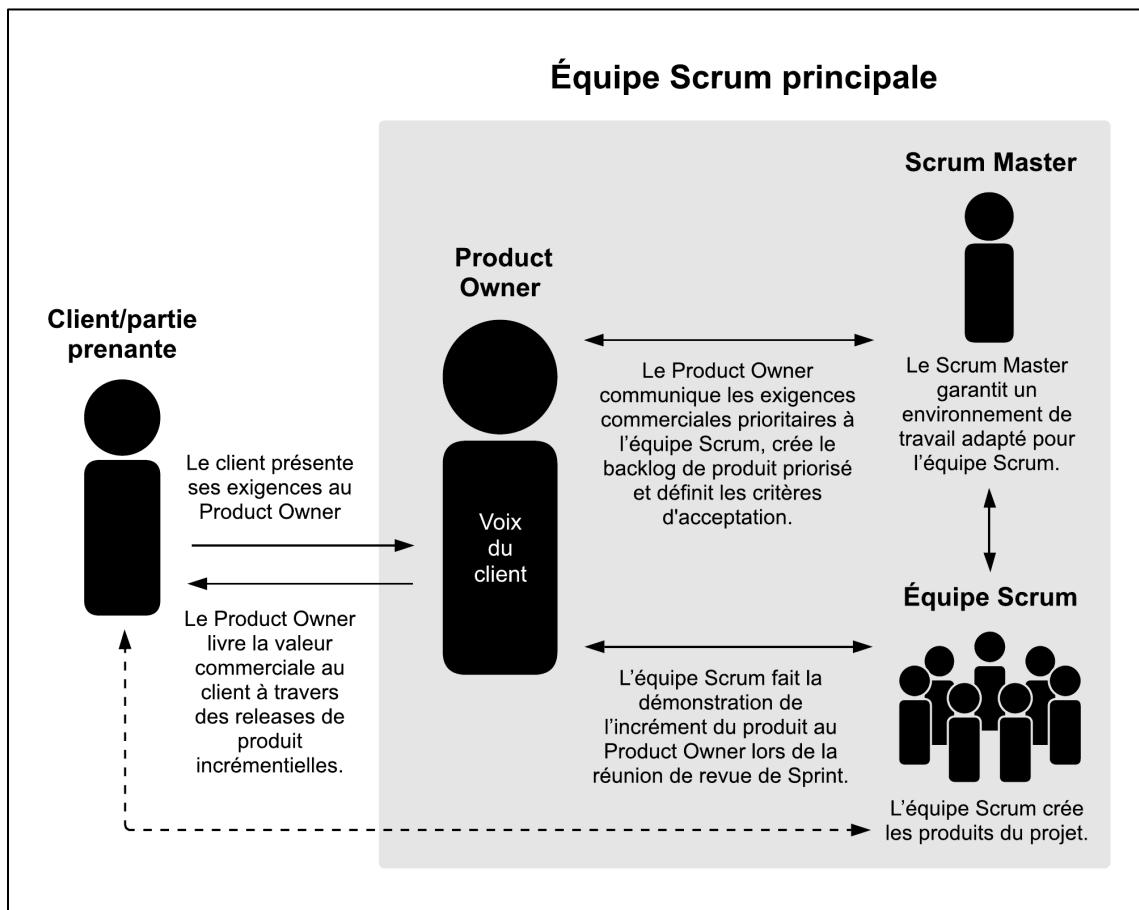


Figure 1-4 : Organisation de Scrum

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 les parties prenantes intéressées.

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 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 incrémentés 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 incrémentés 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 les parties prenantes (y compris les clients et les utilisateurs) avec de vrais incrémentés 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 incrémentés 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 essayent 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 (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 les parties prenantes 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. 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	1. Création de la vision de projet 2. Identification du Scrum Master et des parties prenantes 3. Formation de l'équipe Scrum 4. Développement des épicis 5. Création du backlog de produit priorisé 6. Planification de la release
9	Phase de planification et d'estimation	7. Création des user stories 8. Estimation des user stories 9. Validation des user stories 10. Identification des tâches 11. Estimation des tâches 12. Création du backlog de sprint
10	Phase d'implémentation	13. Création des incrémentums de produit 14. Mêlées quotidiennes 15. Affinage du backlog de produit priorisé
11	Phase de revue et de rétrospective	16. Démonstration et validation du sprint 17. Rétrospective de sprint
12	Phase de release	18. Livraison des produits 19. Rétrospective du projet

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 la coordination de plusieurs équipes, il existe trois processus Scrum supplémentaires, définis dans le chapitre 13 : Scrum pour les gros projets. Il existe aussi des processus spécifiques définis pour la mise en œuvre de Scrum à l'échelle de l'entreprise, ceux-ci sont

abordés dans le chapitre 14 : Scrum pour l'entreprise. Ces processus Scrum supplémentaires sont résumés dans le tableau 1-2.

Chapitre	Applicabilité	Processus Scrum supplémentaires
13	Scrum pour les gros projets	<ol style="list-style-type: none"> 1. Création de composants pour les gros projets 2. Mener et coordonner les sprints 3. Préparation de la release d'un gros projet
14	Scrum pour l'entreprise	<ol style="list-style-type: none"> 1. Création de composants pour les programmes ou les portefeuilles 2. Revue et mise à jour du Scrum Guidance Body 3. Création et affinage du backlog de programme ou de portefeuille 4. Coordination des composants pour les programmes ou les portefeuilles 5. Réunion de rétrospective de release de programme ou de portefeuille

Tableau 1-2 : Résumé des processus Scrum supplémentaires

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 des parties prenantes* — le Scrum Master et les parties prenantes 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 les parties prenantes 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 les parties prenantes. 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. *Création du backlog de sprint* — 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 incrément de produit* — l'équipe Scrum travaille sur les tâches du backlog de sprint afin de créer des incrément 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. *Mélées quotidiennes* — une réunion très ciblée et limitée dans le temps est organisée chaque jour. Elle est appelée *mélée 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. *Affinage du backlog de produit priorisé* — l'affinage du backlog de produit 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 incrément de produit au Product Owner et aux parties prenantes 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 incrément 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 concernées. Un accord officiel sur les produits finis documente la réussite du sprint.
19. *Rétrospective du projet* — Ce processus conclue le projet. Les parties prenantes de l'organisation et l'équipe Scrum principale se rassemblent afin d'effectuer une rétrospective du projet 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 Scrum pour les gros projets

Création de composants pour les gros projets — ce processus sert à définir la façon dont plusieurs Product Owners ou plusieurs équipes Scrum travaillent ensemble. Les ressources spécialisées ou communes ainsi que les composants communs sont également identifiés.

Déroulement et coordination des sprints — ce processus aborde les aspects spécifiques qui doivent être considérés durant chaque sprint pour les gros projets. Le cas échéant, des réunions Scrum of Scrums sont organisées afin de coordonner les efforts entre les différentes équipes Scrum.

Préparation de la release d'un gros projet — pour les gros projets, il est parfois commercialement nécessaire d'organiser un sprint spécial pour préparer la release du produit (l'équipe du projet en prend la décision selon les besoins commerciaux). Ce processus est un sprint de préparation de release. Décider de l'organisation d'un sprint de préparation ne signifie pas que les activités devant être exécutées au cours des autres sprints peuvent être retardées jusqu'à ce point.

1.4.4.7 Scrum pour l'entreprise

Création de composants pour les programmes ou les portefeuilles — le Product Owner du programme ou du portefeuille et les parties prenantes principales identifient les ressources et composants communs requis pour le programme ou le portefeuille. La définition minimum de terminé est établie et toutes les parties prenantes sont identifiées.

Revue et mise à jour du Scrum Guidance Body — les recommandations du Scrum Guidance Body sont régulièrement passées en revue par les membres du Scrum Guidance Body et celles-ci sont mises à jour le cas échéant. Le changement des membres du Scrum Guidance Body est aussi abordé durant ce processus.

Création et affinage du backlog de programme ou de portefeuille — le backlog de programme ou de portefeuille est créé, mis à jour et entretenu. Il est possible de formuler des suggestions pour améliorer les recommandations du Scrum Guidance Body et de modifier les échéances de mise en œuvre en fonction du changement des exigences et/ou du progrès des projets du programme ou du portefeuille.

Coordination des composants pour les programmes ou les portefeuilles — les composants du programme ou du portefeuille sont coordonnés. Les dépendances entre les projets sont abordées, les obstacles communs sont discutés et les meilleures pratiques sont partagées. Des recommandations pour l'amélioration du Scrum Guidance Body sont parfois formulées.

Rétrospective de release de programme ou de portefeuille — le Product Owner du programme ou du portefeuille et les parties prenantes principales se réunissent afin de faire la rétrospective de la release d'un programme ou d'un portefeuille et d'en assimiler les enseignements. Généralement, les enseignements mènent à la mise en œuvre des améliorations actionnables convenues lors des futurs projets.

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. PRINCIPLES

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
- 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 les parties prenantes 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 les parties prenantes 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 repose sur trois principes fondamentaux : la transparence, l'inspection et l'adaptation.

2.3.1 Transparency

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 les parties prenantes 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 calendrier de planification de la release qui peut être coordonné à travers plusieurs équipes Scrum
- 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
- les mélées 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 des incrémentés de produit du sprint potentiellement livrables

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

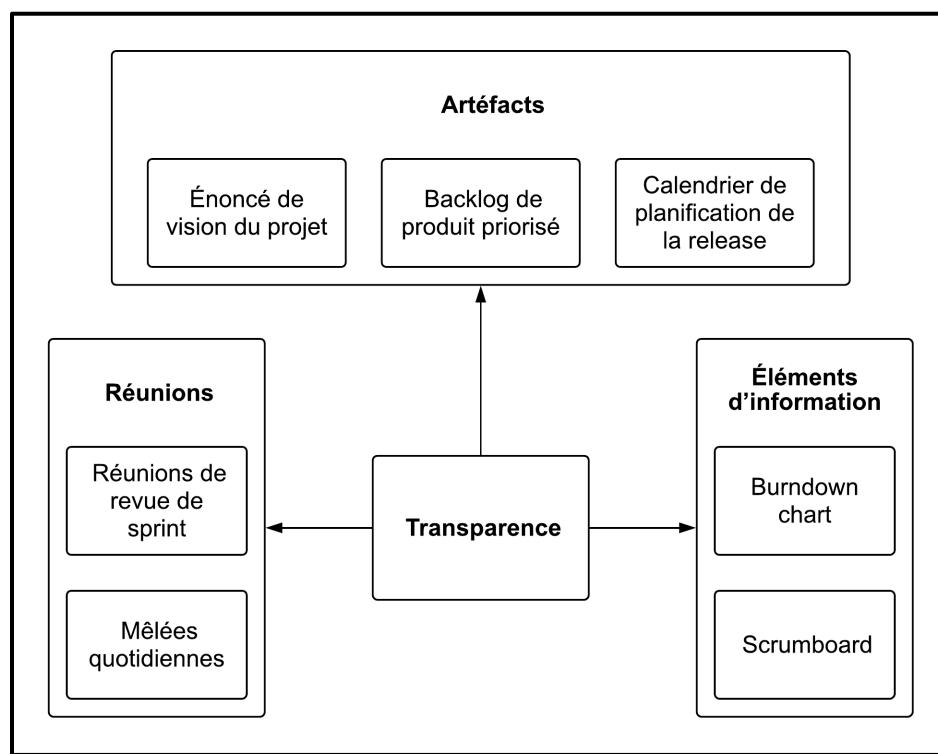


Figure 2-1 : La transparence dans Scrum

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 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 incrémentés 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.

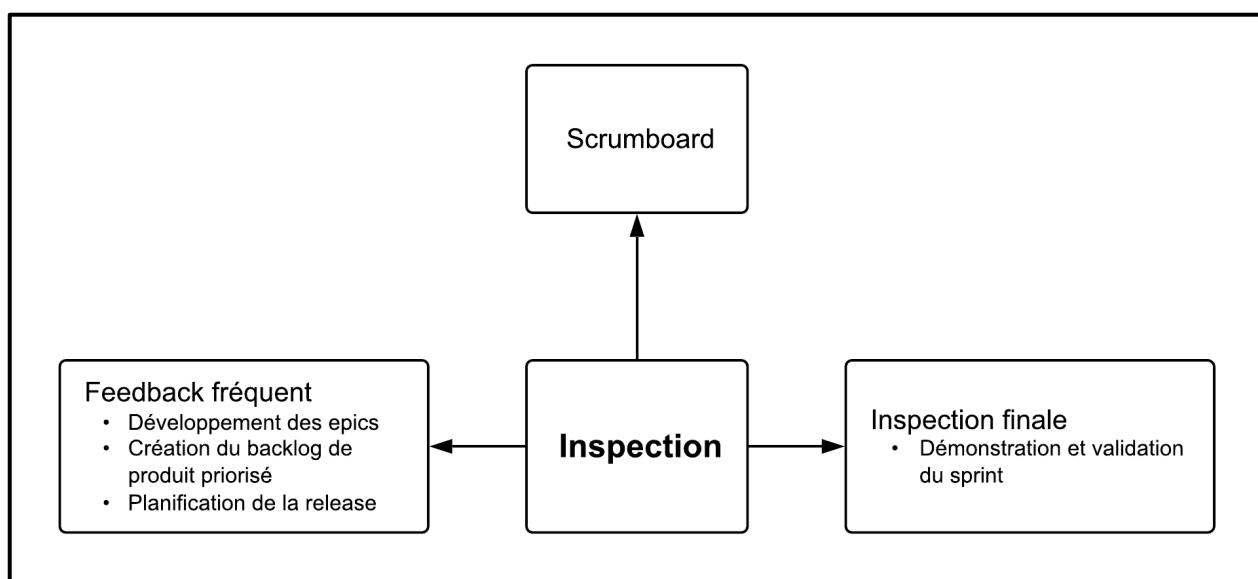


Figure 2-2 : L'inspection dans Scrum

2.3.3 Adaptation

On parle d'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 qu'elles effectuent. Voici quelques exemples d'adaptation :

- lors des mélées 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*, l'*affinage du backlog de produit 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 d'*affinage du backlog de produit priorisé*.
- le Scrum Guidance Body interagit avec les membres de l'équipe Scrum lors des processus de *création des user stories*, d'*estimation des tâches*, de *création des incrément de produit* et d'*affinage du backlog de produit 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*.
- durant la réunion de rétrospective du projet, 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.

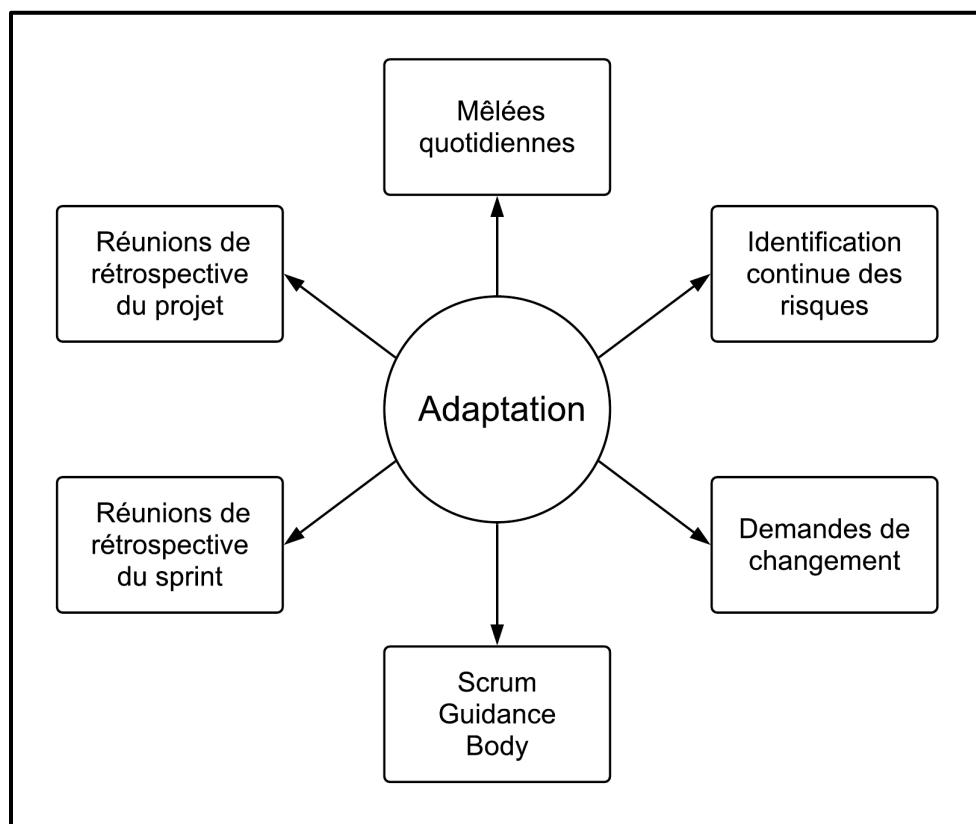


Figure 2-3 : L'adaptation dans Scrum

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.

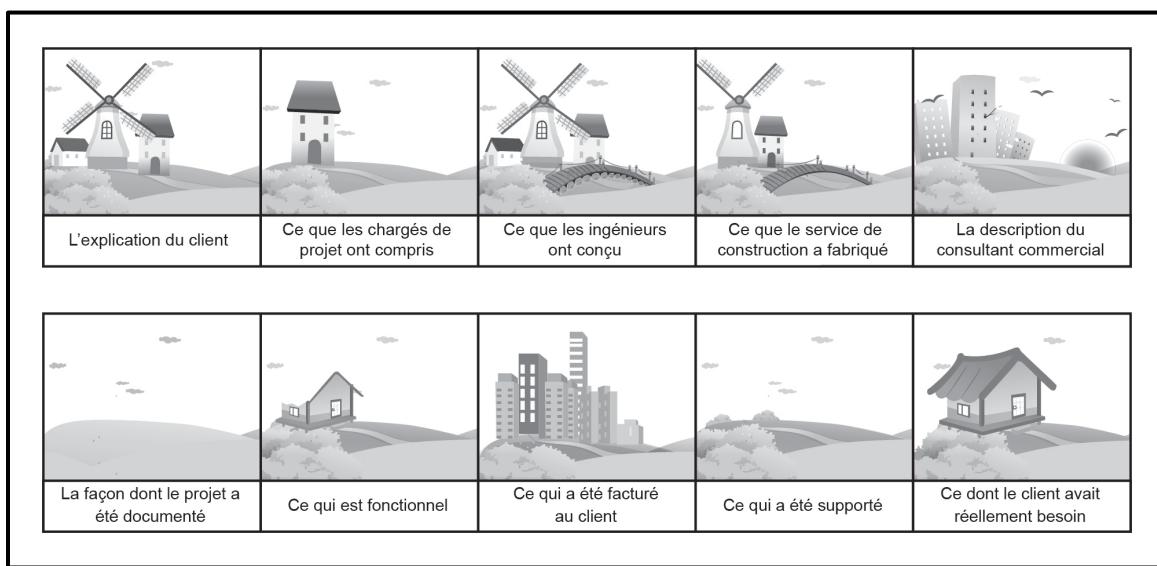


Figure 2-4 : Les défis relatifs à la gestion de projet traditionnelle

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 autoorganisés.

Le style de leadership privilégié dans l'approche Scrum est celui de « servant leadership » (leader serviteur) et met en avant la réalisation des résultats à travers l'attention portée aux besoins de l'équipe. Voir la section 3.10.3 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

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 les parties prenantes 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 incrément 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.

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 *création du backlog de sprint*
- 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 incrément de produit*
- offrir des résultats tangibles acceptés par le client et les autres parties prenantes lors du processus de *démonstration et validation du sprint*

- aborder et résoudre les problèmes individuels en équipe lors des *mélées 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.

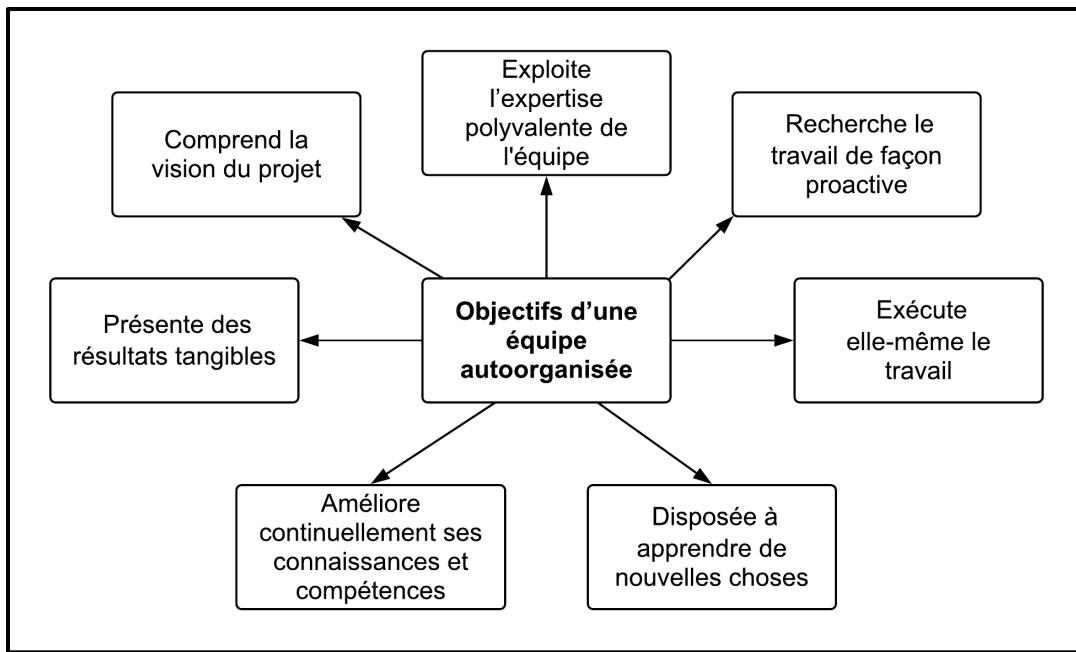


Figure 2-5 : Les objectifs d'une équipe autoorganisée

2.5 Collaboration

Dans l'approche Scrum, la collaboration fait référence à l'équipe Scrum principale qui travaille avec les parties prenantes 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.

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 les parties prenantes 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 les parties prenantes 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 les parties prenantes grâce à un backlog de produit priorisé transparent afin de créer les produits du projet. Les *mêlées quotidiennes*, les processus *d'affinage du backlog de produit 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 incrément 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.

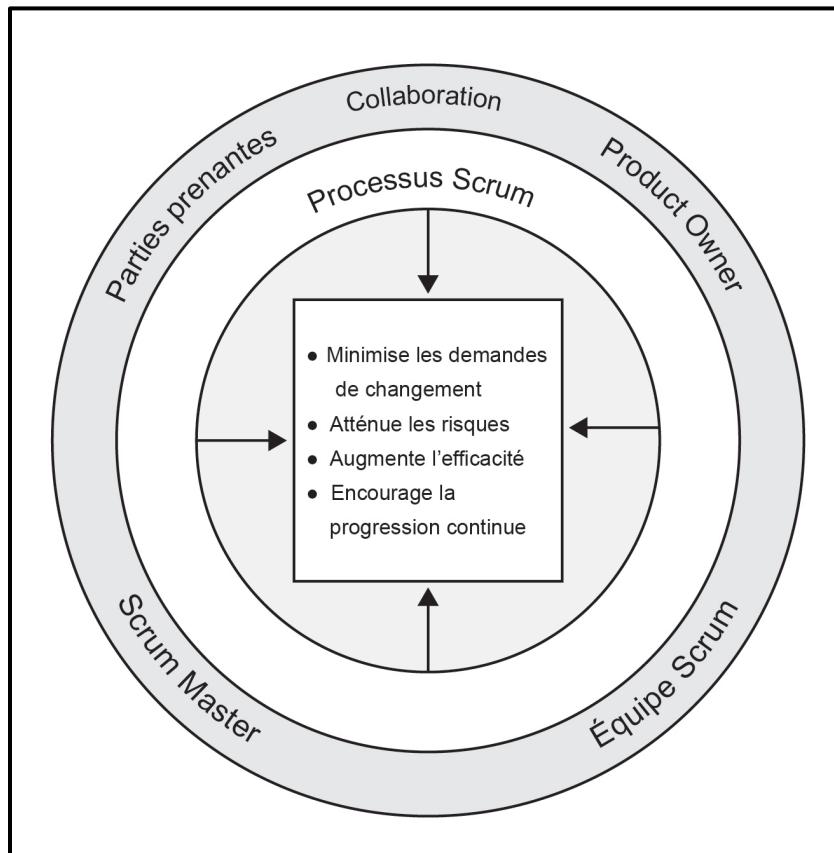


Figure 2-6 : Les avantages de la collaboration dans les projets Scrum

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 scrumboards, 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 scrumboards, les affichages, etc.

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 incrément. 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 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'affinage du backlog de produit 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 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.

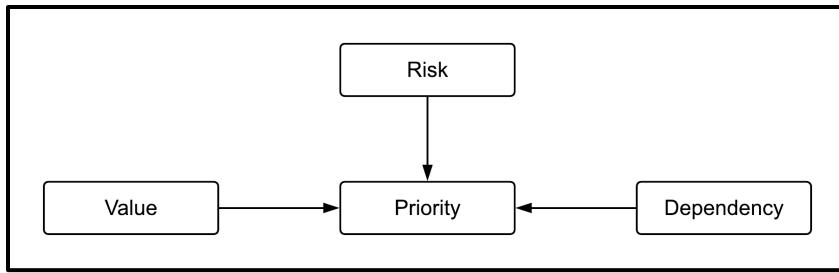


Figure 2-7 : Priorisation basée sur la valeur

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 vitesse des équipes

Le time-boxing peut être utilisé dans de nombreux processus Scrum, par exemple, lors des mélées 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 appliquée 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 six 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.

- **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 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 *création du backlog de sprint*. Elle est limitée à une durée de huit heures pour un sprint d'un mois. La réunion de planification de sprint est divisée en deux parties :
 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.
- **Réunion de revue de sprint** — 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 incrément 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 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*. 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.

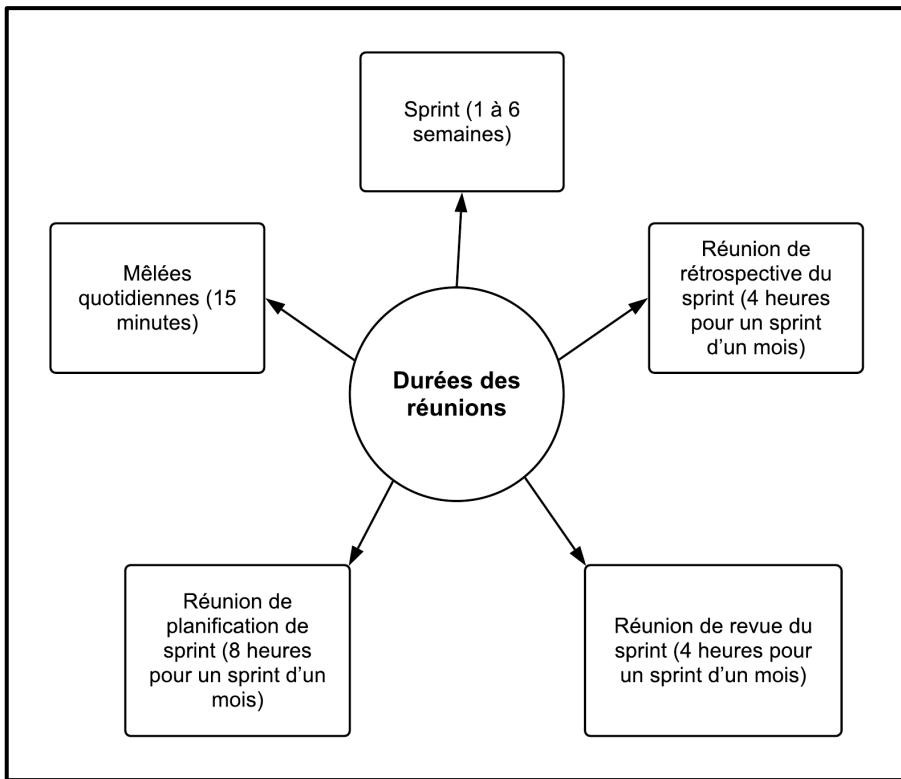


Figure 2-8 : Durées des réunions Scrum

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'*affinage du backlog de produit 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 incrémentés et de la valeur ajoutée livrée lors de chaque sprint. Les processus d'*identification des tâches, d'estimation des tâches et de création du backlog de sprint* permettent ensuite de produire le backlog de sprint que l'équipe utilisera pour développer les incrémentés de produit.

Pour chaque sprint, le processus de *création des incrémentés 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 incrémentés 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 incrémentés 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.

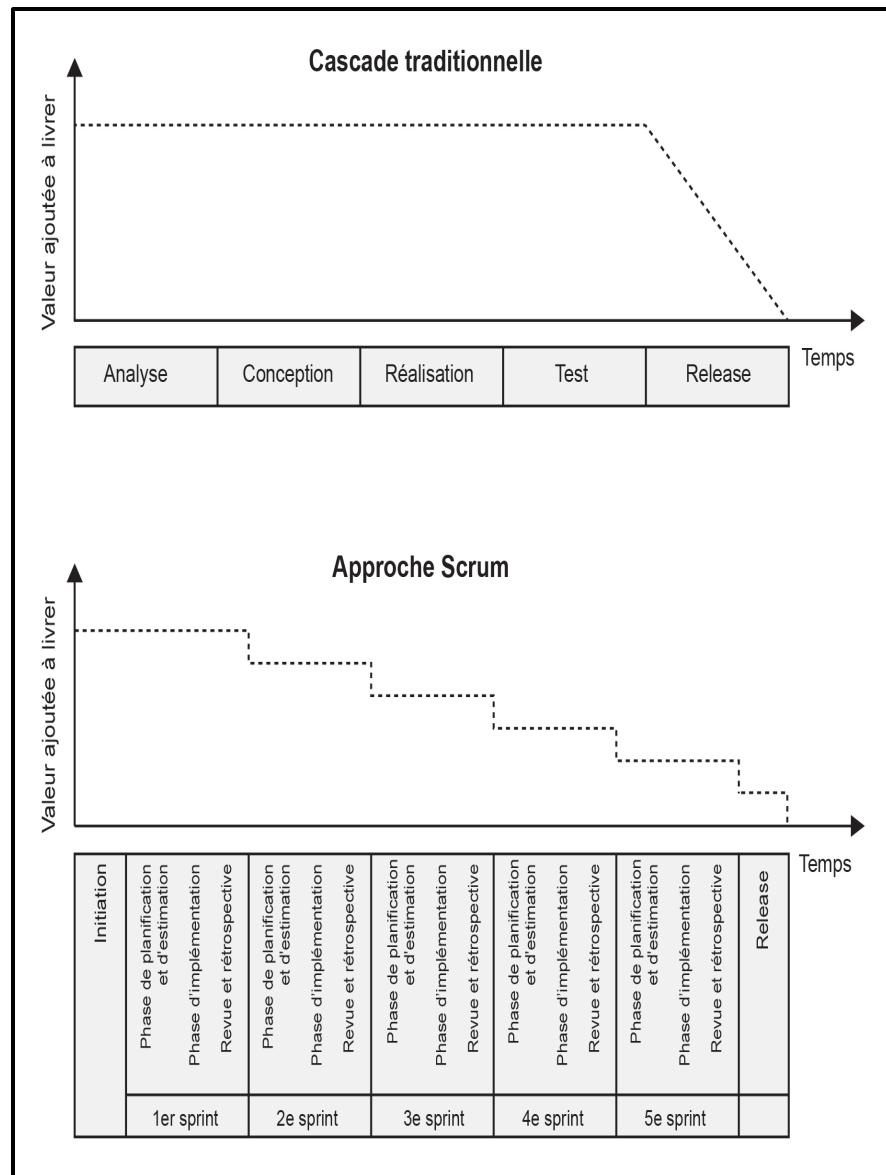


Figure 2-9 : L'approche Scrum comparée à l'approche en cascade traditionnelle

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 incrément 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'incréments 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
- 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.

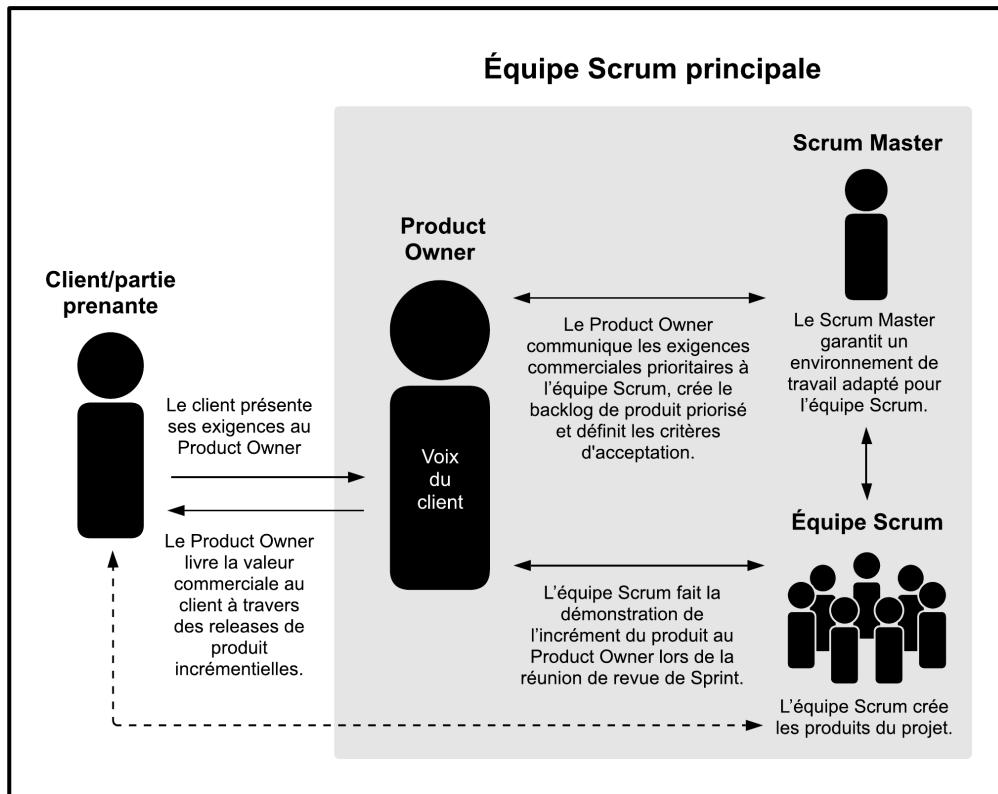


Figure 3-1 : Aperçu des rôles Scrum

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

Parties prenantes est un terme collectif qui inclut les clients, utilisateurs et 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. Les parties prenantes influencent le projet tout au long de son développement. Les parties prenantes peuvent également jouer un rôle durant les processus de développement d'épics, de création du backlog de produit priorisé, de planification de la release, de rétrospective de sprint et d'autres processus Scrum importants.

- **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.

2. 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.

3. 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 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 les parties prenantes, 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, 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 des parties prenantes	<ul style="list-style-type: none"> Aide à l'identification du Scrum Master pour le projet Identifie les parties prenantes
8.3 Formation de l'équipe Scrum	<ul style="list-style-type: none"> Aide à l'identification des membres 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'épics	<ul style="list-style-type: none"> Crée des épics et personnages
8.5 Création du backlog de produit priorisé	<ul style="list-style-type: none"> Priorise les éléments du backlog de produit priorisé Établit la définition de <i>terminé</i>
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"> Aide à 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 Création du backlog de sprint	<ul style="list-style-type: none"> Clarifie les exigences auprès de l'équipe Scrum lors de la création du backlog de sprint
10.1 Création des incrémentums de produit	<ul style="list-style-type: none"> Clarifie les exigences commerciales auprès de l'équipe Scrum

10.3 Affinage du backlog de produit priorisé	<ul style="list-style-type: none"> Affine le backlog de produit priorisé
11.1 Démonstration et validation du sprint	<ul style="list-style-type: none"> Accepte/rejecte les incrément de produit Offre le feedback nécessaire au Scrum Master et à l'équipe Scrum Maintient le calendrier de la release et le backlog de produit priorisé à jour
12.1 Livraison des produits	<ul style="list-style-type: none"> Contribue au et coordonne le déploiement de la release du produit avec le client
12.2 Rétrospective du projet	<ul style="list-style-type: none"> Participe aux réunions de rétrospective du sprint

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.

3.4.1 Voix du client (VOC)

En tant que représentant du client, 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 incrément de produit pour le client.

3.4.2 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 les parties prenantes. 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.

3.4.3 Product Owner du programme

Le Product Owner du programme définit les objectifs stratégiques et les priorités du programme. Il est chargé d'optimiser la valeur commerciale d'un programme en exprimant clairement les exigences du client et en soutenant la justification commerciale du programme. Le Product Owner du programme gère également le backlog de produit du programme.

Il interagit avec le Product Owner du portefeuille afin d'assurer l'adéquation entre le programme et les buts et objectifs du portefeuille. Il participe également à la nomination des Product Owners pour les projets individuels et assure que la vision, les objectifs, résultats et releases des projets individuels sont conformes à ceux du programme.

Ce rôle est similaire à celui du Product Owner sauf qu'il répond aux besoins du programme ou de l'unité commerciale plutôt qu'à ceux d'une unique équipe Scrum.

Il est recommandé que les Product Owners de programme consultent les sections du *Guide SBOK™* définies dans le guide des rôles pour le Product Owner ainsi que le chapitre 14 qui décrit le passage à l'échelle de Scrum pour les entreprises.

3.4.4 Product Owner du portefeuille

Le Product Owner du portefeuille prend les décisions au niveau du portefeuille. Il possède la meilleure perspective afin de décider comment organiser l'entreprise et atteindre la vision. Le rôle de Product Owner de portefeuille est similaire à celui du Product Owner de programme. Il est responsable de et gère la création et l'affinage du backlog de produit du portefeuille.

Ce rôle est similaire à celui du Product Owner sauf qu'il répond aux besoins du portefeuille ou de l'unité commerciale plutôt qu'à ceux d'une unique équipe Scrum.

Il est recommandé que les Product Owners de portefeuille consultent les sections du *Guide SBOK™* définies dans le guide des rôles pour le Product Owner ainsi que le chapitre 14 qui décrit le passage à l'échelle de Scrum pour les entreprises.

3.5 Scrum Master

Le Scrum Master est le « leader serviteur » de l'équipe Scrum. En tant que coach, il modère, facilite et motive les interactions de l'équipe. 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	<ul style="list-style-type: none"> Participe à l'identification des parties prenantes 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 à créer le backlog de produit priorisé et à établir la définition de <i>terminé</i>
8.6 Planification de la release	<ul style="list-style-type: none"> Coordonne la création du calendrier de planification de la release Détermine la durée du sprint
9.1 Création des user stories	<ul style="list-style-type: none"> Assiste l'équipe Scrum lors de 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 Création du backlog de sprint	<ul style="list-style-type: none"> Assiste l'équipe Scrum lors du développement du backlog de sprint et du sprint burndown chart
10.1 Crédit des incrément de produit	<ul style="list-style-type: none"> Supporte l'équipe Scrum lors de la création des incrément de produits décidés pour le sprint Aide à maintenir le scrumboard et le registre des obstacles à jour
10.2 Mélées quotidiennes	<ul style="list-style-type: none"> S'assure que le scrumboard et le registre des obstacles est maintenu à jour
10.3 Affinage du backlog de produit 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 incrément 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 du projet	<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

3.5.1 Chief Scrum Master

Les gros projets requièrent plusieurs équipes Scrum travaillant en parallèle. Il se peut que les informations collectées par une équipe doivent d'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 3-2 présente les questions posées lors d'une réunion Scrum of Scrums (SoS).

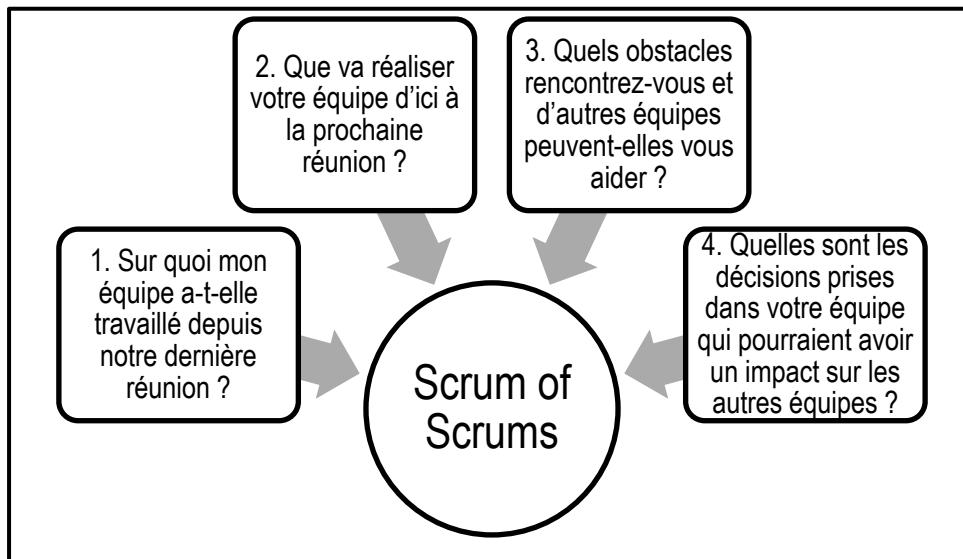


Figure 3-2 : Questions posées lors d'une réunion Scrum of Scrums

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.

3.5.2 Scrum Master du programme

Le Scrum Master du programme est un facilitateur qui s'assure que toutes les équipes du projet disposent d'un environnement favorable à la réussite de leurs projets. Il guide, facilite et enseigne les pratiques Scrum à chaque personne impliquée dans le programme. Il donne des conseils aux Scrum Masters des projets individuels, élimine les obstacles pour les différentes équipes du projet, collabore avec le Scrum Guidance Body afin de définir les objectifs associés à la qualité, aux règlements gouvernementaux, à la sécurité, et à d'autres paramètres organisationnels clés et assure que les processus Scrum sont suivis de façon efficace à travers le programme.

Le Scrum Master du programme interagit avec le Scrum Master du portefeuille afin d'assurer l'adéquation entre le programme et les buts et objectifs du portefeuille. Il participe également à la nomination des Scrum Masters pour les projets individuels et assure que la vision, les objectifs, résultats et releases des projets individuels au sein du programme sont conformes à ceux du programme dans son ensemble.

Ce rôle est similaire à celui du Scrum Master sauf qu'il répond aux besoins du programme ou de l'unité commerciale plutôt qu'à ceux d'une unique équipe Scrum.

Il est recommandé que les Scrum Masters de programme consultent les sections du *Guide SBOK™* définies dans le guide des rôles pour le Scrum Master ainsi que le chapitre 14 qui décrit le passage à l'échelle de Scrum pour l'entreprise.

3.5.3 Scrum Master du portefeuille

Ce rôle est similaire à celui du Scrum Master sauf qu'il répond aux besoins du portefeuille ou de l'unité commerciale plutôt qu'à ceux d'une unique équipe Scrum.

Il est recommandé que les Scrum Masters de portefeuille consultent les sections du *Guide SBOK™* définies dans le guide des rôles pour le Scrum Master ainsi que le chapitre 14 qui décrit le passage à l'échelle de Scrum pour l'entreprise.

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. Elle consiste en un groupe d'individus qui travaillent sur les user stories du backlog de sprint afin de créer les incrémentés de produit 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"> Validé 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 Crédit du backlog de sprint	<ul style="list-style-type: none"> Développe le backlog de sprint et le sprint burndown chart
10.1 Crédit des incrémentations de produit	<ul style="list-style-type: none"> Crée les incrémentations de produit Identifie les risques et met en œuvre les actions de limitation des risques, le cas échéant Met à jour le registre des obstacles et les dépendances
10.2 Mélées quotidiennes	<ul style="list-style-type: none"> Met à jour le burndown chart, le scrumboard et le registre des obstacles Discute des problèmes rencontrés par les membres individuels et cherche des solutions pour motiver l'équipe Identifie les risques, le cas échéant Présente des demandes de changement, le cas échéant
10.3 Affinement du backlog de produit priorisé	<ul style="list-style-type: none"> Participe aux réunions de revue du backlog de produit priorisé
11.1 Démonstration et validation du sprint	<ul style="list-style-type: none"> Fait la démonstration des incrémentations de produit terminés pour approbation par le Product Owner
11.2 Rétrospective du sprint	<ul style="list-style-type: none"> Identifie les opportunités d'amélioration sur la base du sprint en cours, le cas échéant, et convient des mesures d'amélioration à mettre en œuvre lors du prochain sprint.
12.2 Rétrospective du projet	<ul style="list-style-type: none"> Participe à la réunion de rétrospective du projet

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

3.6.1 Sélection du personnel

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

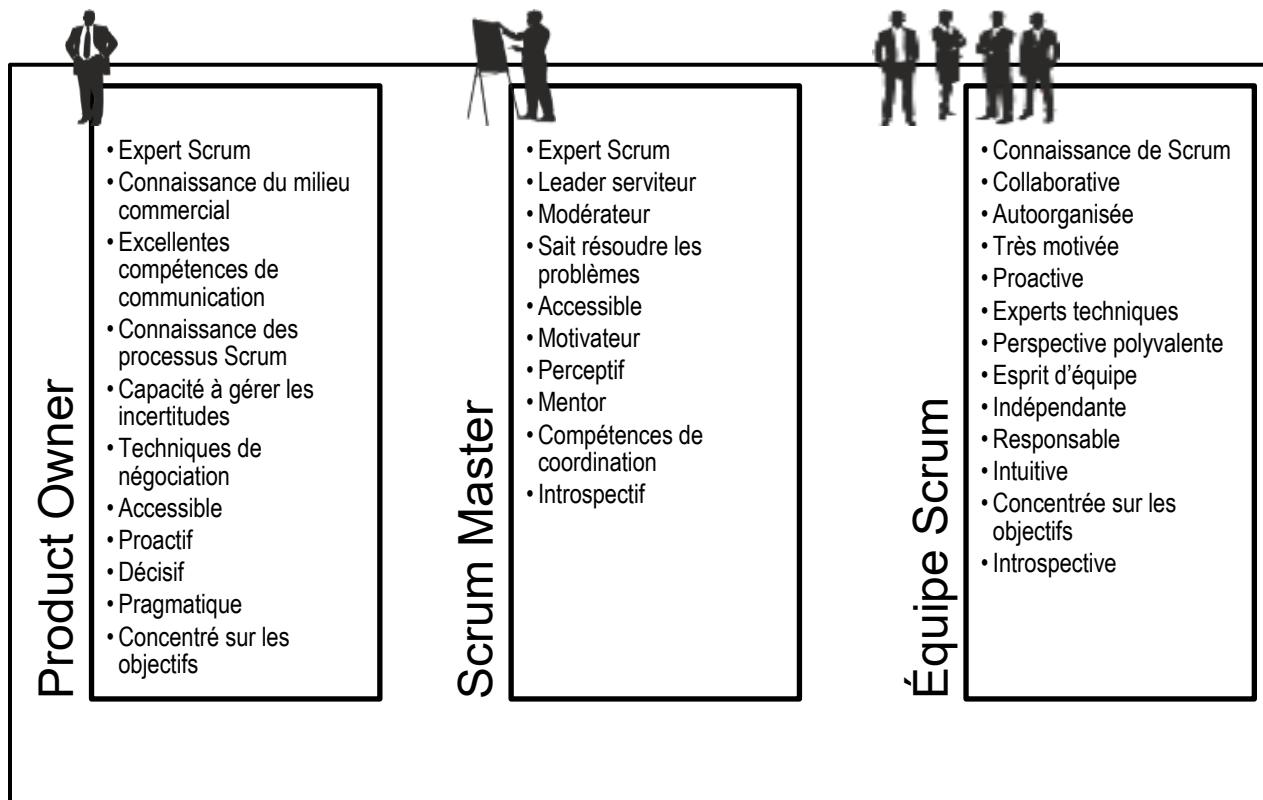


Figure 3-3 : Traits désirables pour les rôles Scrum principaux

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. Les principaux avantages d'une équipe de six à dix membres sont que la communication au sein de l'équipe et sa gestion sont normalement simples et requièrent un effort minimal. 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 Projets, programmes et portefeuilles Scrum

3.7.1 Définition de projet, programme et portefeuille

- **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 personnels et de capacité organisationnelle. L'objectif de l'équipe du projet est de créer les incrémentés de produit tels qu'ils sont définis dans le backlog de produit priorisé.
- **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.
- **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.

Les exemples suivants illustrent des projets, programmes et portefeuilles dans différents secteurs d'activité :

Exemple 1 : Entreprise de construction

- Projet — construction d'une maison
- Programme — construction d'un lotissement résidentiel
- Portefeuille — tous les projets résidentiels de l'entreprise

Exemple 2 : Organisation aérospatiale

- Projet — construction d'un lanceur
- Programme — lancement réussi d'un satellite
- Portefeuille — tous les programmes satellites actifs

Exemple 3 : Entreprise informatique

- Projet — développement d'un module de panier virtuel
- Programme — développement d'un site de commerce en ligne entièrement opérationnel
- Portefeuille — tous les sites Web développés par l'entreprise jusqu'à présent

La figure 3-4 illustre la manière dont Scrum peut être utilisé dans les organisations pour les portefeuilles, programmes ou projets.

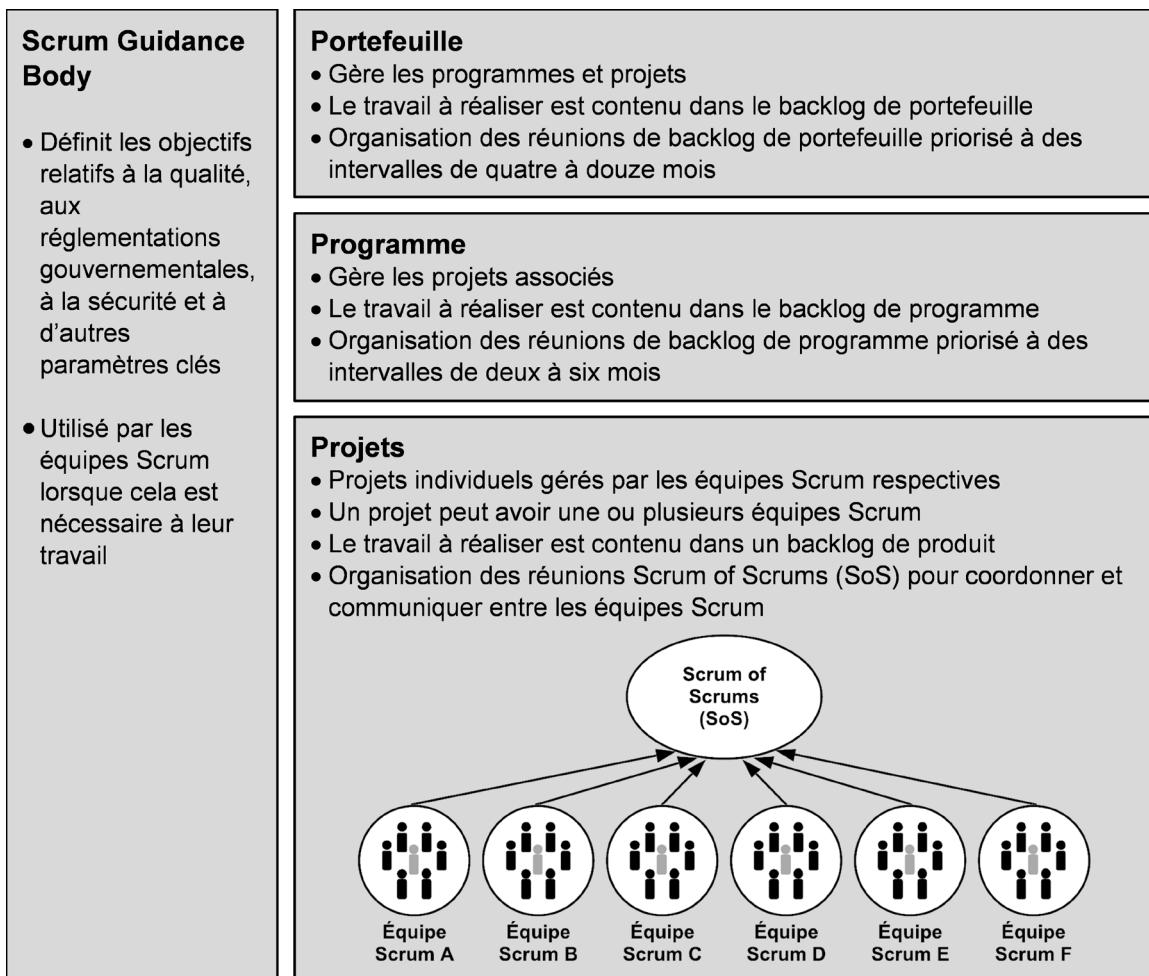


Figure 3-4 : Scrum dans les organisations pour les projets, programmes ou portefeuilles

3.7.1.1 Travailler avec des équipes de programme ou de portefeuille

Lorsque Scrum est utilisé pour gérer des projets dans le cadre d'un programme ou d'un portefeuille, il est fortement recommandé de respecter les principes généraux de Scrum tels qu'ils sont présentés dans cette

publication. Cependant, des ajustements mineurs des outils et de la structure organisationnelle peuvent être requis afin de s'adapter aux activités globales et aux interdépendances du programme ou du portefeuille. S'il existe un Scrum Guidance Body, celui-ci pourra être responsable de l'organisation à différents niveaux afin de comprendre et de définir la mise en œuvre adéquate de Scrum et pourra agir comme organisme de conseil pour toute personne travaillant sur un projet, programme ou portefeuille.

Les portefeuilles et les programmes disposent de différentes équipes avec différents objectifs. Les équipes de gestion de programme tendent à fournir les capacités et réaliser certains buts pour contribuer à la réalisation des objectifs spécifiques du programme. En revanche, l'équipe de portefeuille doit équilibrer les objectifs des différents programmes afin de réaliser les objectifs stratégiques de l'organisation dans son ensemble. Le chapitre 14 aborde dans les détails le passage à l'échelle de Scrum pour l'entreprise.

3.7.1.2 Gérer la communication des équipes de programme et de portefeuille

Les questions et problèmes rencontrés lorsque Scrum est appliqué à un programme ou portefeuille impliquent principalement la coordination entre plusieurs équipes. Ces problèmes peuvent conduire à des échecs s'ils ne sont pas correctement gérés. Les outils de communication doivent être mis à l'échelle afin de répondre aux exigences des nombreuses équipes impliquées dans un programme ou portefeuille. Chaque équipe Scrum doit non seulement gérer ses communications internes mais aussi ses communications externes avec les autres équipes et avec les parties prenantes impliquées dans le programme ou le portefeuille.

3.7.2 Maintenir la participation des parties prenantes

L'approche Scrum requiert le soutien total des parties prenantes du projet. C'est le Product Owner qui est responsable d'encourager l'implication des parties prenantes. Les actions suivantes sont recommandées afin de maintenir le soutien et la participation des parties prenantes :

- garantir la collaboration efficace et la participation des parties prenantes dans le projet
- évaluer l'impact commercial de façon continue
- maintenir une communication régulière avec les parties prenantes
- gérer les attentes des parties prenantes

Le sponsor est l'une des principales parties prenantes — c'est l'individu ou l'organisation qui fournit les fonds et les autres ressources d'un projet. Les sponsors cherchent à comprendre les résultats financiers relatifs à un produit ou service et sont normalement plus préoccupés par les résultats finaux que par les tâches individuelles.

Il est important que les sponsors qui financent le projet soient au fait des éléments suivants :

- les avantages relatifs à l'application de Scrum
- les échéances et coûts estimés des projets Scrum
- les risques globaux impliqués par les projets Scrum et les mesures prises pour les atténuer
- les dates prévues pour les releases et les produits finaux

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 • 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	<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 incrémentés 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 les parties prenantes 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.

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.

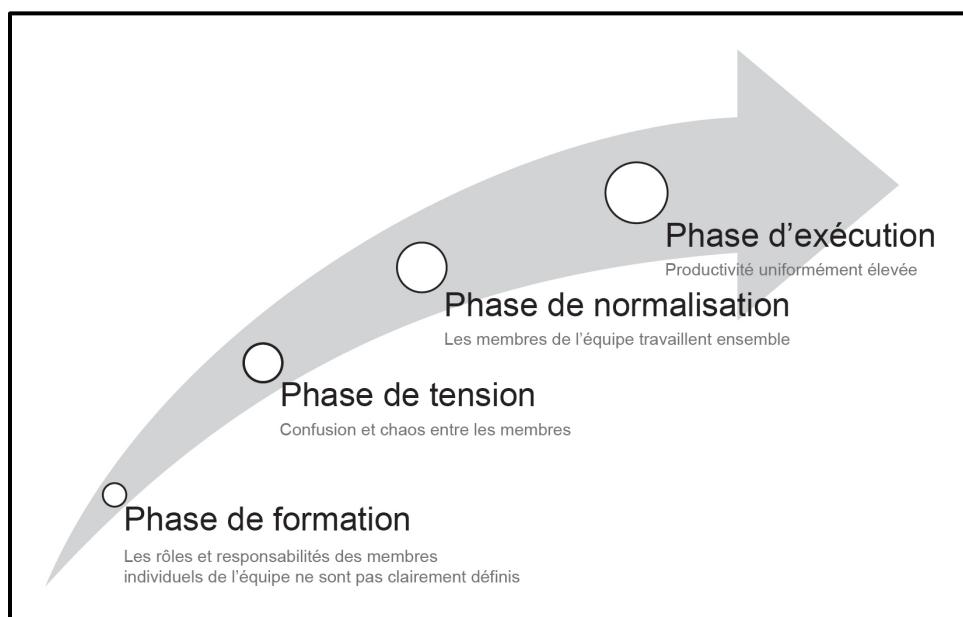


Figure 3-5 : Les phases du modèle de Tuckman sur le développement des groupes

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 mélée 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 mélées 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 serviteur** — les leaders serviteurs 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 serviteurs 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 serviteur

Le style de leadership préféré pour les projets Scrum est le leadership serviteur. À l'origine, ce terme a été développé par Robert K. Greenleaf dans un essai intitulé *The Servant as Leader* (« Le Serviteur en tant que Leader »). L'exemple ci-dessous explique son concept :

Le leader serviteur est avant tout un serviteur... Cela commence avec le sentiment naturel de vouloir servir et servir avant tout. Puis le choix conscient amène à vouloir mener. Cette personne

est différente de celle qui veut avant tout être *leader*, peut-être à cause du besoin d'assouvir une soif de pouvoir ou d'acquérir des possessions matérielles... Le leader avant tout et le serviteur avant tout représentent deux extrêmes. Entre eux il existe des nuances et des mélanges qui représentent les variations infinies de la nature humaine...

La différence se manifeste dans l'attention portée par les serviteurs avant tout à s'assurer que les besoins les plus importants pour les autres personnes sont servis. Le meilleur test, et difficulté à gérer, est : Ceux qui sont servis évoluent-ils en tant qu'individus ? Développent-ils, *alors qu'ils sont servis*, une meilleure santé, une plus grande sagesse, plus de liberté et d'autonomie, une plus grande aptitude à servir eux-mêmes ? *Et*, quel est l'effet sur les moins privilégiés dans la société ? En tireront-ils les bénéfices, ou ne seront-ils du moins pas davantage défavorisés ? (Greenleaf 1970, 6)

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 serviteurs 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 serviteurs 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 serviteurs. Les leaders serviteurs 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 serviteurs. 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 serviteurs 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 serviteurs 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 serviteurs.

7. **L'anticipation** — leur esprit intuitif permet aux leaders serviteurs 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 serviteurs 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 serviteurs 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 serviteurs 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 serviteurs 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.

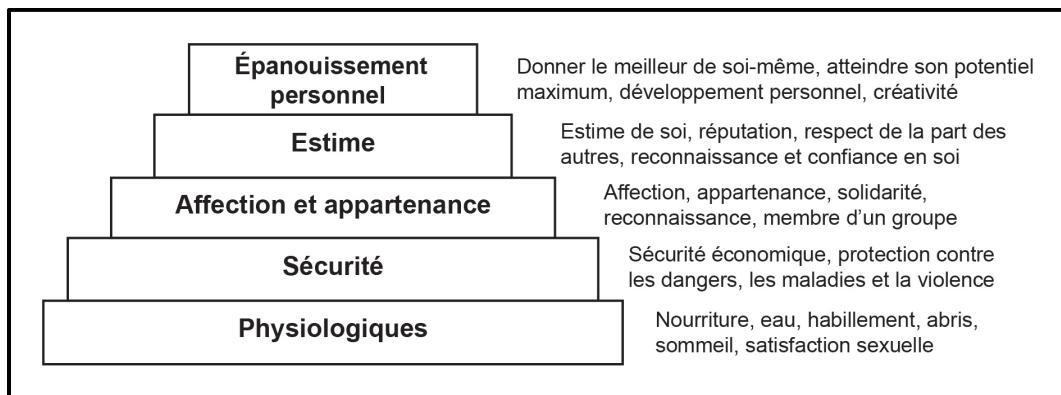


Figure 3-6 : La pyramide des besoins de Maslow

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.

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

4

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.

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
- 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 les parties prenantes 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 les parties prenantes du projet pour leur montrer les incrémentés de produit à la fin de chaque sprint et permettre la gestion efficace des changements.
3. créer des *incrémentés de produit* qui s'appuient sur les priorités établies en produisant des incrémentés 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 les parties prenantes 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.

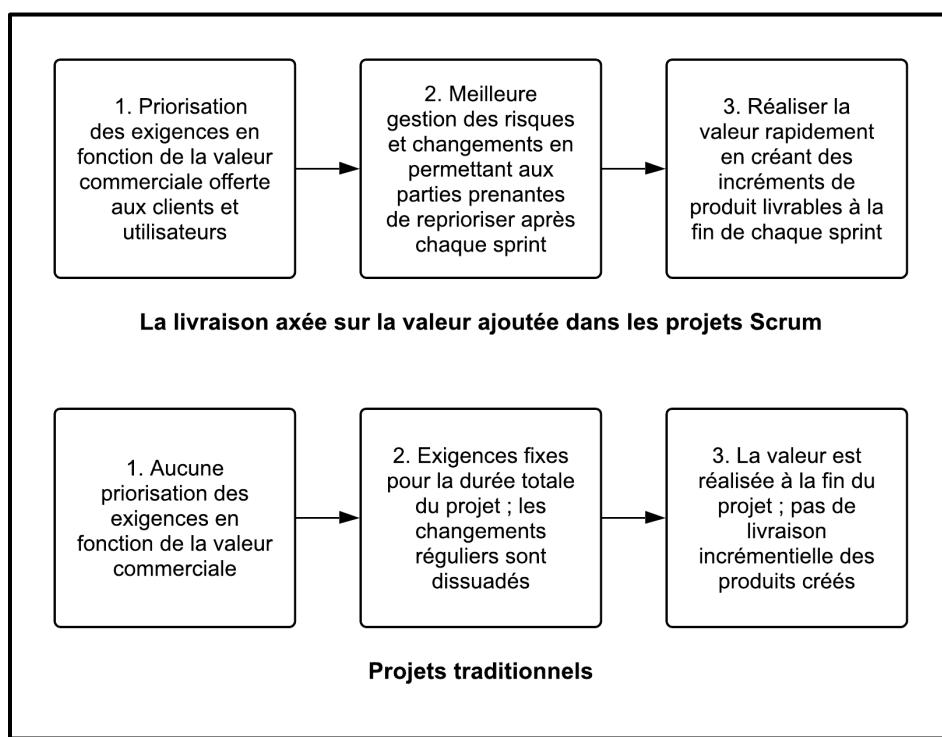


Figure 4-1 : Livraison de la valeur ajoutée dans Scrum comparée aux projets traditionnels

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 portfolio. 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

Figure 4-2 : 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 incrémentés 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 les parties prenantes.
4. Le **Scrum Master** facilite la création d'incrémentés 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 incrémentés de produit et avec le Product Owner et les autres parties prenantes afin de garantir que les bénéfices du projet sont réalisés.
5. L'**équipe Scrum** travaille à la création des incrémentés de produit du projet et contribue à la création de la valeur commerciale pour toutes les parties prenantes 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 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 évènement 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 *mélées quotidiennes* ou de l'*affinage du backlog de produit 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 du projet*.

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

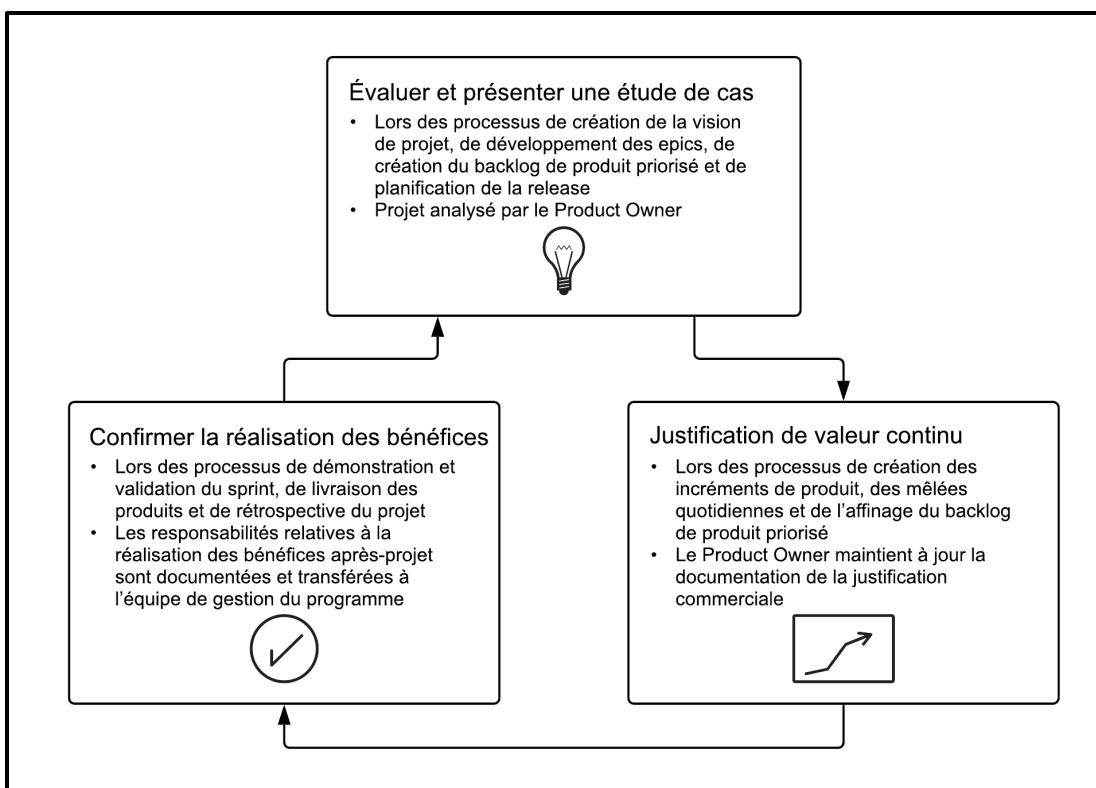


Figure 4-3 : Justification commerciale et cycle du projet

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 :

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

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 (sponsor, clients et/ou utilisateurs), tandis que l'équipe Scrum se concentre sur ce qui doit être développé. Certains outils courants recommandées par le Scrum Guidance Body incluent notamment :

1. La cartographie de la chaîne de valeur

La cartographie de la chaîne de valeur utilise des organigrammes afin d'illustrer le flux d'informations nécessaires afin de réaliser un processus. Cette technique peut être utilisée afin de rationaliser un processus en aidant à déterminer les éléments qui n'apportent pas de valeur ajoutée.

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 4-4 illustre l'analyse Kano.

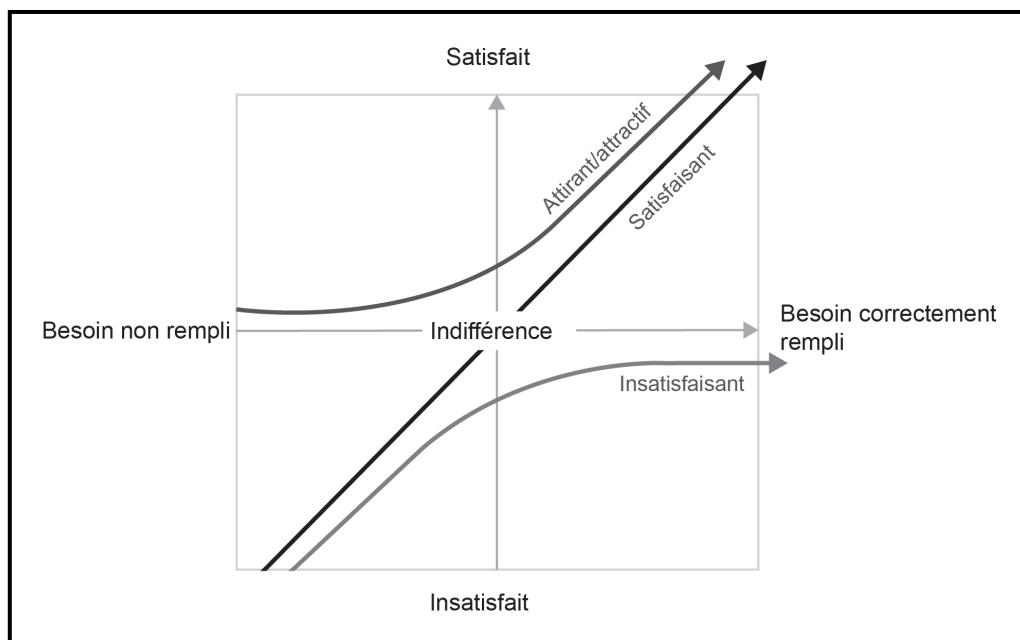


Figure 4-4 : Analyse Kano

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

Cette technique offre un aperçu visuel d'un produit et de ses éléments principaux. La représentation des stories, formulée pour la première fois pas 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.

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.

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

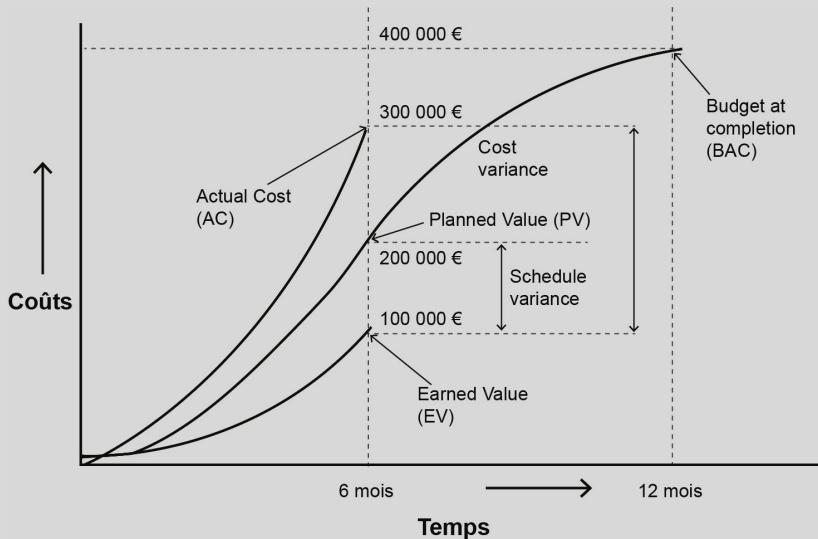
Tableau 4-1 : Formules de valeur acquise

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 avions 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 - AC = 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 4-5 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.

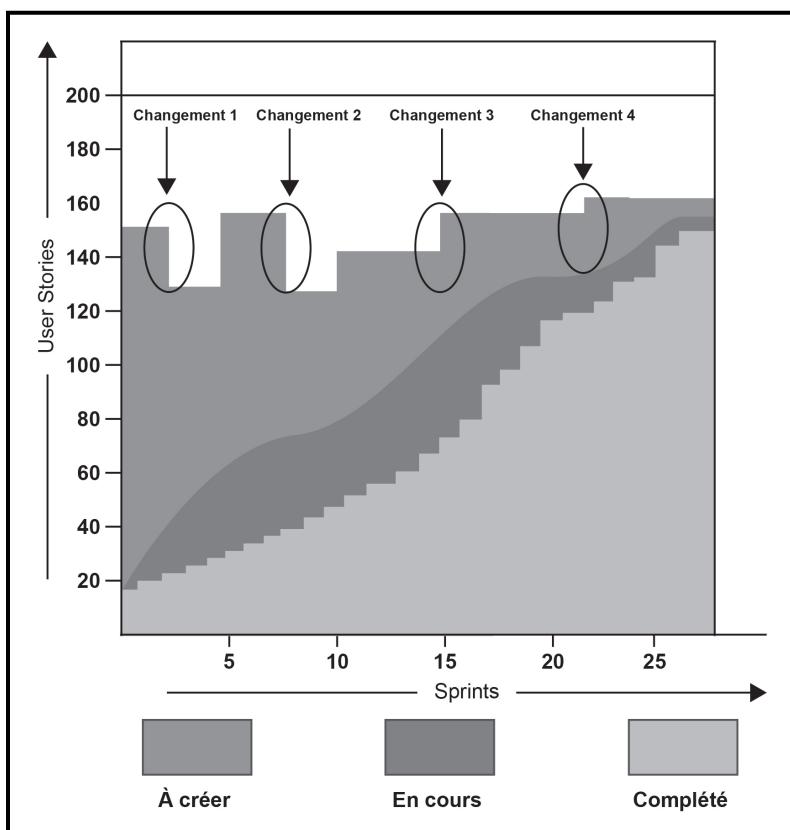


Figure 4-5 : Exemple de diagramme de flux cumulatif (DFC)

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 incrément 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 incrément 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
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	<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 4-2 : 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

5

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
- 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 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 incrémentés 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 les parties prenantes (y compris les clients et les utilisateurs) avec de réels incrémentés 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 incrémentés 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. Pendant le sprint, le Product Owner, le client et l'équipe Scrum peuvent discuter de la liste de fonctionnalités du produit afin de répondre aux besoins changeants du client.

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. 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 incrément 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 à des éléments livrables rejetés sont rajoutées au backlog de produit priorisé mis à jour durant le processus d'*affinage du backlog de produit priorisé*, afin d'être réalisées lors d'un sprint à venir. 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 incrément de produit.

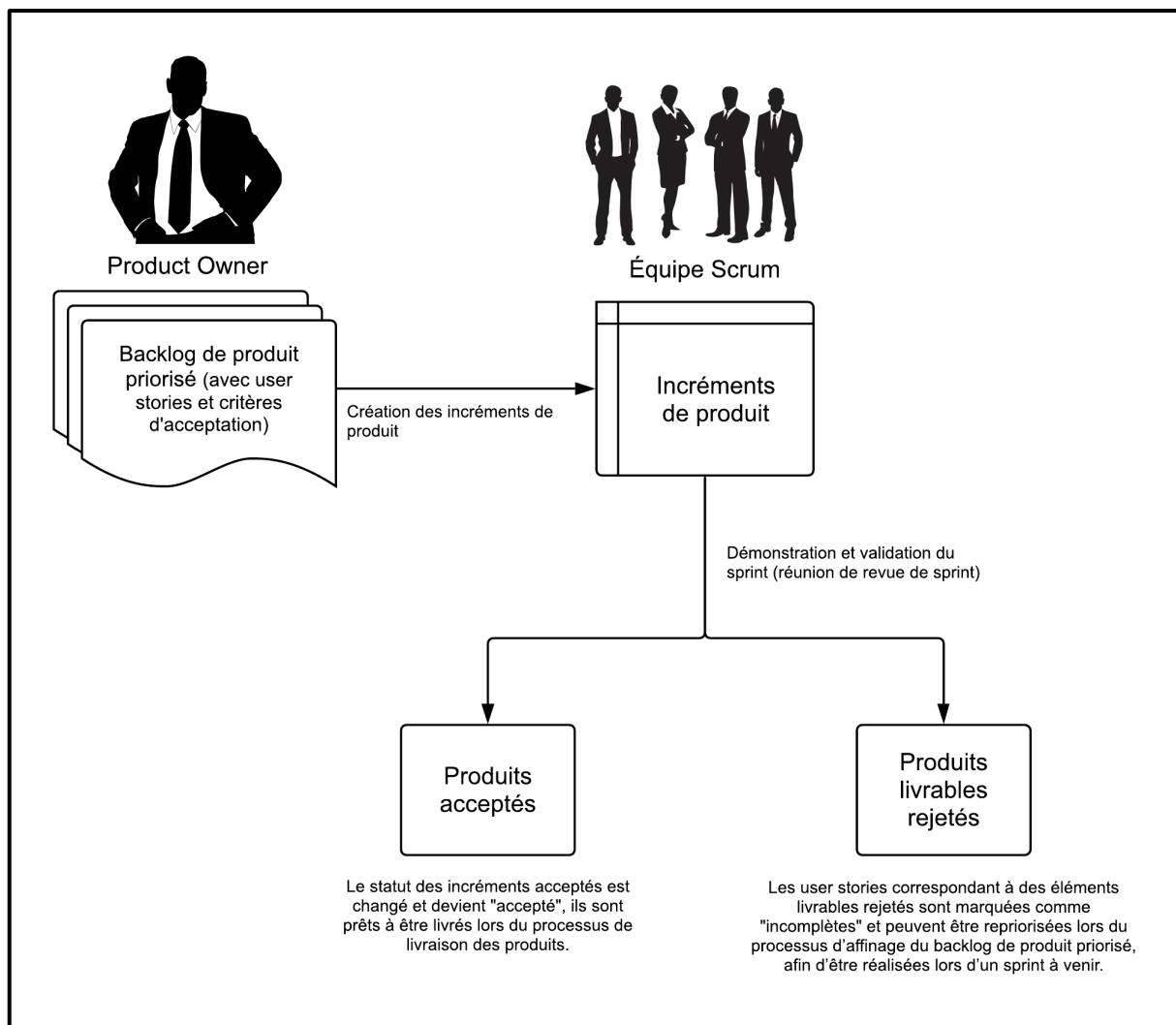


Figure 5-1 : Diagramme de flux des incréments du projet

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 *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 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*. Une définition claire de *terminé* est essentielle car elle permet d'éviter toute ambiguïté et aide l'équipe à respecter les normes de qualité obligatoires. La définition de *terminé* est généralement documentée par le Scrum Guidance Body.

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.

5.4.3 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 conséquent, 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.

5.4.4 Acceptation ou rejet des éléments du backlog de produit priorisé

Vers la fin de toute itération, les unités commerciales et parties prenantes 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 les parties prenantes 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 les parties prenantes 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 incrémentés 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 les parties prenantes 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.

5.5.2 Contrôle et assurance 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 du projet. 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/Swehart (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.

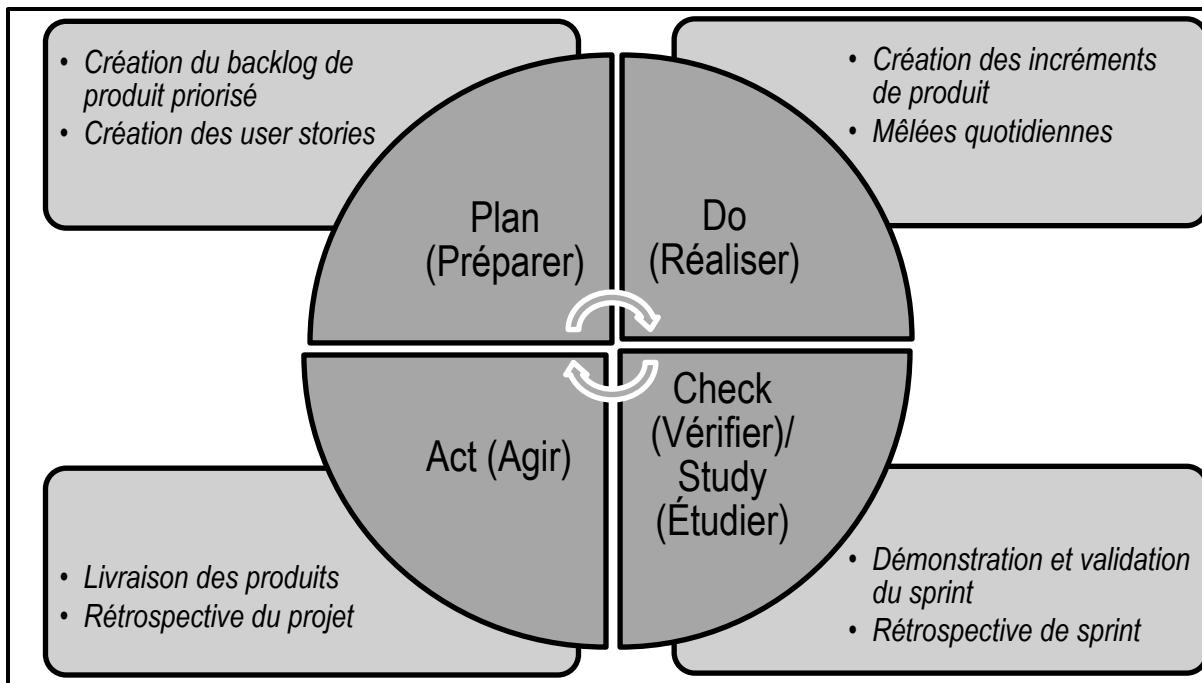


Figure 5-3 : Le cycle PDCA dans Scrum

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
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

	<ul style="list-style-type: none"> • 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"> • Évaluent et acceptent les éléments livrables et le produit final
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 les parties prenantes 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 six 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 essayent 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
- 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 au 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 les parties prenantes (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 les parties prenantes 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 incrémentums 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 adaptive 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. Le Product Owner joue donc un rôle très important dans la gestion des changements pour un projet Scrum.

Les changements qui se trouvent au-delà du degré de tolérance du Product Owner peuvent nécessiter d'être approuvés par les parties prenantes 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 épics*, de *création du backlog de produit priorisé* et d'*affinage du backlog de produit 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.

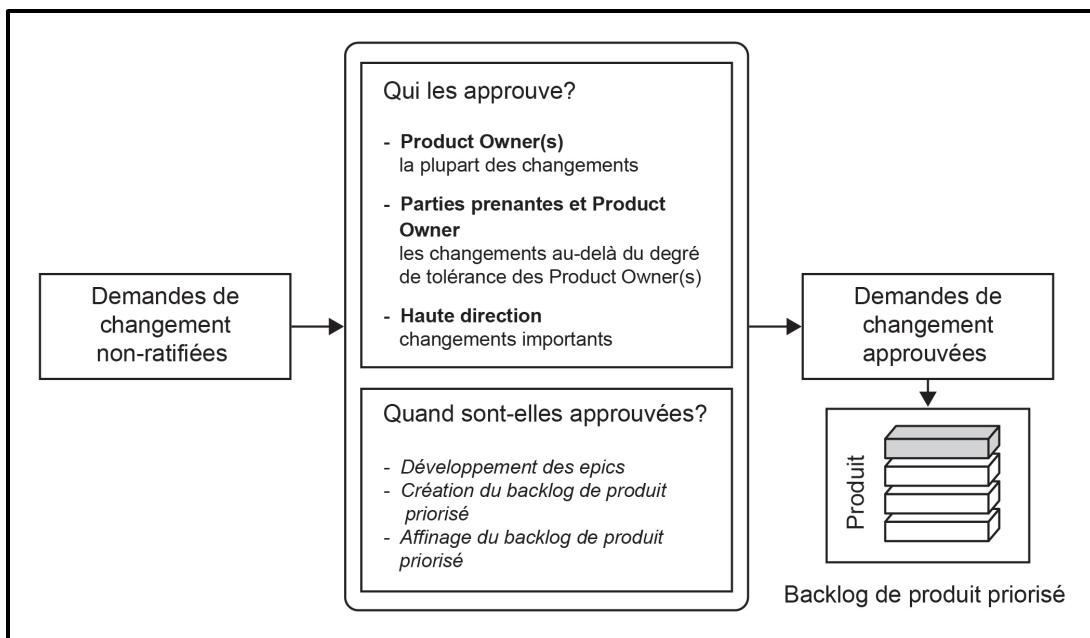


Figure 6-1 : Exemple de processus d'approbation des changements

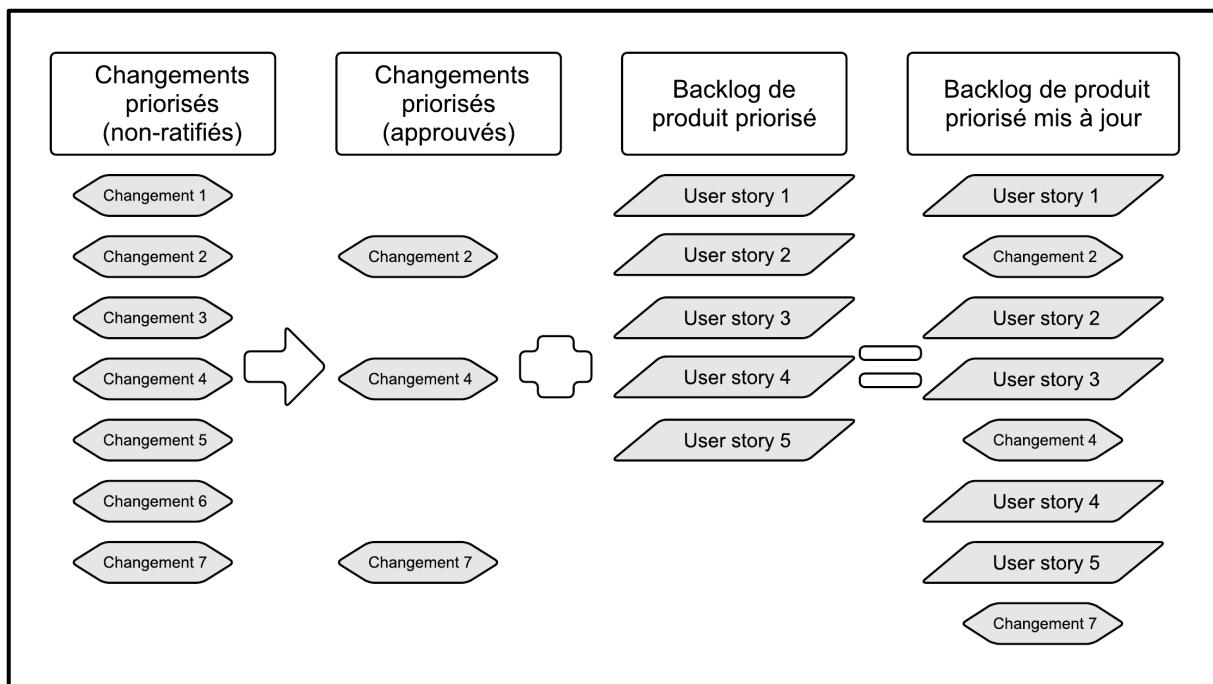


Figure 6-2 : Mise à jour du backlog de produit priorisé avec les changements approuvés

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

Scrum encourage la flexibilité afin de réaliser à terme les résultats commerciaux ayant une plus forte valeur grâce à la *transparence*, à l'*inspection* et à l'*adaptation*. 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.

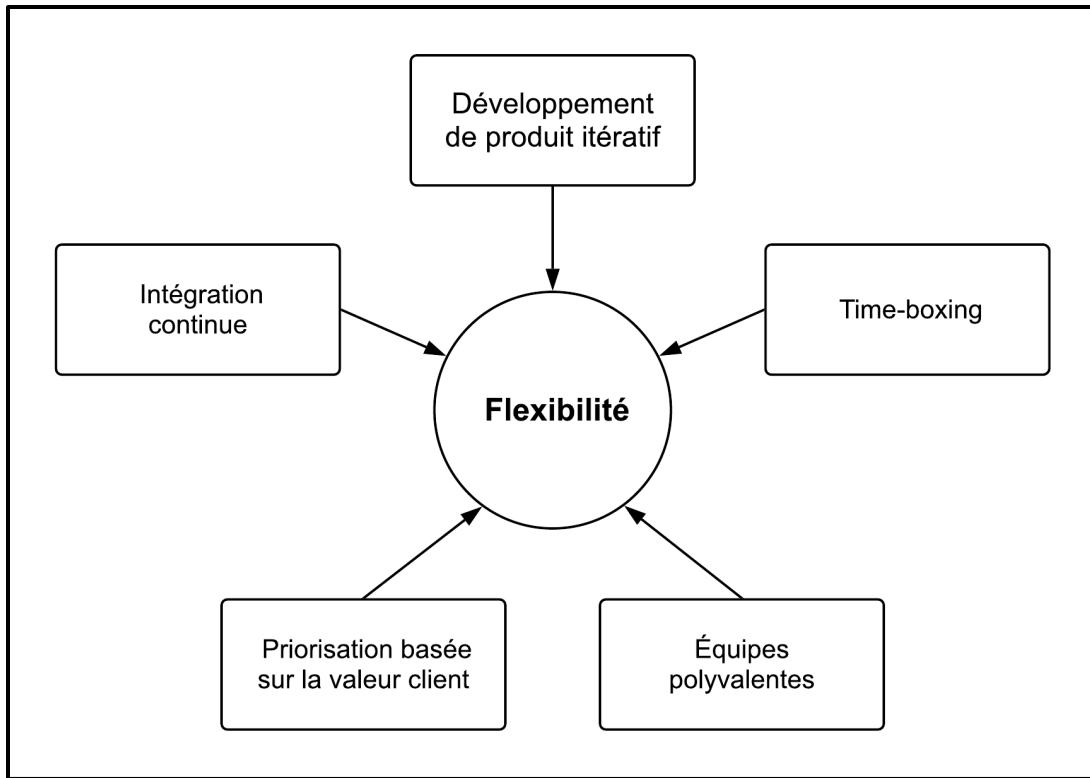


Figure 6-3 : Caractéristiques Scrum pour atteindre la flexibilité

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

Les parties prenantes 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, les parties prenantes 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 du projet*. 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 initient le processus de demande de changement.

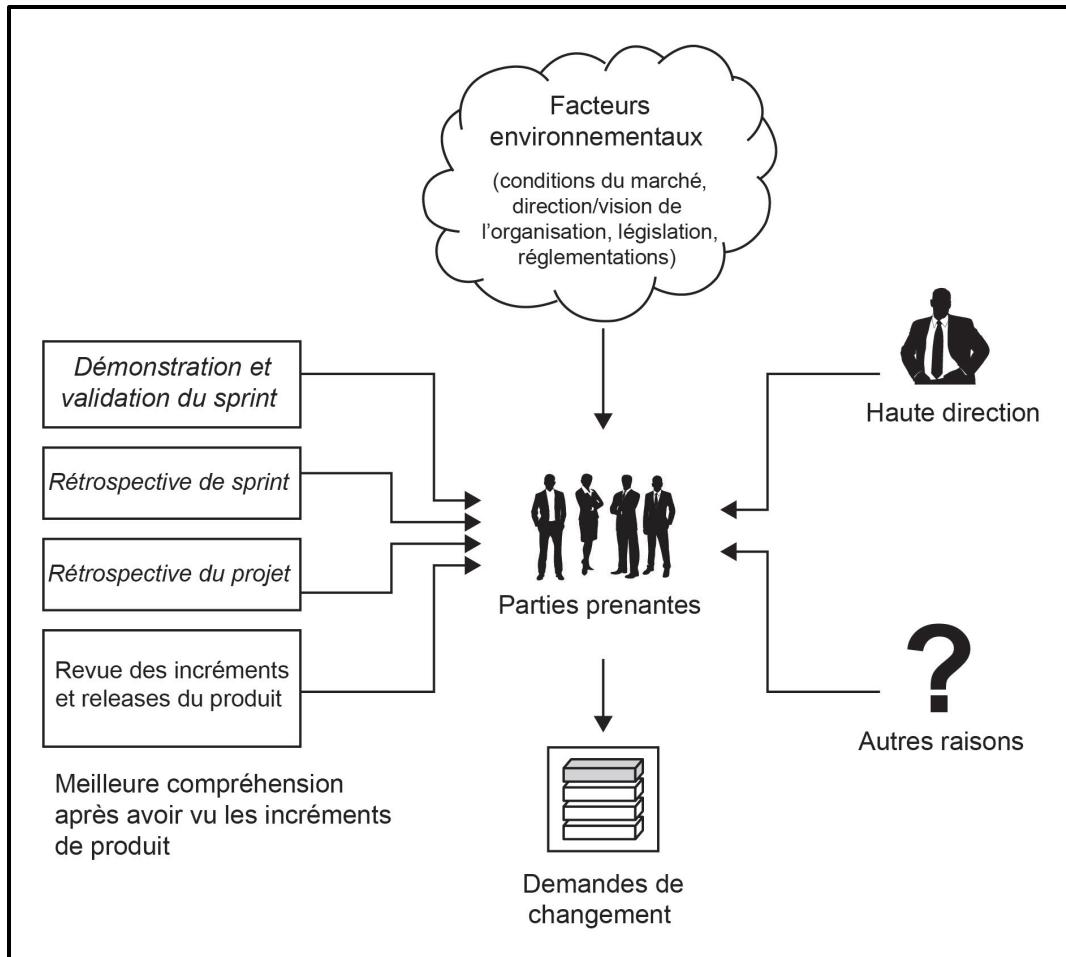


Figure 6-4 : Motivations des parties prenantes pour demander des changements

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 les parties prenantes 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 *mélées 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.

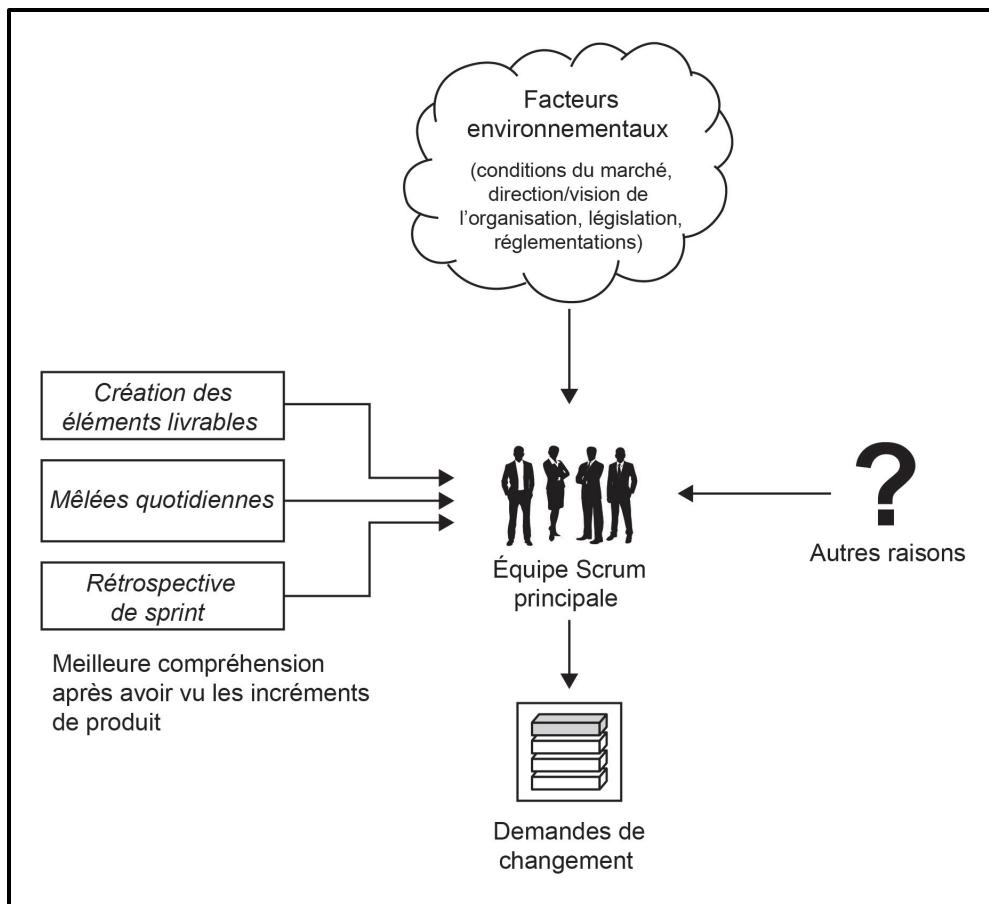


Figure 6-5 : Motivations de l'équipe Scrum principale pour effectuer des demandes de changement

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 adaptive. 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 mélées quotidiennes à 15 minutes et d'établir la durée des sprints entre une et six 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 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

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.

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'*affinage du backlog de produit 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 établit 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'affinage du backlog de produit 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 les parties prenantes 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).

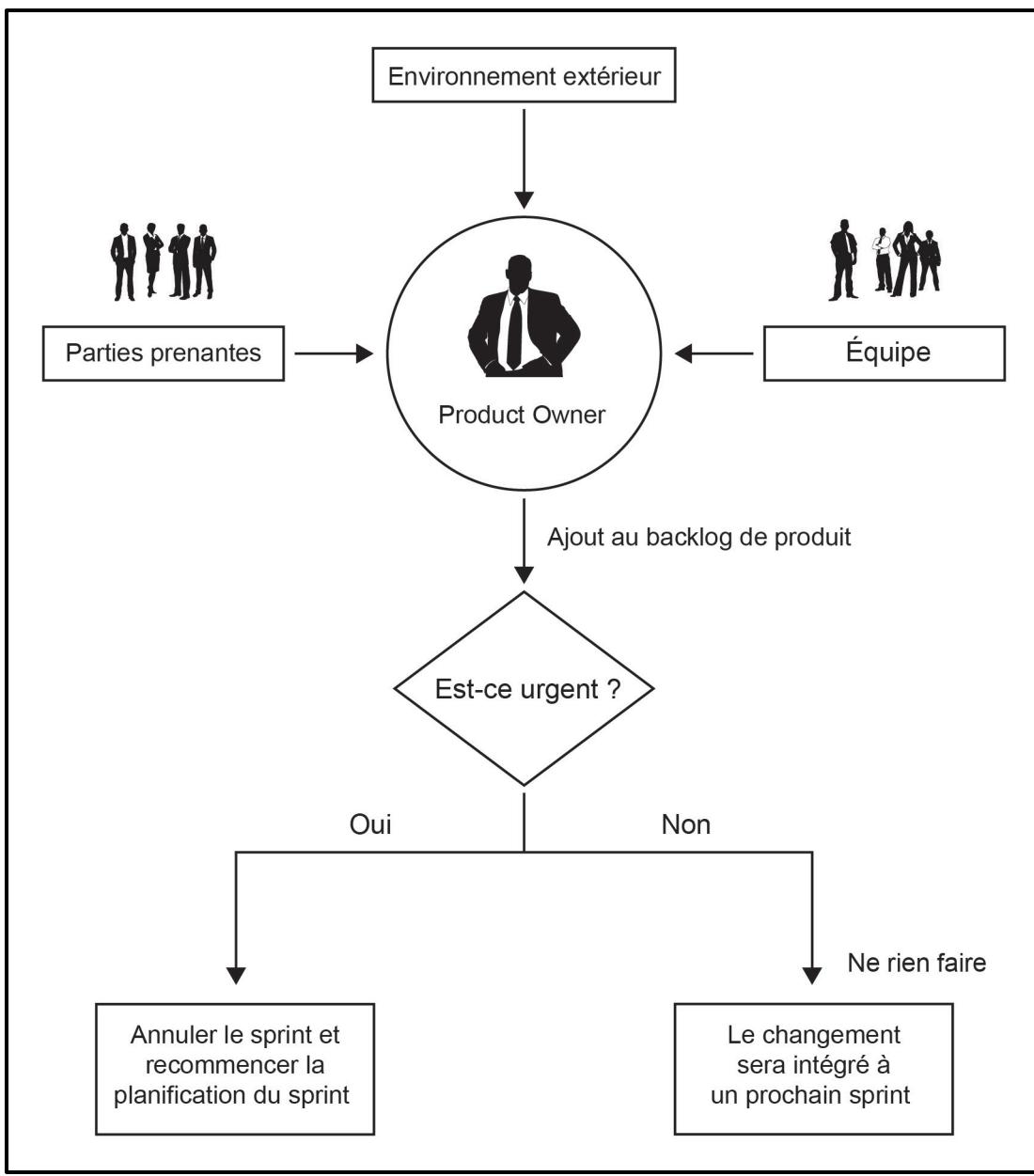


Figure 6-6 : Intégration des changements dans Scrum

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 incrémentations potentiellement livrables à la fin de chaque sprint.

De plus, puisque le Product Owner et les parties prenantes 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'*affinage du backlog de produit 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 associés 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 sprints plus courts et fournir des résultats qui peuvent être évalués par le Product Owner et les parties prenantes 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 des changements prévus sur la durée du sprint.

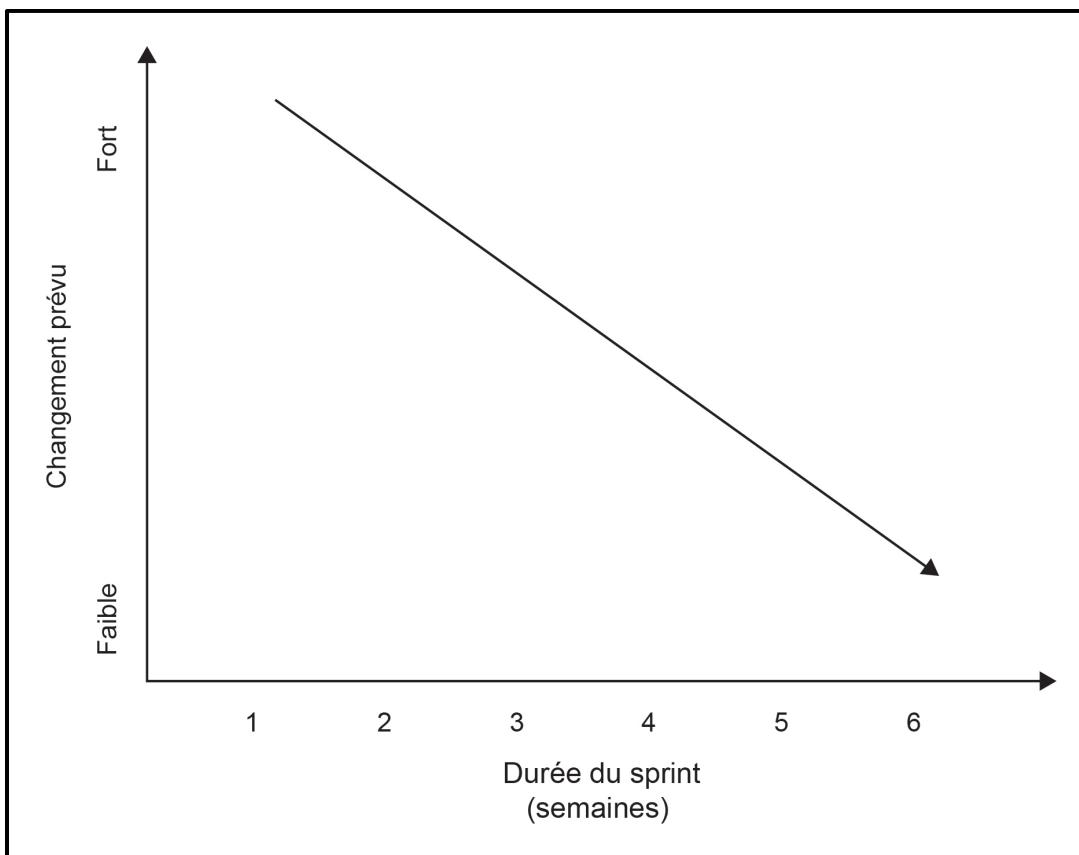


Figure 6-7 : Impact des changements prévus sur la durée du sprint

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 établit 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 backlog de produit 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 les parties prenantes 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

création du backlog de sprint. 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 backlog de produit priorisé)

C'est le Product Owner qui mène la réunion de revue du backlog de produit est tenue lors du processus d'affinage du backlog de produit 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 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 les parties prenantes 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 backlog de produit priorisé est une activité continue pour le Product Owner.

6.5.1.3 Gestion des changements 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 les parties prenantes 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 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 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 illustre la manière dont les changements peuvent être gérés dans le flux Scrum pour les portefeuilles et les programmes.

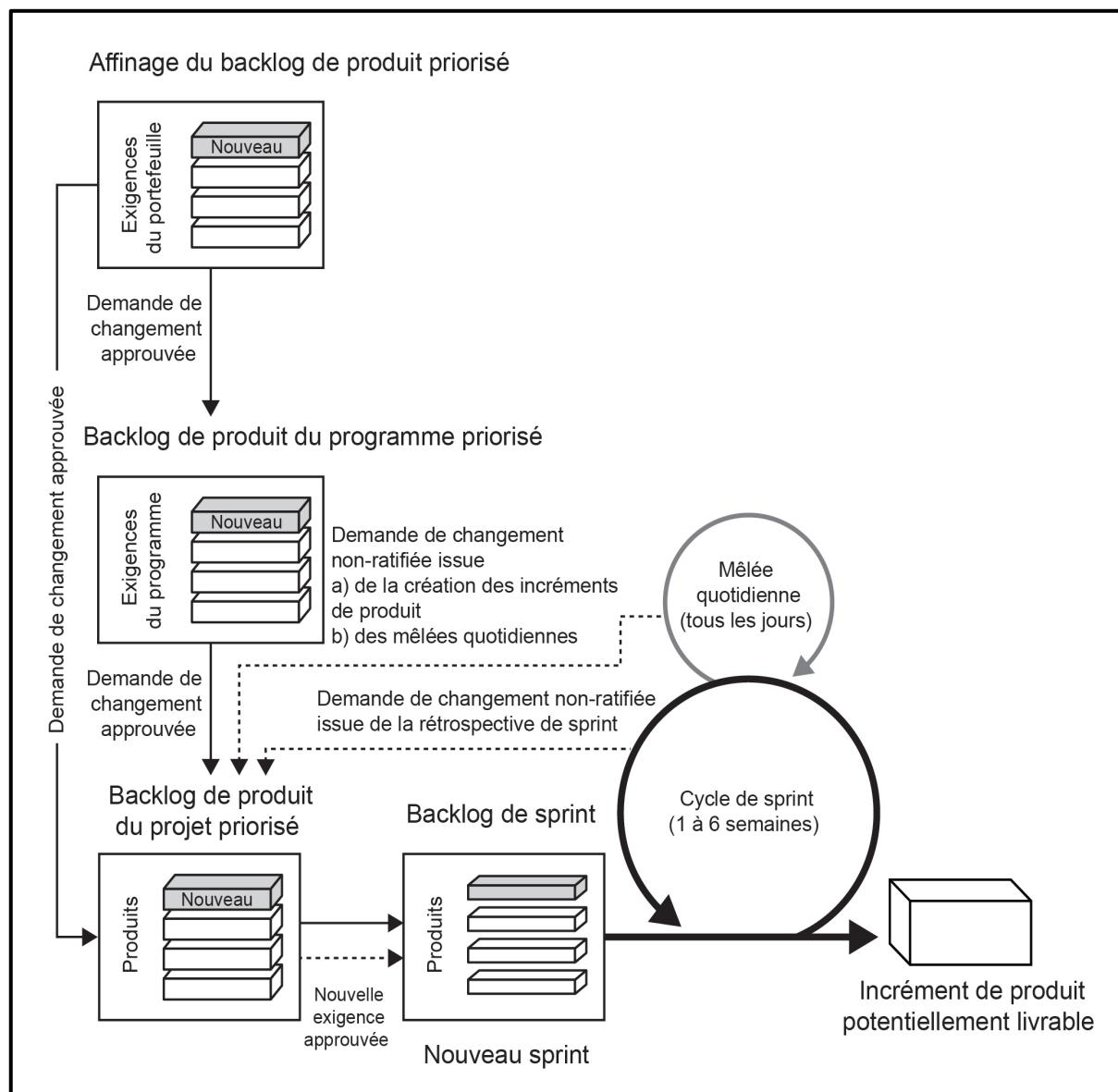


Figure 6-8 : Intégrer les changements dans les portefeuilles et les programmes

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 les parties prenantes 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	<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. 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 les parties prenantes 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 les parties prenantes peuvent ajouter des demandes de changement au backlog de produit priorisé. Le Product Owner et les parties prenantes 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 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 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.

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
- 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 élaborer 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

Les parties prenantes 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 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 les parties prenantes 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 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 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 face aux risques. Il définit le niveau ou la propension des parties prenantes 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

même 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 concernées et détermination de leur perception des évènements 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'*affinage du backlog de produit 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 les parties prenantes 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.

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 les parties prenantes 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).

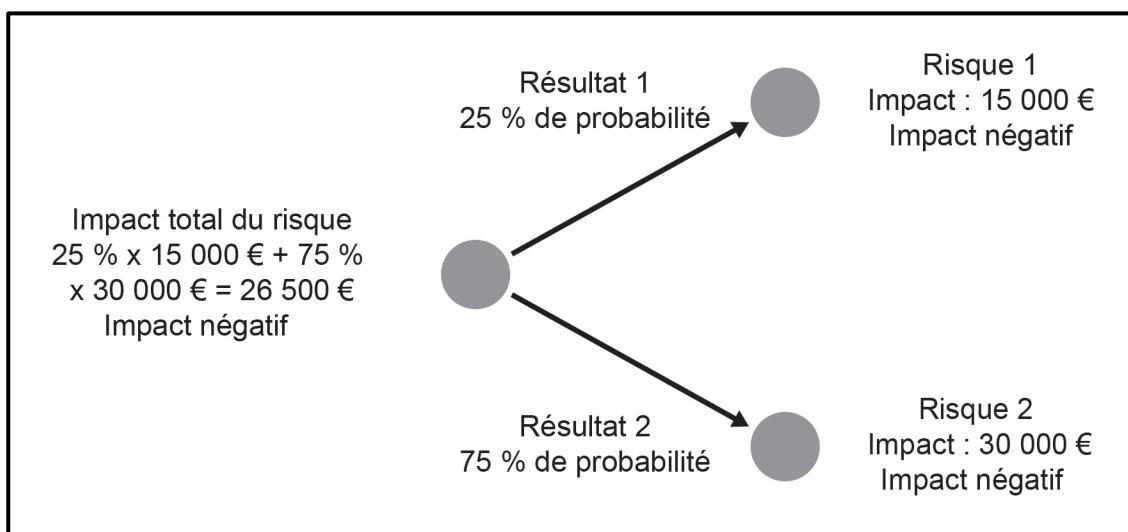


Figure 7-1 : Exemple d'arbre de probabilité

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 préféablement être traité en premier.

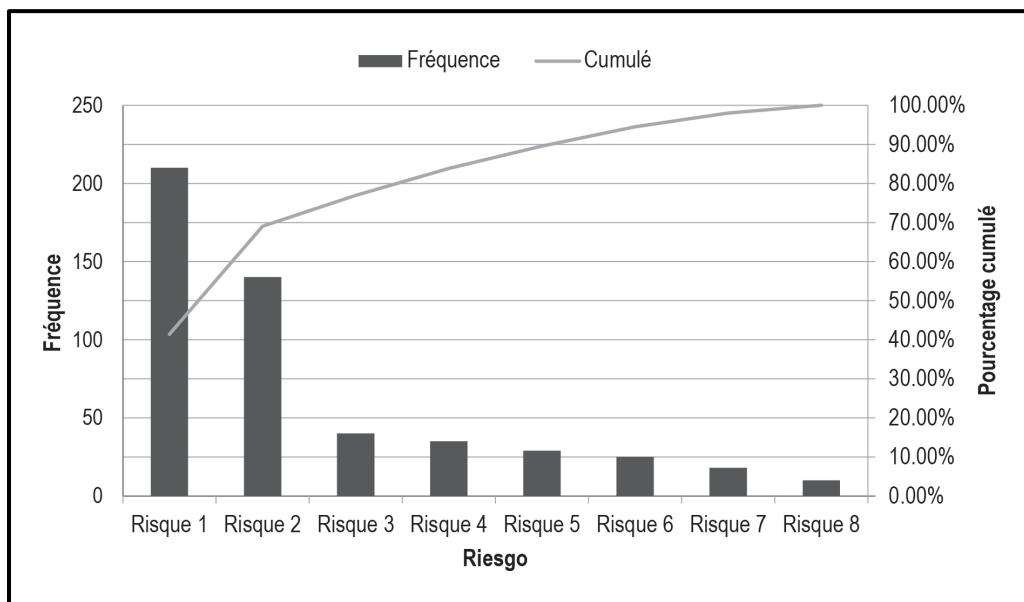


Figure 7-2 : Exemple de diagramme de Pareto

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 sévérité de risque pour un risque ayant une probabilité de 50% et un taux d'impact de 6 sera 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.

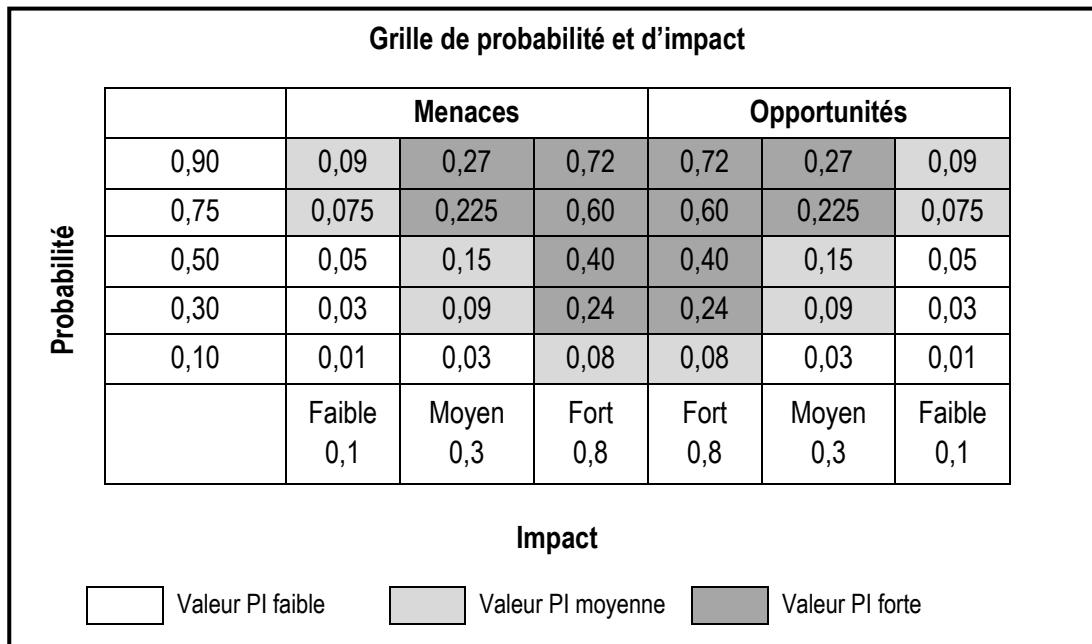


Figure 7-3 : Exemple de grille de probabilité et d'impact

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'*affinage du backlog de produit 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'*affinage du backlog de produit 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. créer une liste des user stories dans le backlog de produit, priorisées en fonction de leur valeur (p. ex. la valeur de chaque user story peut être évaluée en fonction du retour sur capital investi attendu)
4. fusionner les listes des étapes 2 et 3 et les prioriser en fonction de leur valeur afin d'obtenir un backlog de produit priorisé mis à jour

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

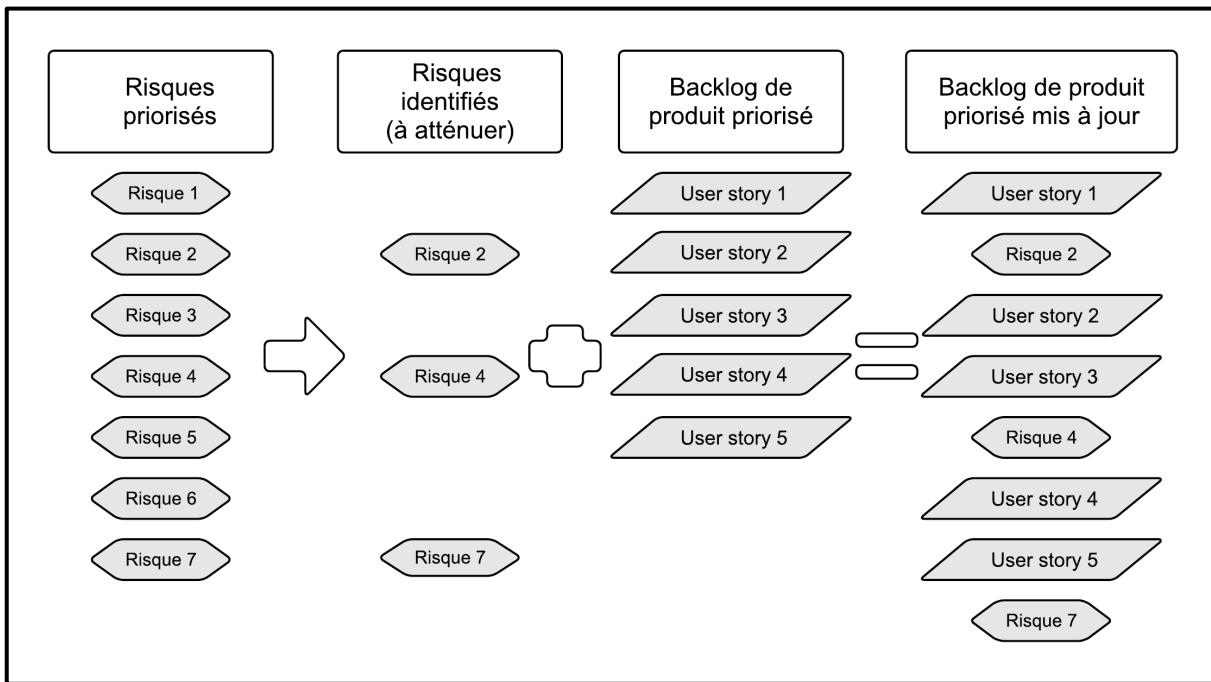


Figure 7-4 : Processus de priorisation des risques

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.

7.4.5 Communication des risques

Étant donné que les parties prenantes ont un intérêt pour le projet, il est important de communiquer avec elles au sujet des risques. Les informations fournies aux parties prenantes 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 restants 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

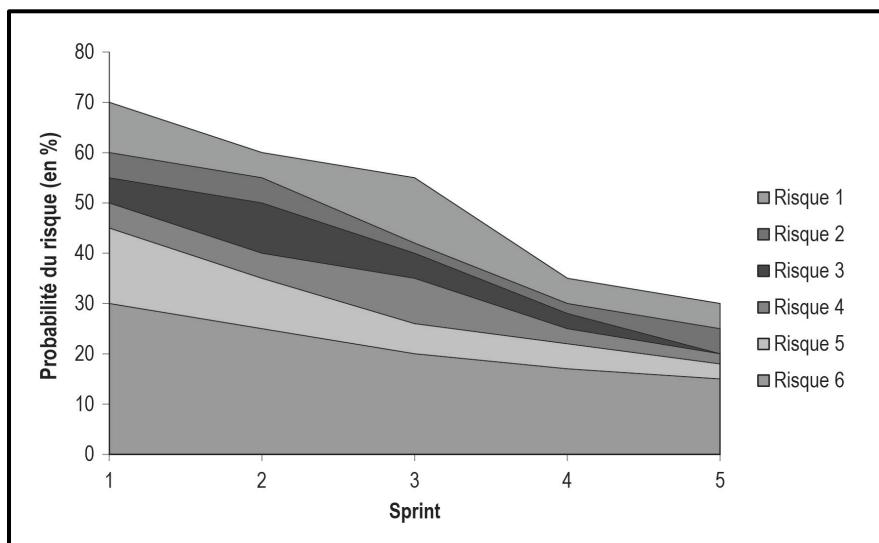


Figure 7-5 : Exemple de risk burndown chart

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 les parties prenantes 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 incrément 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, 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 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 illustre la manière dont les risques peuvent être gérés dans le flux Scrum pour les portefeuilles et les programmes.

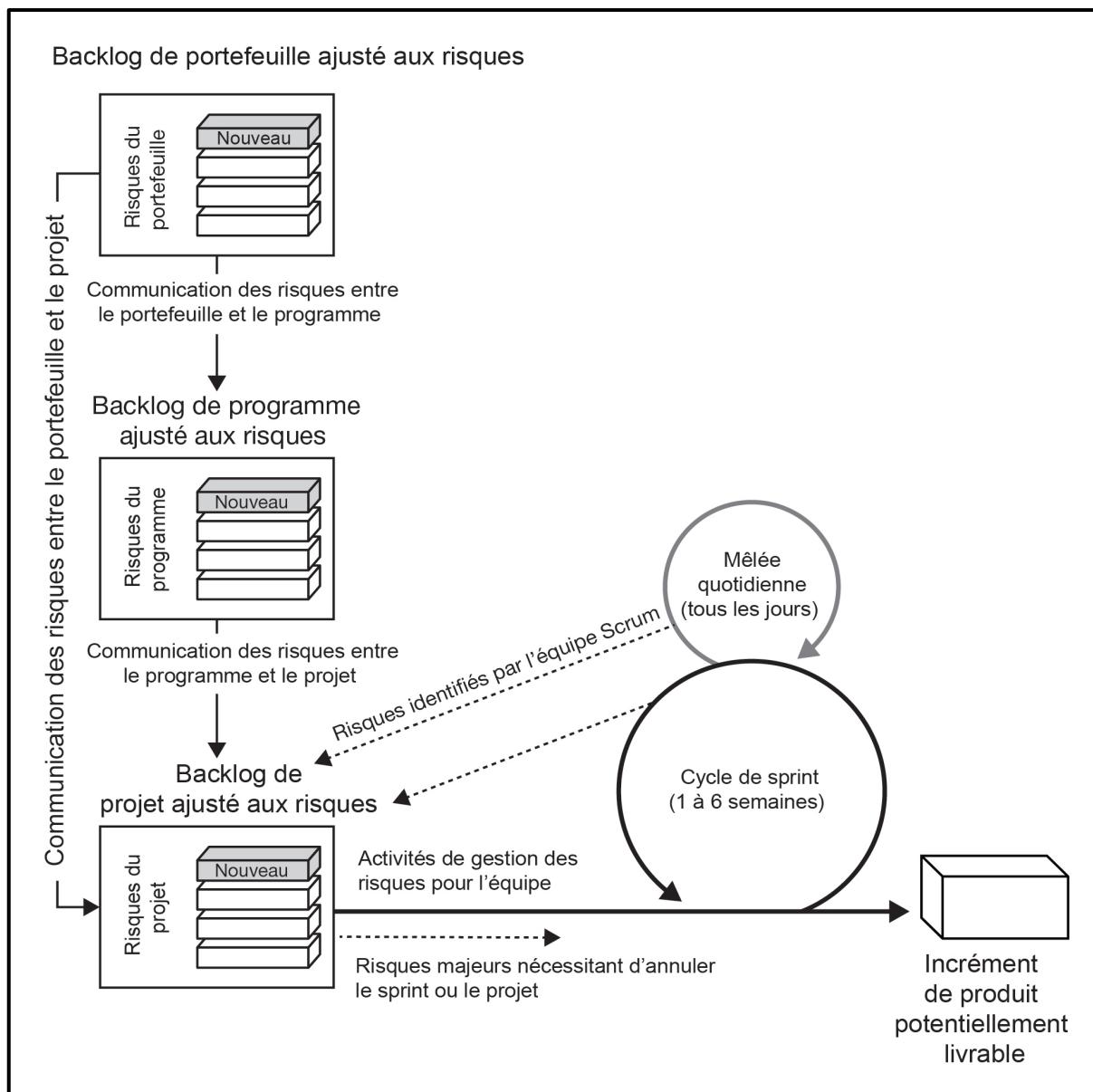


Figure 7-6 : Gestion des risques dans les portefeuilles et les programmes

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 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 et équipes du projet concernées
Scrum Master du programme	<ul style="list-style-type: none"> Facilite l'identification, l'évaluation et la remonté 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 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 remonté des risques pour les portefeuilles
Parties prenantes	<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 évènement(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, formation de l'équipe Scrum, développement des épics, 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
- les projets de toute taille et de tout niveau de complexité

8

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 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 — 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 centrer le projet tout entier. Le Product Owner est également identifié au cours de ce processus.

8.2 Identification du Scrum Master et des parties prenantes — le Scrum Master et les parties prenantes sont identifiés grâce à des critères de sélection spécifiques.

8.3 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.

8.4 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.

8.5 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.

8.6 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 les parties prenantes du projet. La durée des sprints est également décidée au cours de ce processus.

<h3>8.1 Création de la vision de projet</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Étude de cas du projet* 2. Product Owner de programme 3. Scrum Master du programme 4. Parties prenantes du programme 5. Backlog de produit du programme 6. Projet pilote 7. Démonstration de faisabilité 8. Vision de l'entreprise 9. Mission de l'entreprise 10. Étude de marché 11. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Réunion de vision du projet* 2. Sessions JAD 3. Analyse FFOM 4. Analyse des lacunes <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Product Owner identifié* 2. Énoncé de vision du projet* 3. Charte de projet 4. Budget du projet 	<h3>8.2 Identification du Scrum Master et des parties prenantes</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Product Owner* 2. Énoncé de vision du projet* 3. Product Owner de programme 4. Scrum Master de programme 5. Parties prenantes du programme 6. Besoins en personnel 7. Disponibilités et engagements du personnel 8. Matrice d'organisation des ressources 9. Matrice des compétences exigées 10. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Critères de sélection* 2. Conseils experts RH 3. Formation et coûts de formation 4. Coûts des ressources <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Scrum Master identifié* 2. Parties prenantes identifiées* 	<h3>8.3 Formation de l'équipe Scrum</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Product Owner* 2. Scrum Master* 3. Énoncé de vision du projet* 4. Besoins en personnel 5. Disponibilités et engagements du personnel 6. Matrice d'organisation des ressources 7. Matrice des compétences exigées 8. Ressources requises 9. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Sélection de l'équipe Scrum* 2. Conseils experts RH 3. Coûts du personnel 4. Formation et coûts de formation 5. Coûts des ressources <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Équipe Scrum identifiée* 2. Personnel remplaçant 3. Stratégie de collaboration 4. Programme de consolidation d'équipe
<h3>8.4 Développement des épics</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Équipe Scrum principale* 2. Énoncé de vision du projet* 3. Parties prenantes 4. Backlog de produit du programme 5. Demandes de changement approuvées 6. Demandes de changement non-ratifiées 7. Risques du programme et du portefeuille 8. Législations et règlements 9. Contrats applicables 10. Informations relatives aux projets précédents 11. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Réunions de groupe d'utilisateurs* 2. Atelier d'user stories 3. Réunions de groupe thématique 4. Entretiens d'utilisateurs ou de clients 5. Questionnaires 6. Techniques d'identification des risques 7. Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Epics* 2. Personnages* 3. Changements approuvés 4. Risques identifiés 	<h3>8.5 Crédit de backlog de produit priorisé</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Équipe Scrum principale* 2. Epics* 3. Personnages* 4. Parties prenantes 5. Énoncé de vision du projet 6. Backlog de produit du programme 7. Exigences commerciales 8. Demandes de changement approuvées 9. Risques identifiés 10. Contrats applicables 11. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Méthodes de priorisation des user stories* 2. Atelier d'user stories 3. Planification de la valeur 4. Techniques d'évaluation des risques 5. Estimation de la valeur du projet 6. Méthodes d'estimation des user stories 7. Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Backlog de produit priorisé* 2. Définition de terminé* 	<h3>8.6 Planification de la release</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Équipe Scrum principale* 2. Parties prenantes* 3. Énoncé de vision du projet* 4. Backlog de produit priorisé* 5. Définition de terminé* 6. Product Owner de programme 7. Scrum Master de programme 8. Backlog de produit du programme 9. Exigences commerciales 10. Calendrier de vacances 11. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Sessions de planification de la release* 2. Méthodes de priorisation de la release* <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Calendrier de planification de la release* 2. Durée du sprint* 3. Cibler les clients pour la release 4. Backlog de produit priorisé affiné

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.

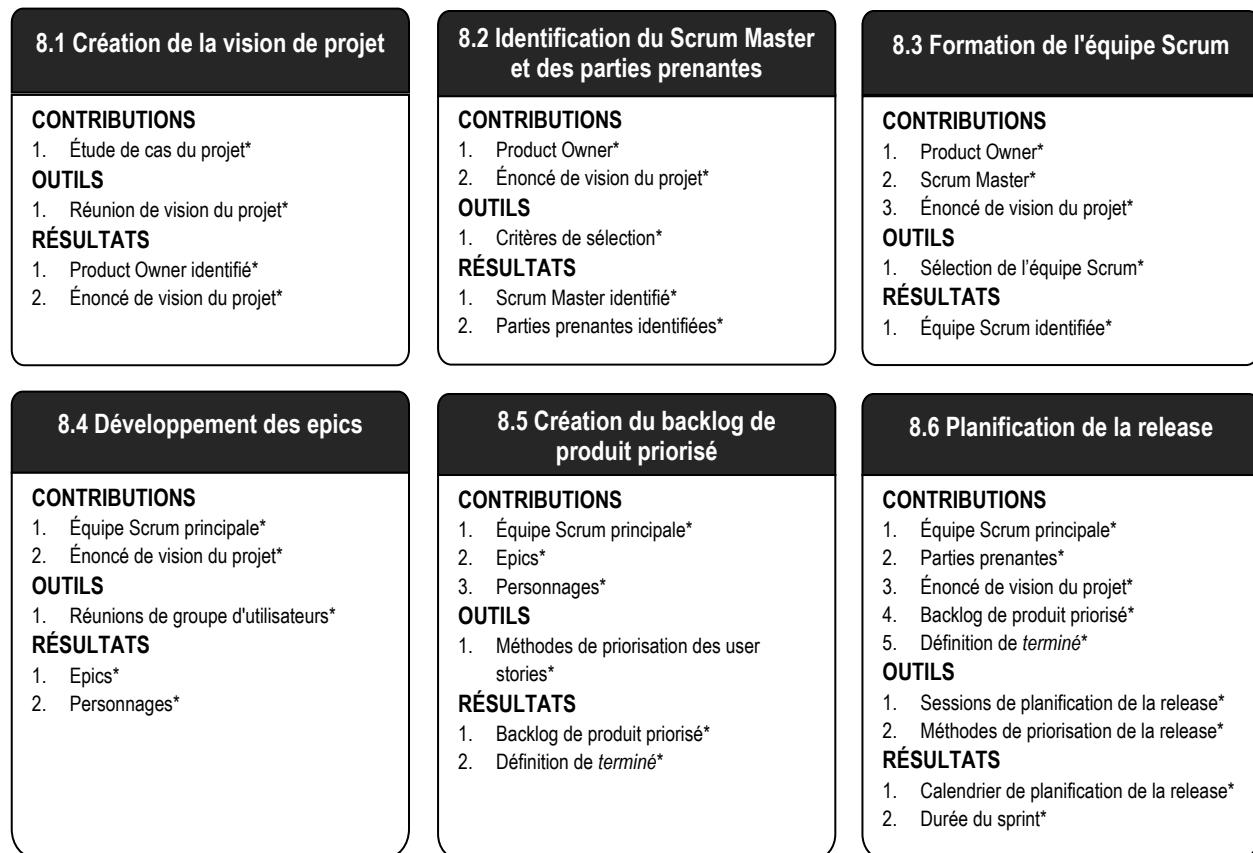


Figure 8-2 : Aperçu de la phase d'initiation (éléments essentiels)

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

8.1 Création de la vision de projet

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

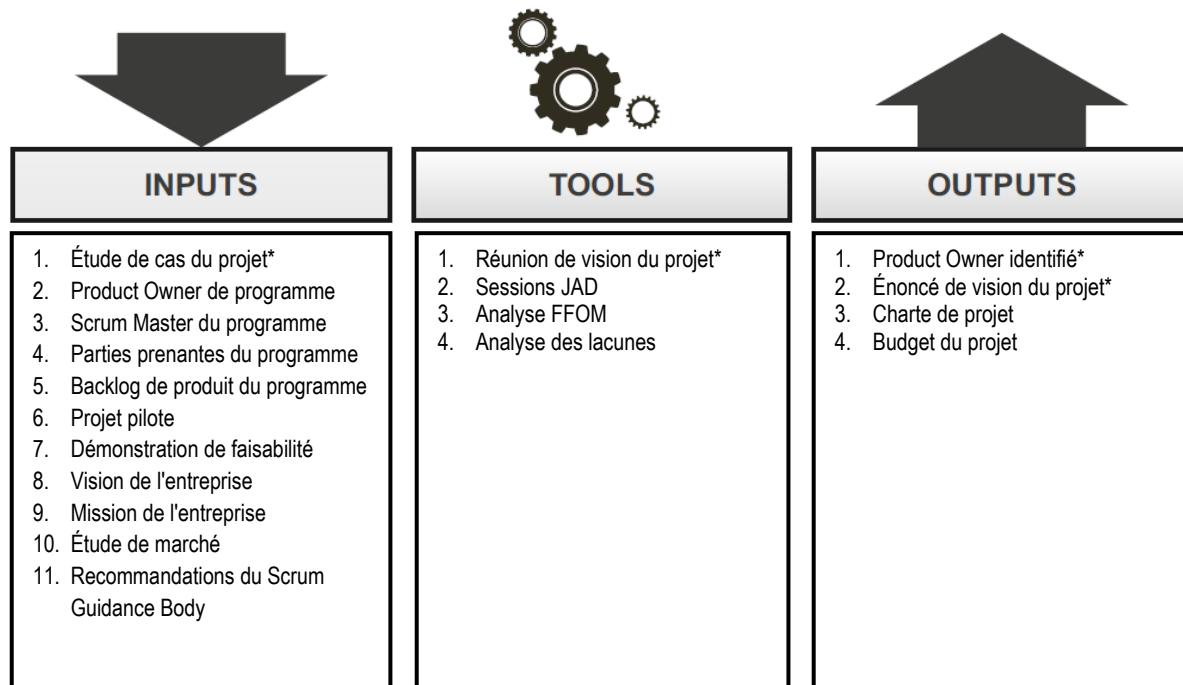


Figure 8-3 : Crédit de la vision de projet — Contributions, outils et résultats

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

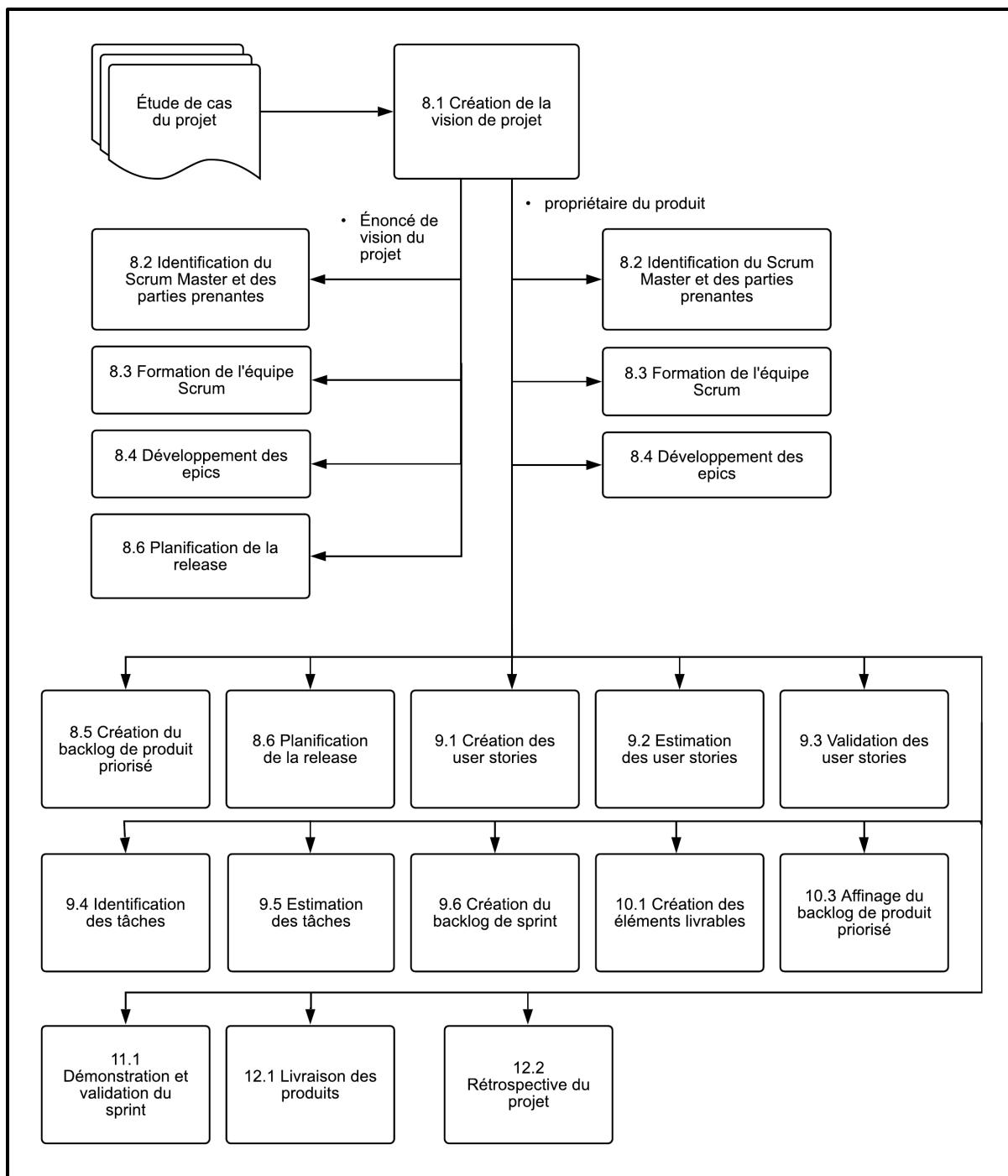


Figure 8-4 : Crédation de la vision de projet — diagramme de flux de données

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.

Le projet débute avec la présentation de l'étude de cas du projet. Une étude de cas est présentée aux parties prenantes et sponsors. Les parties prenantes comprennent les retombées commerciales attendues du projet et les sponsors confirment qu'ils fourniront les ressources financières pour le projet.

8.1.1.2 Product Owner du programme

Décrit dans la section 3.4.3.

8.1.1.3 Scrum Master du programme

Décrit dans la section 3.5.2.

8.1.1.4 Parties prenantes du programme

C'est un terme collectif qui inclut les clients, utilisateurs et sponsors d'un programme. Ils influencent tous les projets d'un programme tout au long du développement des projets. Les parties prenantes du programme peuvent également aider à définir la vision du projet et offrent des conseils au sujet de la valeur commerciale.

Les parties prenantes du programme interagissent avec les parties prenantes du portefeuille afin d'assurer l'adéquation entre le programme et les buts et objectifs du portefeuille. Ils participent également à la nomination des parties prenantes pour les projets individuels et assurent que la vision, les objectifs, résultats et releases des projets individuels au sein du programme sont conformes à ceux du programme.

8.1.1.5 Backlog de produit du programme

Le Product Owner du programme développe un backlog de produit de programme contenant une liste priorisée des exigences commerciales de haut niveau ainsi que des exigences du projet, écrit de préférence sous la forme de gros éléments dans backlog du programme. Celles-ci seront ensuite affinées par les Product Owners pour leurs projets individuels lors de la création des backlogs de produit priorisés. Ces backlogs de produit priorisés contiennent des user stories beaucoup plus petites et détaillées qui peuvent être estimées et validées par chaque équipe Scrum.

Le backlog de produit du programme est continuellement affiné par le Product Owner du programme afin de garantir que les nouvelles exigences commerciales sont ajoutées et que les exigences existantes sont correctement documentées et priorisées. Cela garantit la priorisation à haut niveau des exigences ayant le plus de valeur pour répondre aux objectifs du programme tandis que les autres exigences se voient attribuer un niveau de priorité plus faible.

Le backlog de produit créé pour le programme donne un aperçu plus large de tous les projets faisant partie du programme. Il peut donc fournir des indications importantes quant aux buts, à la portée, aux objectifs et aux retombées commerciales attendues pour les projets.

8.1.1.6 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.

8.1.1.7 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.

8.1.1.8 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.

8.1.1.9 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.

8.1.1.10 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.

8.1.1.11 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, 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 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 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 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 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. 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 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.

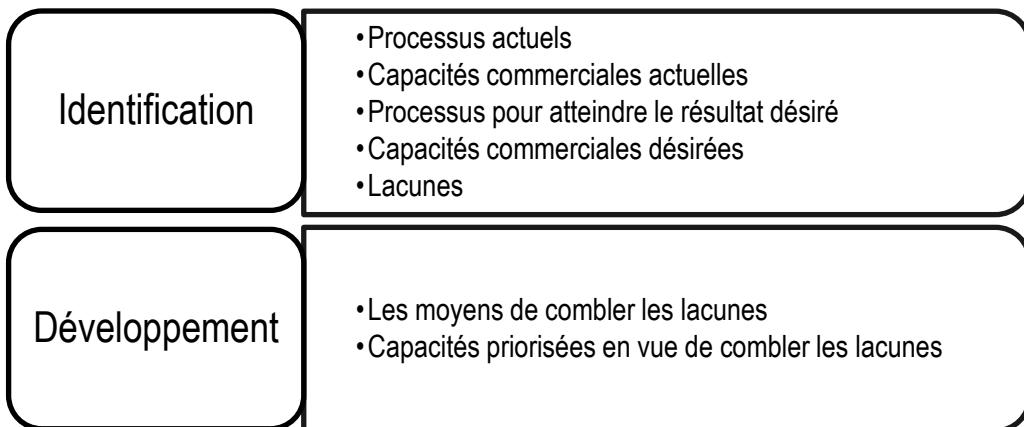


Figure 8-5 : Processus de l'analyse des lacunes

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.

8.2 Identification du Scrum Master et des parties prenantes

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

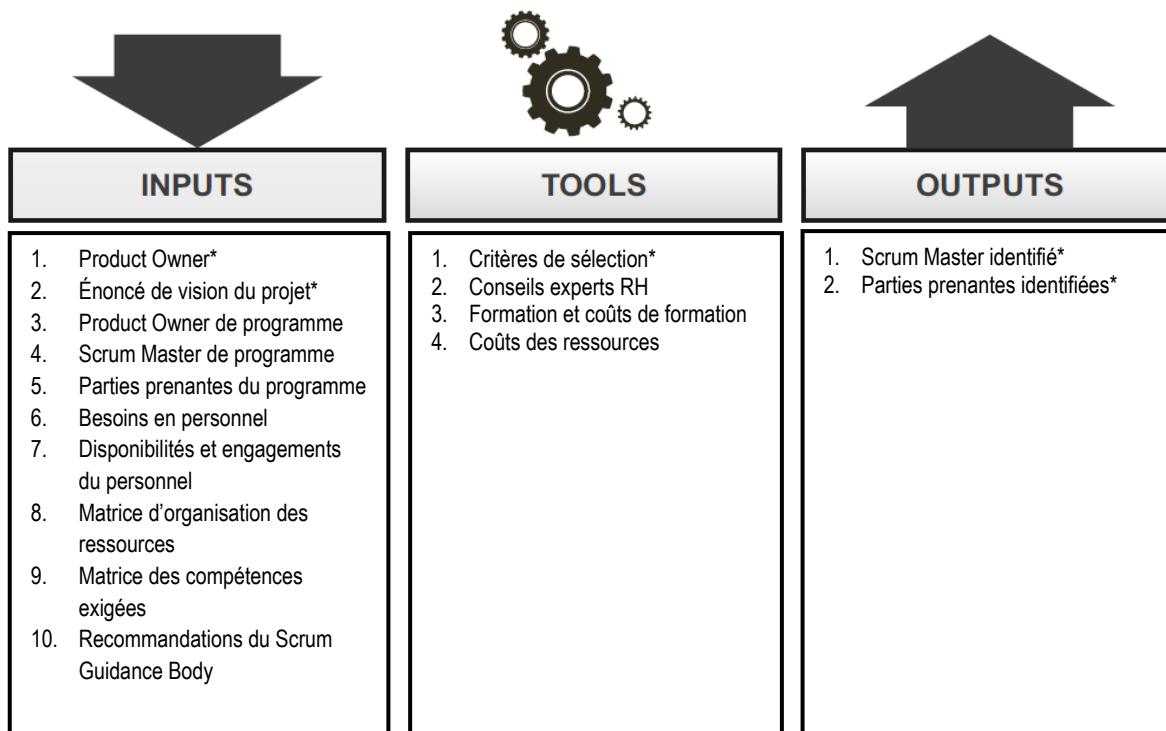


Figure 8-6 : Identification du Scrum Master et des parties prenantes — Contributions, outils et résultats

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

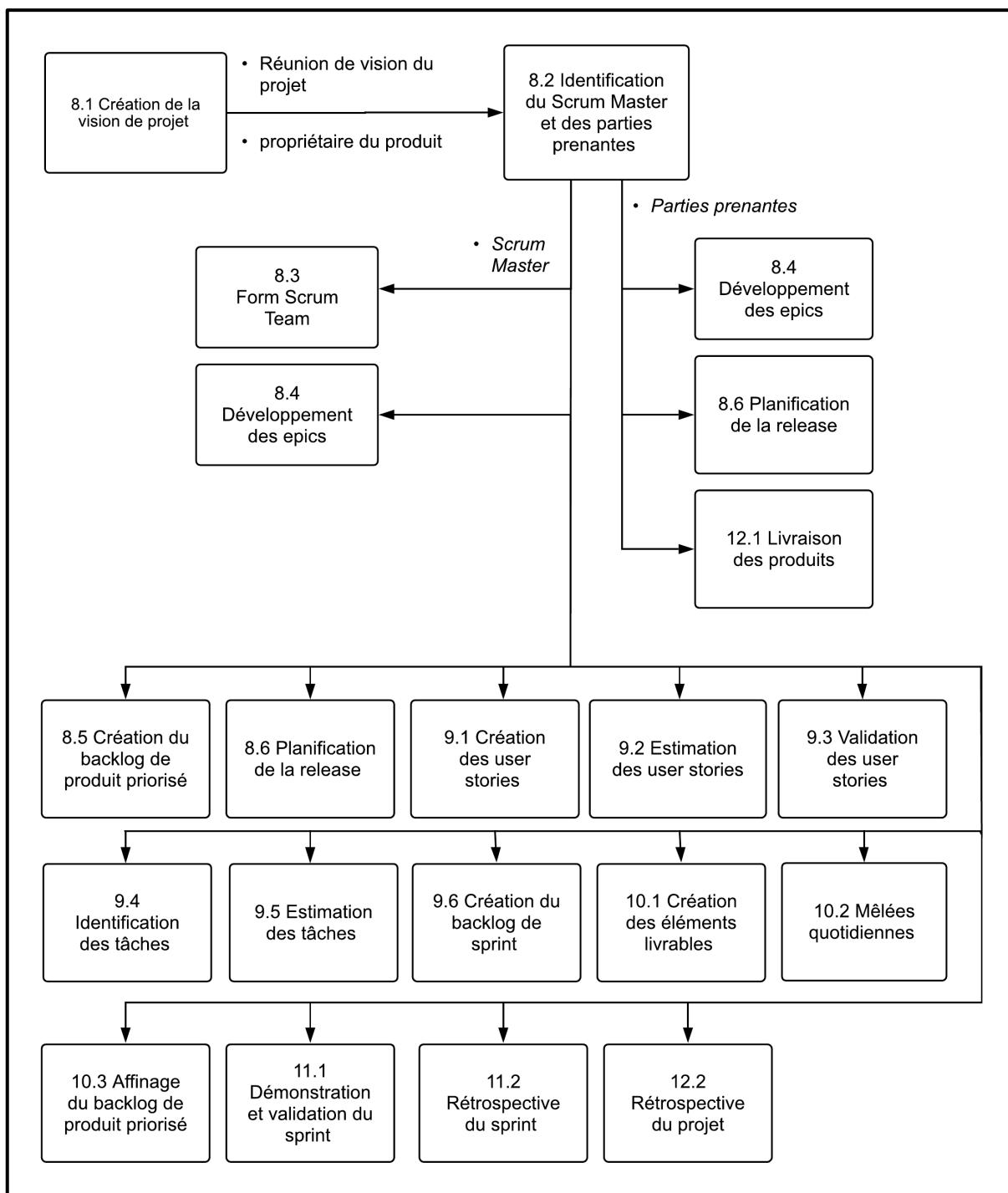


Figure 8-7 : Identification du Scrum Master et des parties prenantes — Diagramme de flux de données

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 Product Owner de programme

Décrit dans la section 3.4.3.

8.2.1.4 Scrum Master de programme

Décrit dans la section 3.5.2.

8.2.1.5 Parties prenantes du programme

Décrivées dans la section 8.1.1.4.

8.2.1.6 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. Il est important de documenter les rôles et responsabilités de tous ceux qui pourraient participer à la réalisation des tâches du projet. Cela inclut toutes les personnes impliquées dans le projet dans quelque capacité que ce soit, que leur rôle soit essentiel ou non.

Le Product Owner et le Scrum Master travaillent normalement avec le département des ressources humaines de l'entreprise afin de déterminer et de finaliser les besoins en termes de personnel pour un projet.

8.2.1.7 Disponibilités et engagements du personnel

Avant de sélectionner le Scrum Master et les parties prenantes, 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.

8.2.1.8 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.

8.2.1.9 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.

8.2.1.10 Recommandations du Scrum Guidance Body

Décrivées 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 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 les parties prenantes 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. Les parties prenantes 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. 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.

8.2.2.3 Formation et coûts de 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 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 identifiées*

Les parties prenantes (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 les parties prenantes.

Le rôle des parties prenantes est décrit dans la section 3.3.2.

8.3 Formation de l'équipe Scrum

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

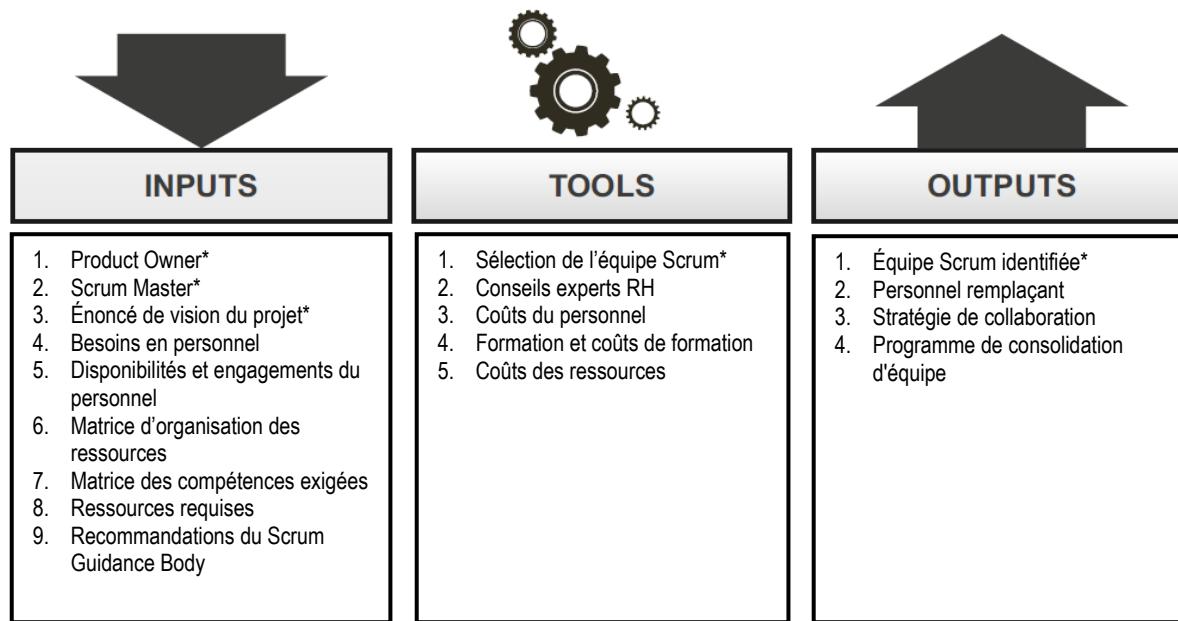


Figure 8-8 : Formation de l'équipe Scrum — Contributions, outils et résultats

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

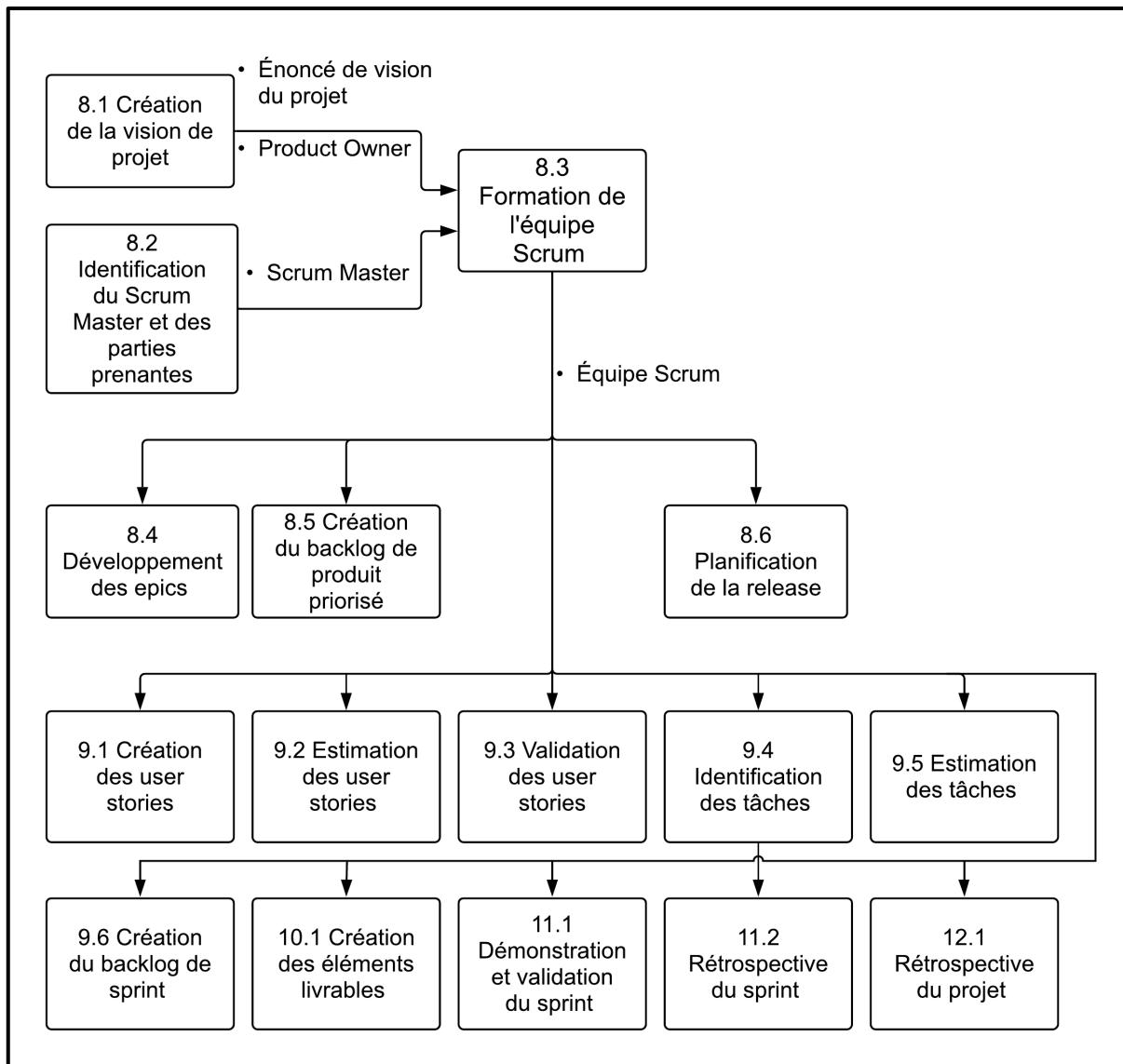


Figure 8-9 : Formation de l'équipe Scrum — Diagramme de flux de données

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**

Décris dans la section 8.2.1.8.

8.3.1.5 **Disponibilités et engagements du personnel**

Décris dans la section 8.2.1.9.

8.3.1.6 **Matrice d'organisation des ressources**

Décrise dans la section 8.2.1.10.

8.3.1.7 **Matrice des compétences exigées**

Décrise 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

Décrivées dans la section 8.1.1.11.

8.3.2 Outils

8.3.2.1 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 et coûts de 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* 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.

8.3.3 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 Personnel remplaçant

Lors de la sélection des équipes, il est également important d'identifier des remplaçants pour chaque membre de l'équipe Scrum. Même si la disponibilité et l'engagement des membres de l'équipe sont confirmés à l'avance, des problèmes peuvent survenir, tels que les maladies, urgences familiales ou un membre de l'équipe qui quitte l'organisation. Les équipes Scrum travaillent en petits groupes de six à dix personnes. Avoir des remplaçants garantit que la perte d'un membre de l'équipe n'occasionnera pas une baisse trop importante de la productivité.

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 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. 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, les parties prenantes 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 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 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

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

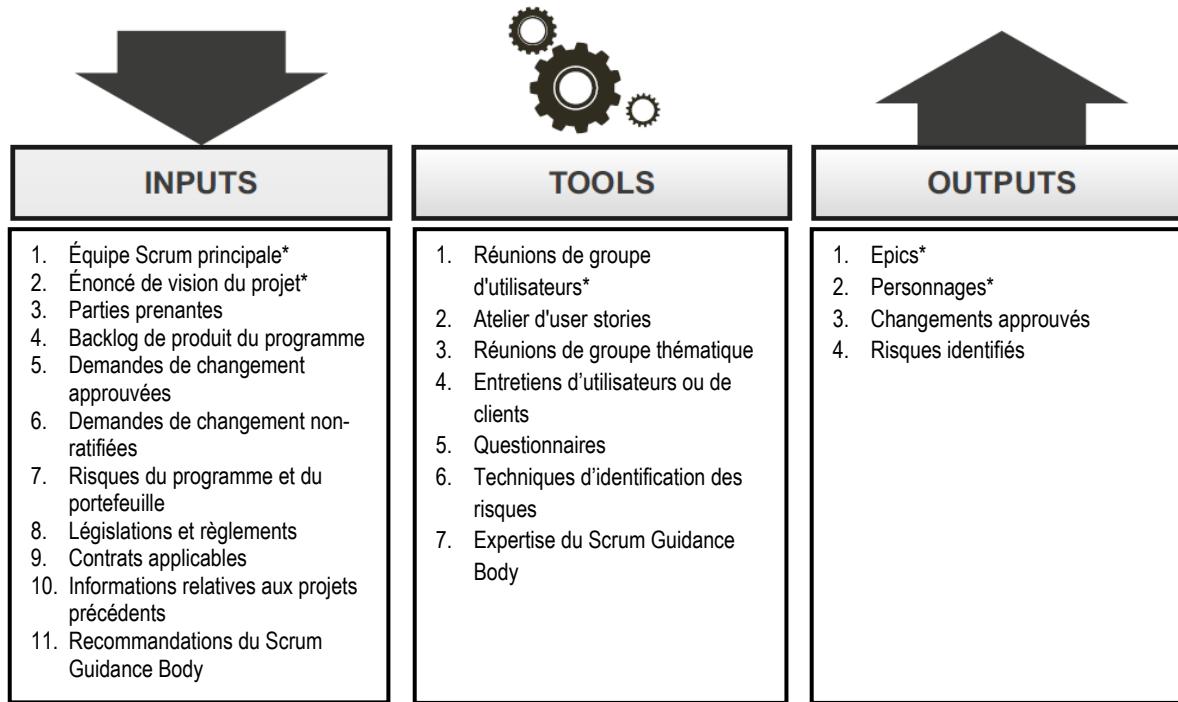


Figure 8-10 : Développement des epics — Contributions, outils et résultats

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

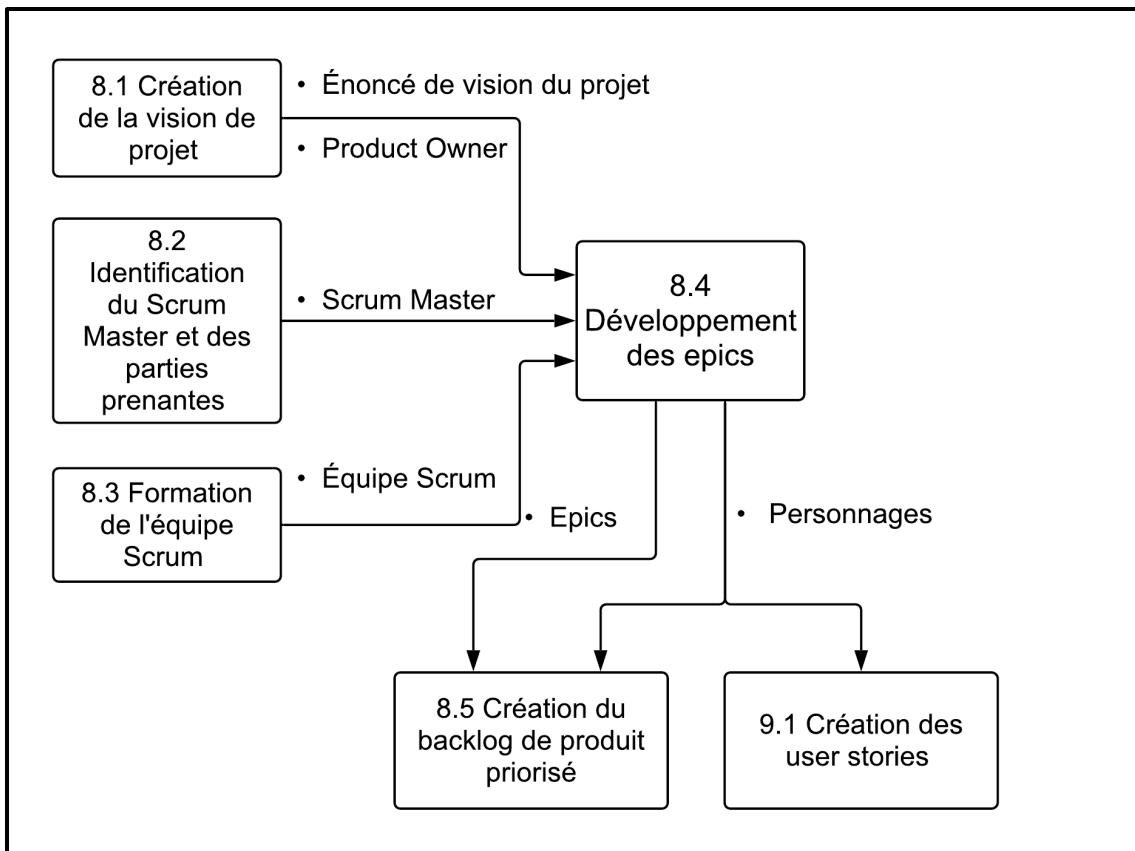


Figure 8-11 : Développement des épics — Diagramme de flux de données

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

Décrisées 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 les parties prenantes 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 protégé 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.

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

Décrivées 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 les parties prenantes concernées (principalement les utilisateurs ou clients du produit). Elles 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 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 les parties prenantes 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. 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 les parties prenantes (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 recueillir des opinions et des faits
- de comprendre la perspective des parties prenantes au sujet du produit fini
- de recueillir du feedback concernant les produits itératifs ou partiellement développés

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 ou de l'équipe Scrum.

8.4.2.6 Techniques d'identification des risques

Décrivées dans la section 7.4.1.1.

8.4.2.7 Expertise du Scrum Guidance Body

Décrivée 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.

8.4.3 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. Ce sont des user stories larges et brutes dans le backlog de produit priorisé.

Une fois que ces 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 et granulaires. 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.

8.4.3.2 Personnages*

Les personnages sont des personnages fictifs très détaillés, représentant la majorité des utilisateurs ainsi que les autres parties prenantes 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 les parties prenantes 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

La figure 8-12 illustre les contributions, outils et résultats du processus de *création du backlog de produit priorisé*.

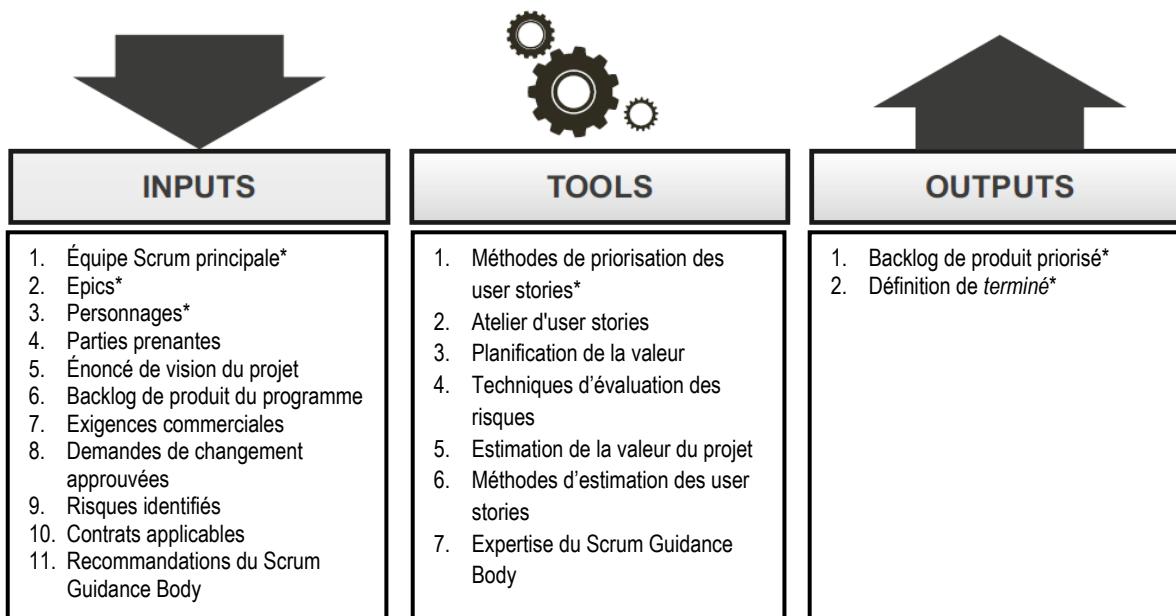


Figure 8-12 : Crédit du backlog de produit priorisé — Contributions, outils et résultats

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

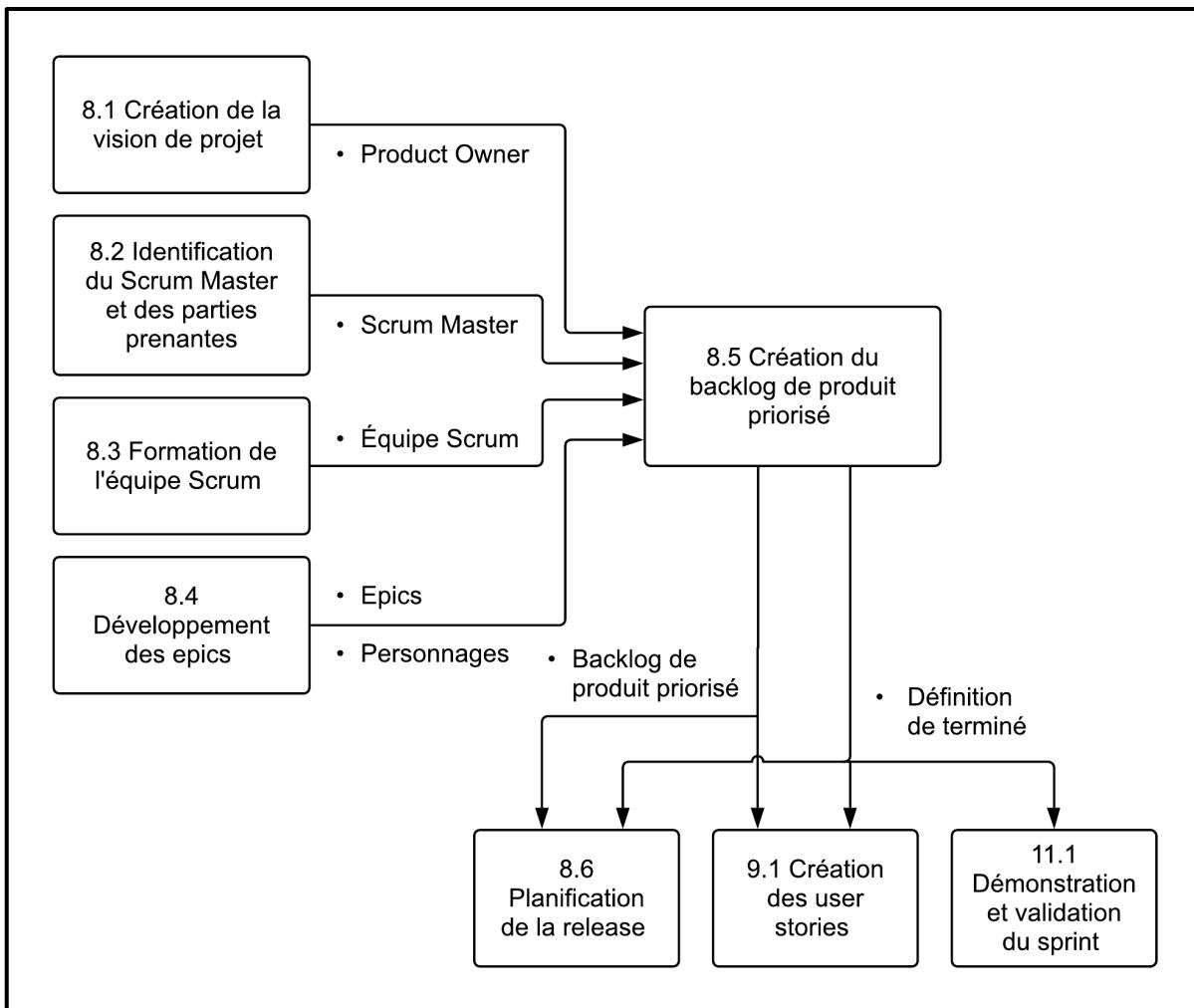


Figure 8-13 : Crédit du backlog de produit priorisé — Diagramme de flux de données

8.5.1 Contributions

8.5.1.1 Équipe Scrum principale*

Décrise dans la section 8.4.1.1.

8.5.1.2 Epics*

Décris 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

Décris 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

Décris 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

Décris dans la section 8.4.1.9.

8.5.1.11 Recommandations du Scrum Guidance Body

Décris 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 des user stories*

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**

Décrise 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

Décrise dans la section 4.5.2.

8.5.2.4 Techniques d'évaluation des risques

Décris dans la section 7.4.2.1.

8.5.2.5 Estimation de la valeur du projet

Décrise dans la section 4.5.1.

8.5.2.6 Méthodes d'estimation des user stories

Tous les outils utilisés afin d'estimer les user stories lors du processus d'estimation des user stories (tels qu'ils sont décrits dans la section 9.2.2.3) peuvent être utilisés afin de créer des estimations de haut niveau pour les épics lors de la création du backlog de produit priorisé. Quelques outils importants incluent notamment :

1. la méthode Wideband Delphi
2. le planning poker
3. le fist of five
4. l'estimation d'affinité

8.5.2.7 Expertise du Scrum Guidance Body

Décrive 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.

8.5.3 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** — il n'est normalement pas possible de créer un backlog de produit priorisé dans lequel il n'existe aucune dépendance entre les user stories. Les exigences fonctionnelles dépendent souvent d'autres exigences fonctionnelles ou même non fonctionnelles. Ces dépendances peuvent avoir un impact sur la façon dont les user stories sont priorisées dans le backlog de produit. Les deux méthodes les plus courantes permettant de résoudre les dépendances sont soit de décomposer une story unique en plusieurs parties ou de combiner les stories interdépendantes.
- **Estimations** — les estimations de haut niveau pour les épics 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.

8.6 Planification de la release

La figure 8-14 illustre les contributions, outils et résultats du processus de *planification de la release*.

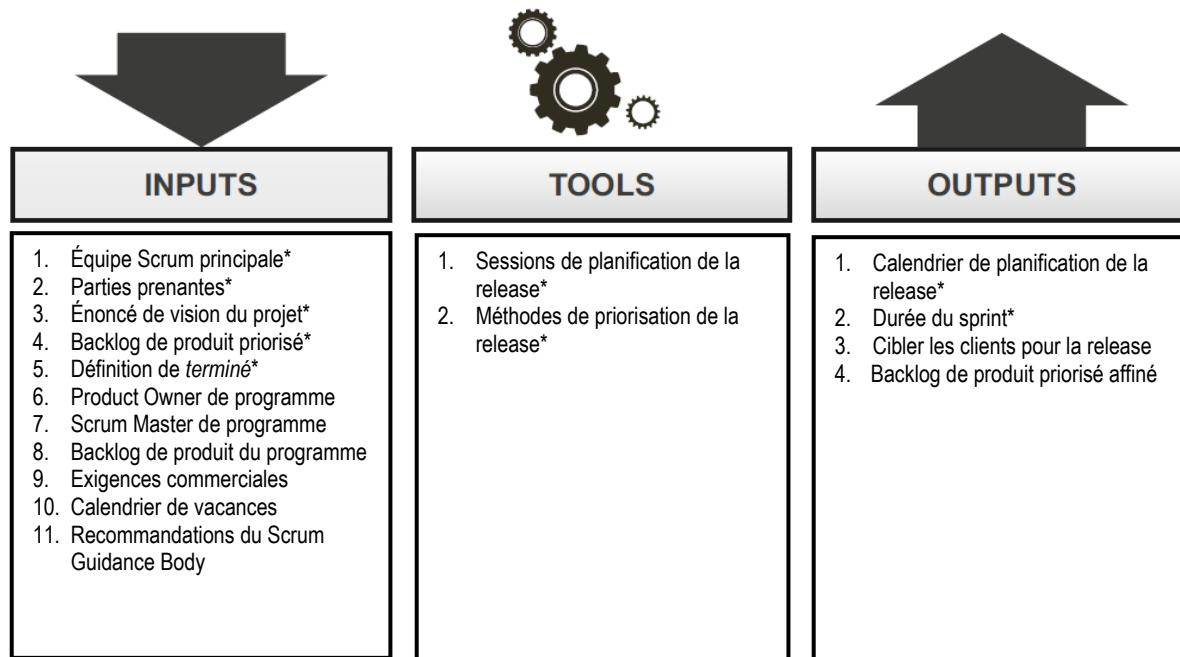


Figure 8-14 : Planification de la release — Contributions, outils et résultats

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

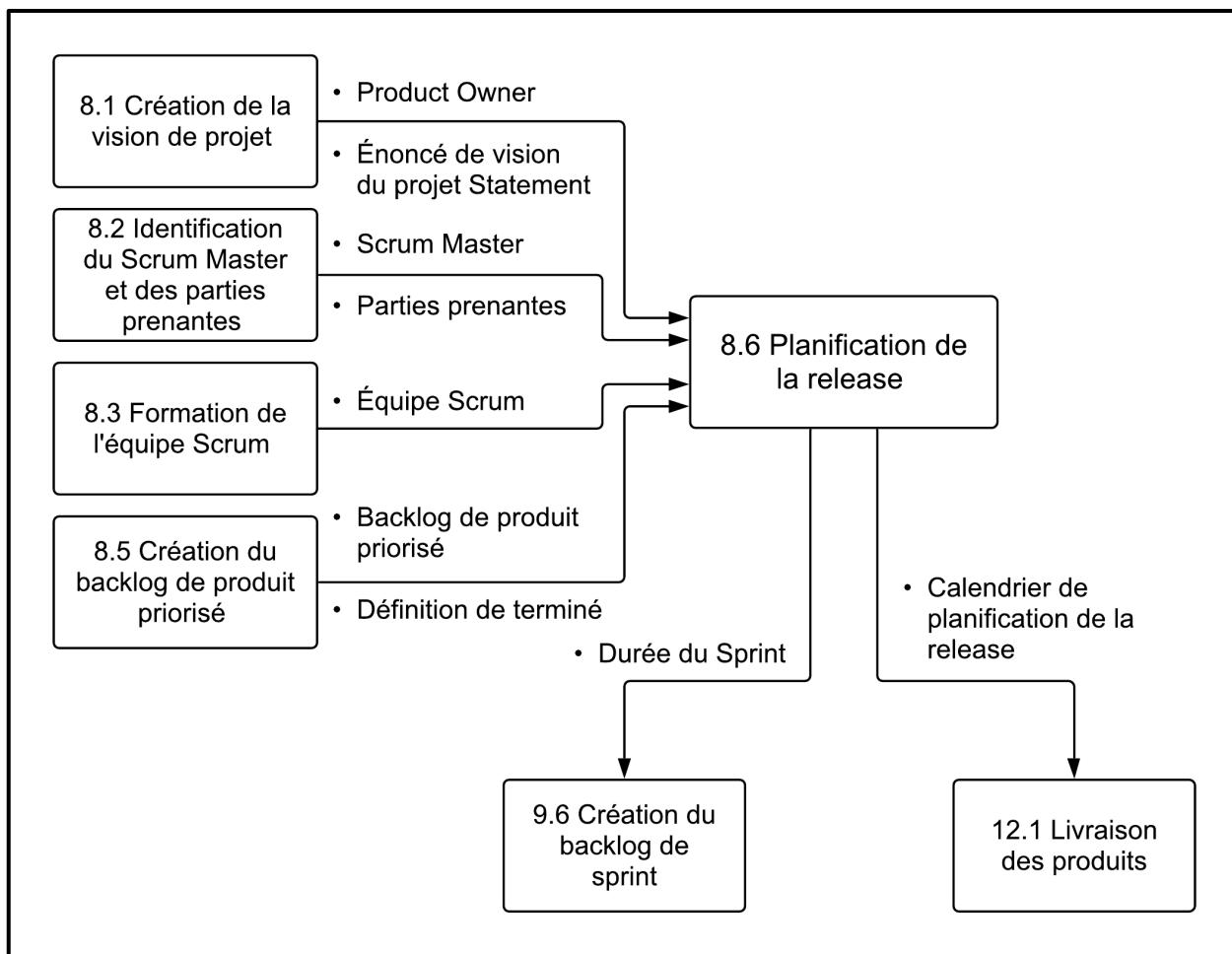


Figure 8-15 : Planification de la release — Diagramme de flux de données

8.6.1 Contributions

8.6.1.1 Équipe Scrum principale*

Décrise dans la section 8.4.1.1.

8.6.1.2 Parties prenantes*

Décrise 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*

Décrise dans la section 8.5.3.2.

8.6.1.6 Product Owner de programme

Décrit dans la section 3.4.3.

8.6.1.7 Scrum Master de programme

Décrit dans la section 3.5.2.

8.6.1.8 Backlog de produit du programme

Décrit dans la section 8.1.1.5.

8.6.1.9 Exigences commerciales

Décris dans la section 8.5.1.7.

8.6.1.10 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.

8.6.1.11 Recommandations du Scrum Guidance Body

Décrivées 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.

8.6.2 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 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.

8.6.3 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. 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. L'élément livrable doit être mis à disposition lorsqu'il offre suffisamment de valeur commerciale au client.

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.

La durée d'un sprint est normalement limitée entre 1 et 6 semaines. Cependant, 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.

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. Les parties prenantes 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 les parties prenantes.

8.7 Diagramme de flux de données de la phase d'initiation

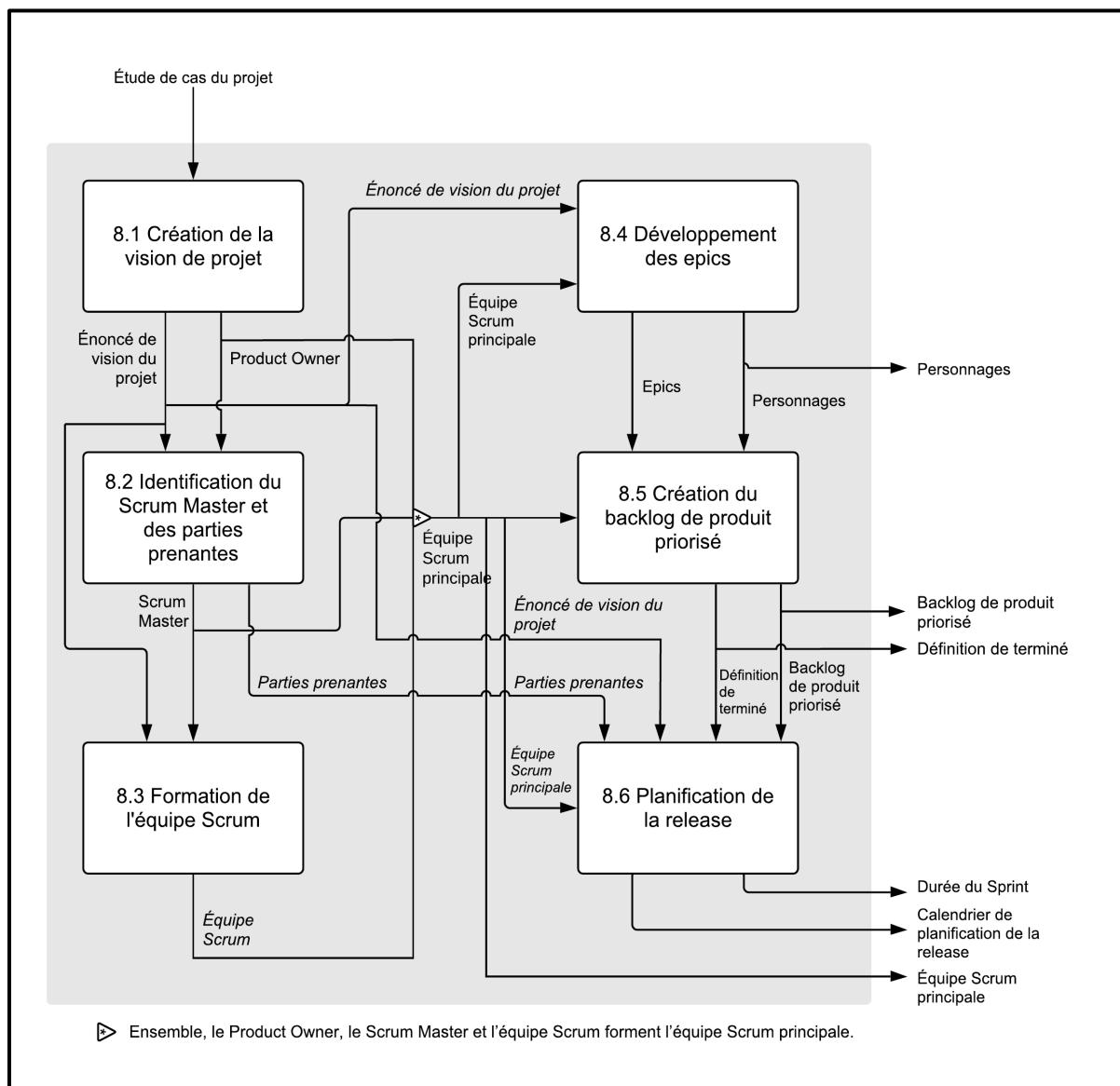


Figure 8-16 : Phase d'initiation — Diagramme de flux de données

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 *création du backlog de sprint*.

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
- les projets de toute taille et de tout niveau de complexité

9

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 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 — 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 les parties prenantes. Des ateliers 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.

9.2 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.

9.3 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.

9.4 Identification des tâches — les user stories validées sont divisées en tâches spécifiques et rassemblées dans une liste de tâches.

9.5 Estimation des tâches — l'équipe Scrum principale estime les efforts requis afin d'effectuer chaque tâche sur la liste des tâches lors d'un atelier d'estimation des tâches. Le résultat de ce processus est une liste des tâches et efforts estimés.

9.6 Création du backlog de sprint — 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 devant être accomplies pendant le sprint.

<h3>9.1 Création des user stories</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> Équipe Scrum principale* Backlog de produit priorisé* Définition de <i>terminé</i>* Personnages* Parties prenantes Epics Exigences commerciales Législations et règlements Contrats applicables Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> Expertise pour l'écriture des user stories* Atelier des user stories Réunions de groupe d'utilisateurs Réunions de groupe thématique Entretiens d'utilisateurs ou de clients Questionnaires Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ol style="list-style-type: none"> User stories* Critères d'acceptation des user stories* Backlog de produit priorisé mis à jour Personnages mis à jour ou affinés 	<h3>9.2 Estimation des user stories</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> Équipe Scrum principale* User stories* Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> Réunions de planification de sprint Réunions de revue du backlog de produit priorisé Méthodes d'estimation* <p>RÉSULTATS</p> <ol style="list-style-type: none"> Estimation des user stories* Backlog de produit priorisé mis à jour Critères d'acceptation des utilisateurs mis à jour 	<h3>9.3 Validation des user stories</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> Équipe Scrum principale* Estimation des user stories* Durée du sprint* Vélocité du sprint précédent Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> Réunions de planification de sprint* Techniques de communication <p>RÉSULTATS</p> <ol style="list-style-type: none"> User stories validées*
<h3>9.4 Identification des tâches</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> Équipe Scrum principale* User stories validées* <p>OUTILS</p> <ol style="list-style-type: none"> Réunions de planification de sprint* Décomposition Détermination des dépendances <p>RÉSULTATS</p> <ol style="list-style-type: none"> Liste des tâches* User stories validées mises à jour* Dépendances 	<h3>9.5 Estimation des tâches</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> Équipe Scrum principale* Liste des tâches* Critères d'acceptation des user stories Dépendances Risques identifiés Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> Réunions de planification de sprint* Critères d'estimation* Méthodes d'estimation* <p>RÉSULTATS</p> <ol style="list-style-type: none"> Liste des tâches et efforts estimés* Liste des tâches mise à jour 	<h3>9.6 Crédit du backlog de sprint</h3> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> Équipe Scrum principale* Liste des tâches et efforts estimés* Durée du sprint* Dépendances Calendrier de l'équipe <p>OUTILS</p> <ol style="list-style-type: none"> Réunions de planification de sprint* Outils de suivi de sprint Indicateurs de suivi de sprint <p>RÉSULTATS</p> <ol style="list-style-type: none"> Sprint backlog* Sprint burndown chart*

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.

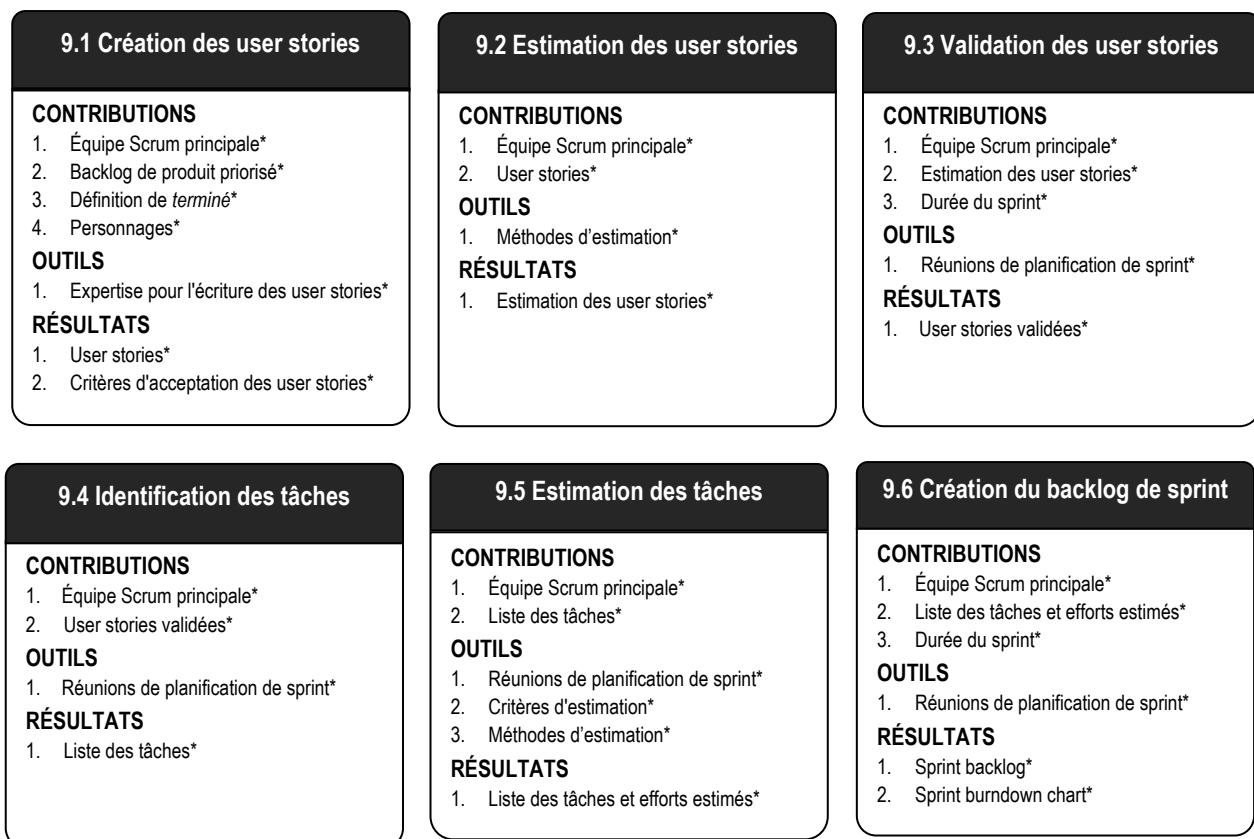


Figure 9-2 : Aperçu de la phase de planification et d'estimation (éléments essentiels)

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

9.1 Création des user stories

La figure 9-3 illustre les contributions, outils et résultats du processus de *création des user stories*.

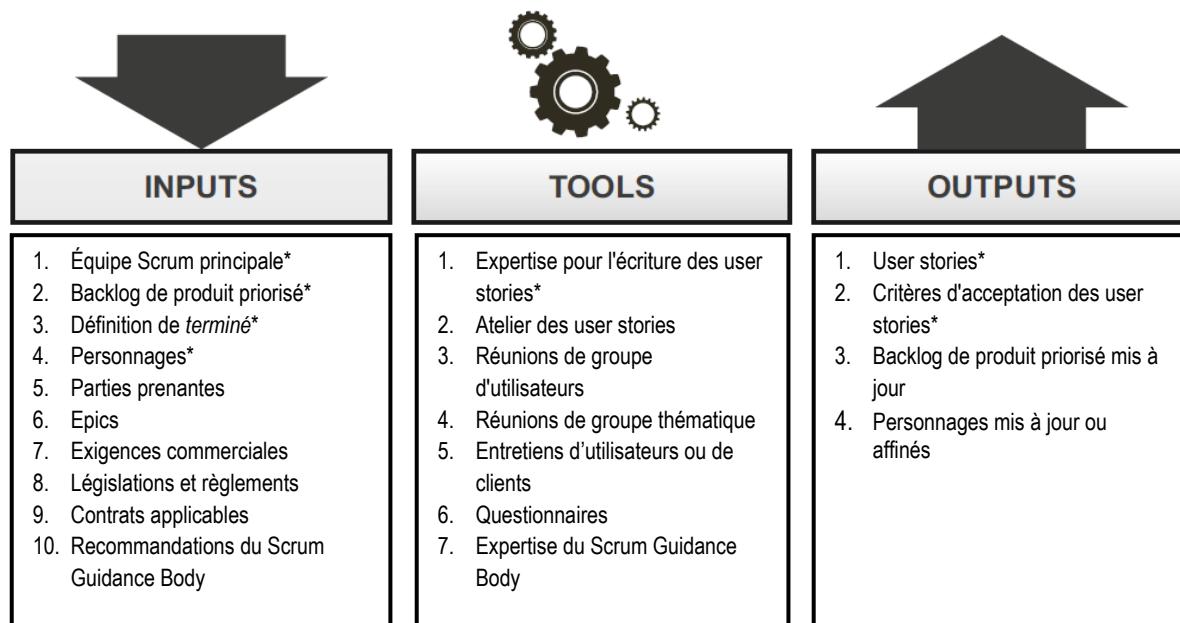


Figure 9-3 : Crédit des user stories — Contributions, outils et résultats

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

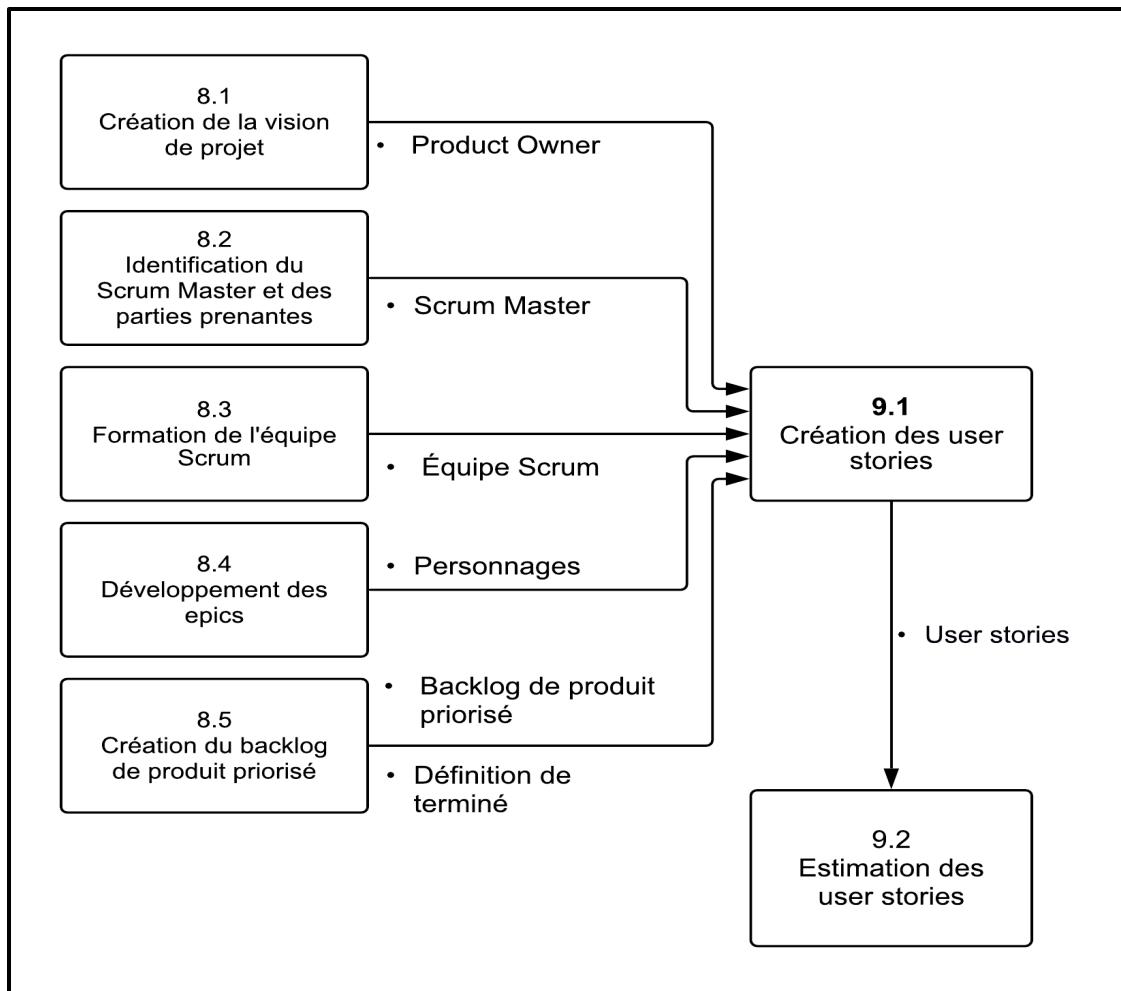


Figure 9-4 : Crédit des user stories — diagramme de flux de données

9

9.1.1 Contributions

9.1.1.1 Équipe Scrum principale*

Décrise 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*

Décrise dans la section 8.5.3.2.

9.1.1.4 Personnages*

Décris dans la section 8.4.3.2.

9.1.1.5 Parties prenantes

Décris dans la section 8.2.3.2.

9

9.1.1.6 Epics

Décris dans la section 8.4.3.1.

9.1.1.7 Exigences commerciales

Décris dans la section 8.5.1.7.

9.1.1.8 Législations et règlements

Décris dans la section 8.4.1.8.

9.1.1.9 Contrats applicables

Décris dans la section 8.4.1.9.

9.1.1.10 Recommandations du Scrum Guidance Body

Décrivées 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*

9

Le Product Owner, en fonction de ses interactions avec les parties prenantes, 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

Décrivées 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

9.1.2.7 Expertise du Scrum Guidance Body

Décrise 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.

9.1.3 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 prédéfinit permet une meilleure communication entre les parties prenantes 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

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

La figure 9-5 illustre les contributions, outils et résultats du processus d'estimation des user stories.

9



Figure 9-5 : Estimation des user stories — Contributions, outils et résultats

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

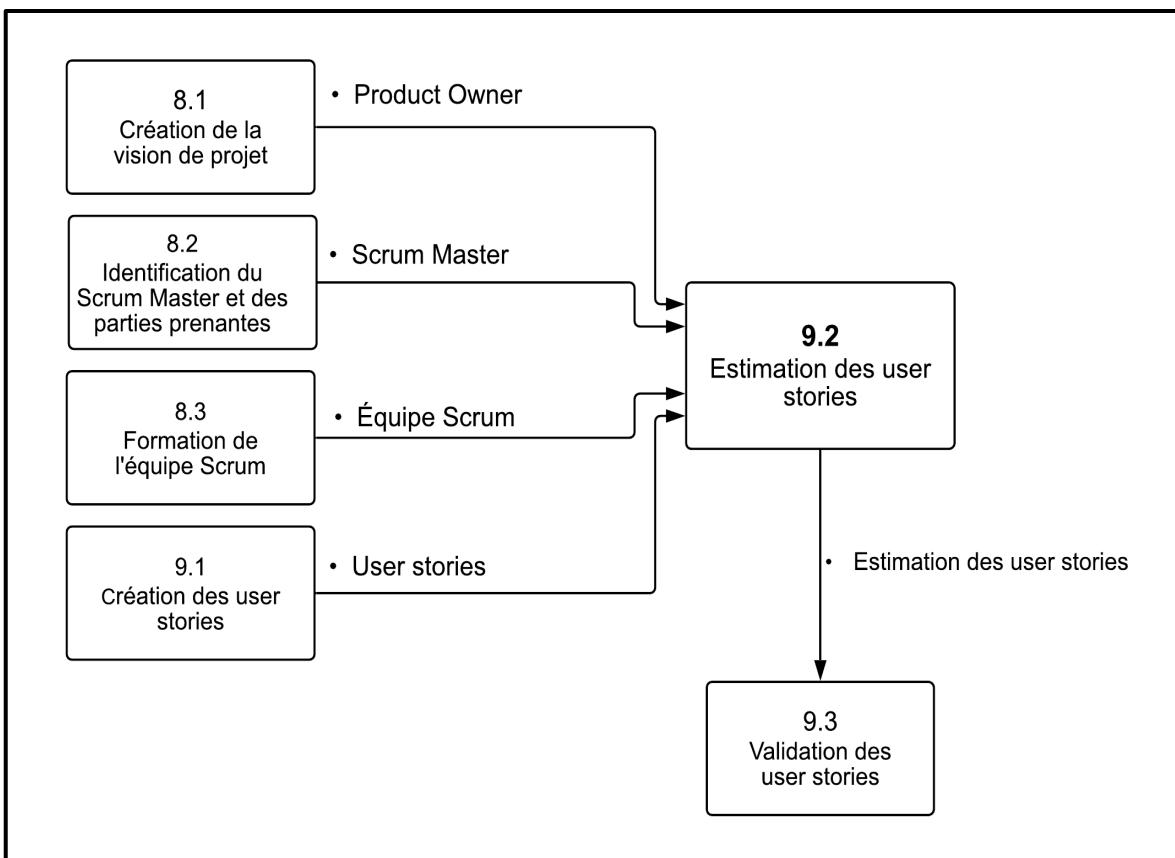


Figure 9-6 : Estimation des user stories — diagramme de flux de données

9.2.1 Contributions

9.2.1.1 Équipe Scrum principale*

Décrise dans la section 8.4.1.1.

9.2.1.2 User stories*

Décrise dans la section 9.1.3.1.

9.2.1.3 Recommandations du Scrum Guidance Body

Décrivées 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.

9.2.2 Outils

9.2.2.1 Réunions de planification de sprint

9

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.

9.2.2.2 Réunions de revue du backlog de produit priorisé

Les réunions de revue du backlog de produit sont organisées durant le processus d'affinage du backlog de produit priorisé. Au fur et à mesure de l'ajout ou de la mise à jour d'user stories affinées dans le backlog, l'équipe Scrum attribuera ou mettra à jour les estimations de haut niveau pour chaque user story. L'estimation en story points peut être utilisée afin d'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 est réalisée par l'équipe Scrum et une valeur en story point est attribuée. Une fois qu'une user story a été évaluée dans le backlog de produit priorisé, l'équipe Scrum peut évaluer d'autres user stories relatives à cette première story.

9.2.2.3 Méthodes d'estimation*

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.

9

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 de 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 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 valeur 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.

9.2.3 Résultats

9.2.3.1 User stories estimées*

Après que les user stories ont été estimées par l'équipe Scrum en utilisant les différentes techniques d'estimation abordées dans cette section, les user stories sont considérées comme étant des user stories estimées. L'estimation en story points peut être utilisée afin d'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 est réalisée par l'équipe Scrum et une valeur en story point est attribuée. Une fois qu'une user story a été évaluée dans le backlog de produit priorisé, l'équipe Scrum peut évaluer d'autres user stories relatives à cette première story. Il est important de noter que la calibration des story points pour chaque équipe sera différente, le nombre de story points réalisés ne peut pas être utilisé pour comparer deux équipes.

9.2.3.2 Backlog de produit priorisé mis à jour

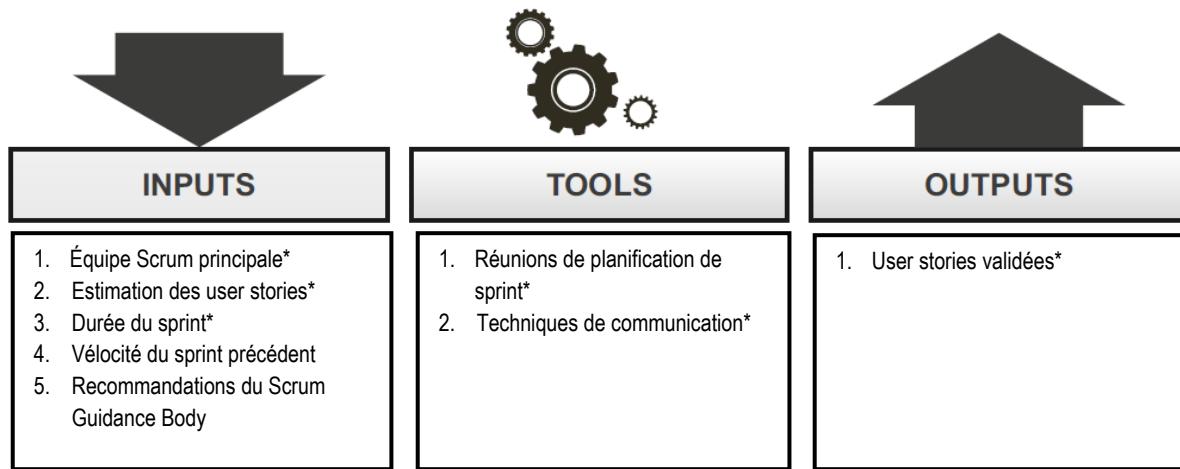
Décrit dans la section 9.1.3.3.

9.2.3.3 Critères d'acceptation des utilisateurs mis à jour

Décris dans la section 9.1.3.2.

9.3 Validation des user stories

La figure 9-7 illustre les contributions, outils et résultats du processus de *validation des user stories*.



9

Figure 9-7 : Validation des user stories — Contributions, outils et résultats

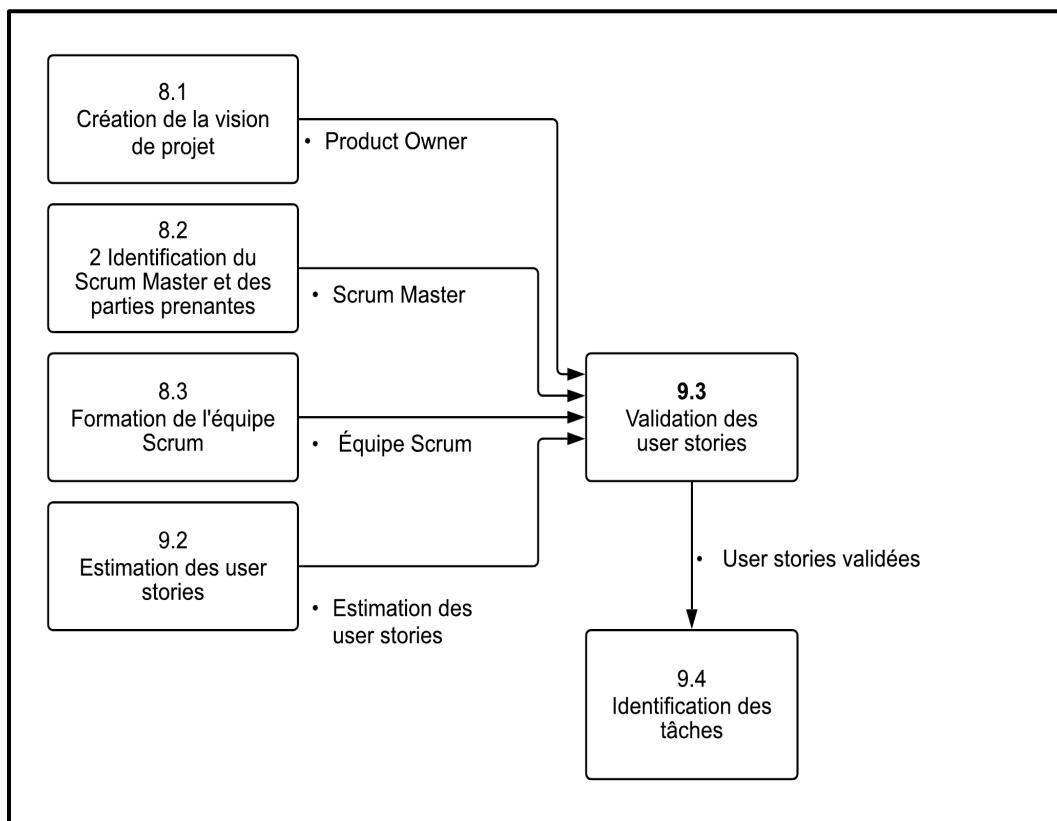


Figure 9-8 : Validation des user stories — diagramme de flux de données

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*

Décrise dans la section 8.4.1.1.

9.3.1.2 User stories estimées*

Décrises dans la section 9.2.3.1.

9.3.1.3 Durée du Sprint*

Décrise dans la section 8.6.3.2.

9.3.1.4 Vélocité du sprint précédent

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 celle utilisée pour les estimations, normalement en story points ou en durée idéale. Un registre de la vélocité de sprint de l'équipe pour chaque sprint est maintenu et utilisé comme référence lors des futurs sprints. La vélocité 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 vélocité de sprint pour le prochain sprint.

9.3.1.5 Recommandations du Scrum Guidance Body

Décrises 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.

9

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 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 l'équipe puisse s'autoorganiser, collaborer et travailler efficacement.

9.3.3 Résultats

9.3.3.1 User stories validées*

En fonction de sa vitesse, 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.

9.4 Identification des tâches

La figure 9-9 illustre les contributions, outils et résultats du processus d'*identification des tâches*.



9

Figure 9-9 : Identification des tâches — Contributions, outils et résultats

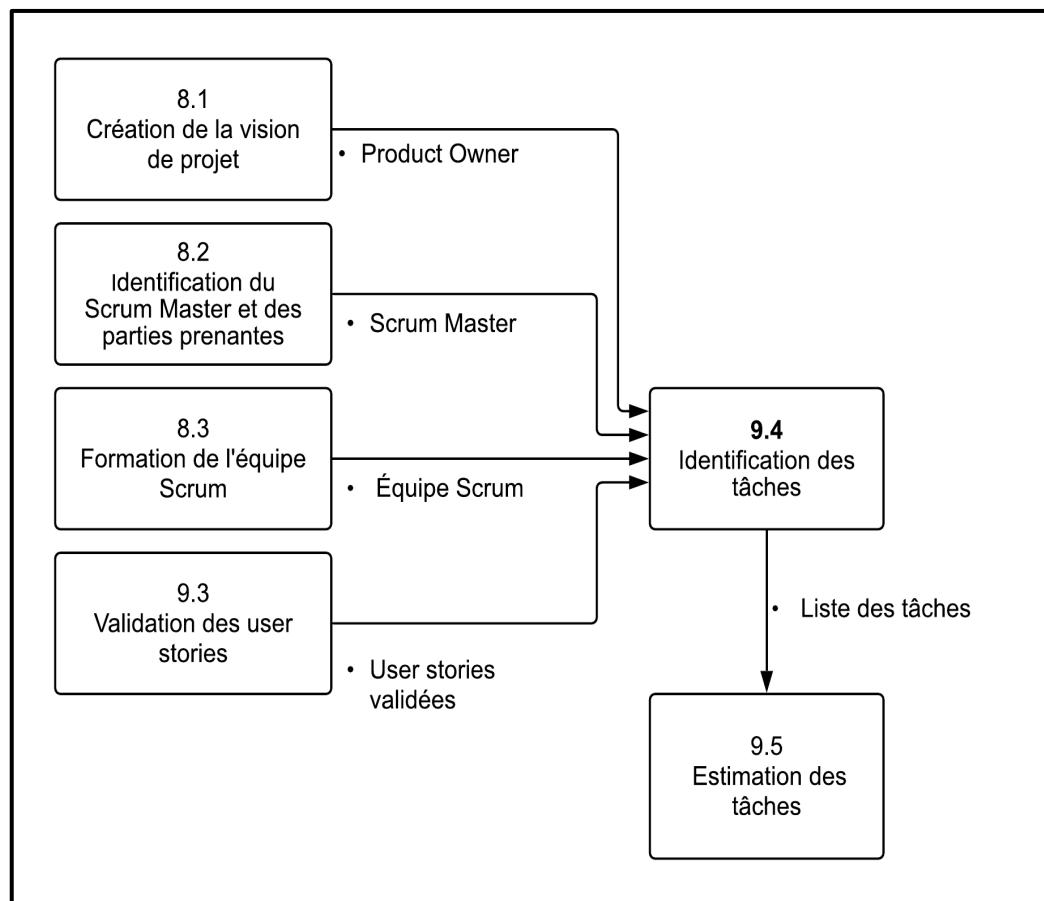


Figure 9-10 : Identification des tâches — Diagramme de flux de données

9.4.1 Contributions

9.4.1.1 Équipe Scrum principale*

Décrise dans la section 8.4.1.1.

9

9.4.1.2 User stories validées*

Décrises dans la section 9.3.3.1.

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. 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 afin de créer des incrément de produits à partir des tâches mentionnées dans la liste de tâches.

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 incrément 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.

9.4.3 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

9.4.3.2 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.

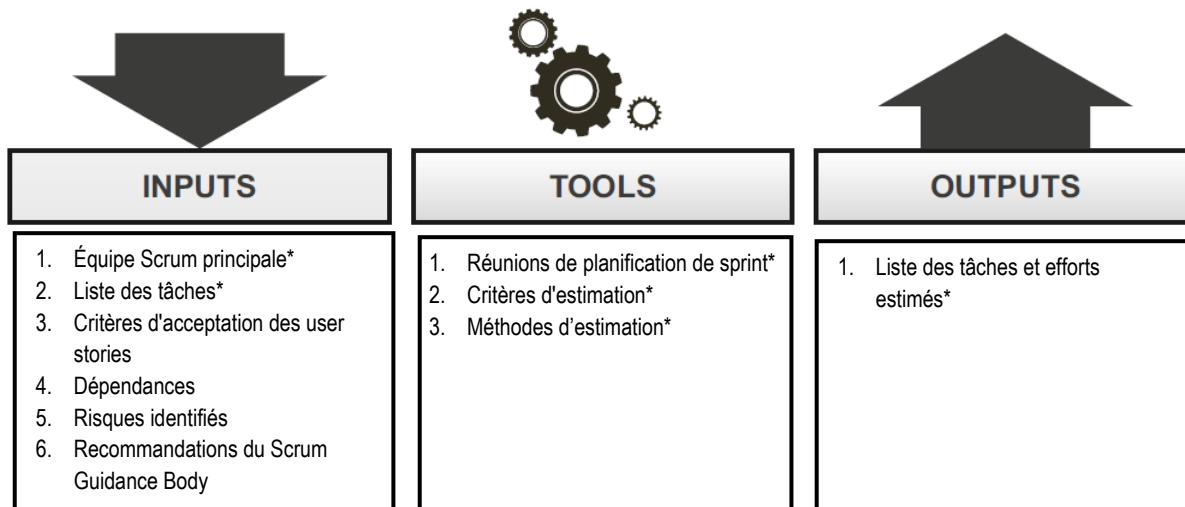
9.4.3.3 Dépendances

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 façons 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

La figure 9-11 illustre les contributions, outils et résultats du processus d'estimation des tâches.



9

Figure 9-11 : Estimation des tâches — Contributions, outils et résultats

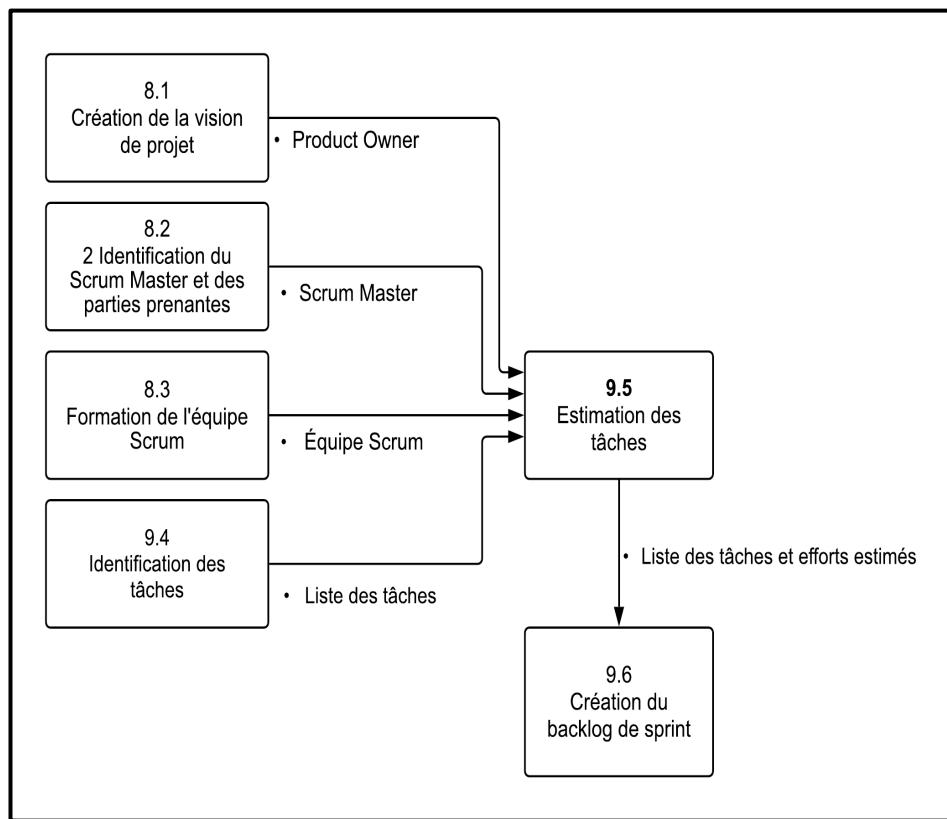


Figure 9-12 : Estimation des tâches — Diagramme de flux de données

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*

Décrise dans la section 8.4.1.1.

9.5.1.2 Liste des tâches*

Décrise dans la section 9.4.3.1.

9.5.1.3 Critères d'acceptation des user stories

Décris 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

Décrises dans la section 9.4.3.3.

9.5.1.5 Risques identifiés

Décris dans la section 8.4.3.4.

9.5.1.6 Recommandations du Scrum Guidance Body

Décrivées 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.

9.5.2 Outils

9.5.2.1 Réunions de planification de sprint

9

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*

Décrivées dans la section 9.2.2.3.

Les méthodes d'estimation utilisées pour estimer les user stories peuvent également être appliquées aux tâches.

9.5.3 Résultats

9.5.3.1 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. La précision des estimation varie en général en fonction des compétences de l'équipe. 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. Elle est également utilisée pour déterminer les moments où l'équipe doit réduire ses engagements ou peut prendre plus d'user stories lors de la planification de sprint.

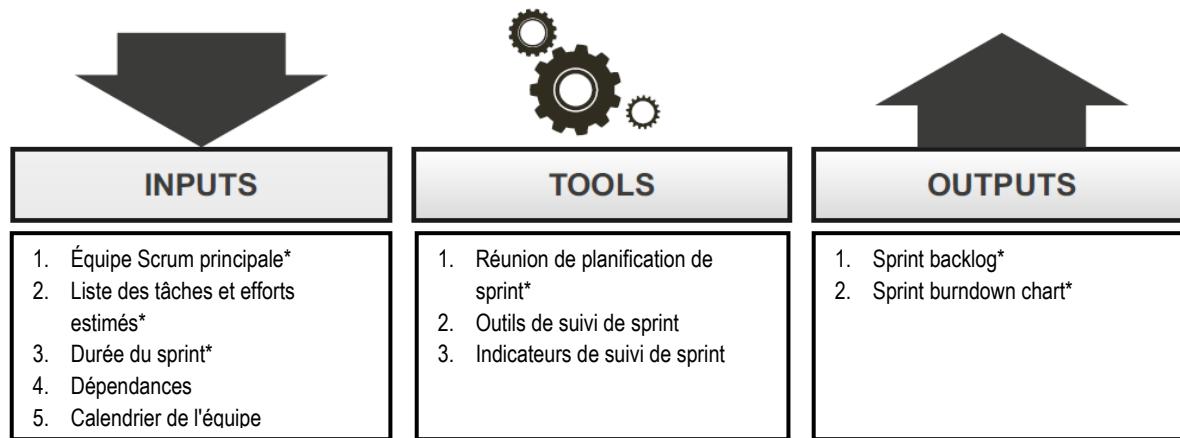
9

9.5.3.2 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.

9.6 Création du backlog de sprint

La figure 9-13 illustre les contributions, outils et résultats du processus de *création du backlog de sprint*.



9

Figure 9-13 : Crédit à la Scrumstudy pour la création du backlog de sprint.

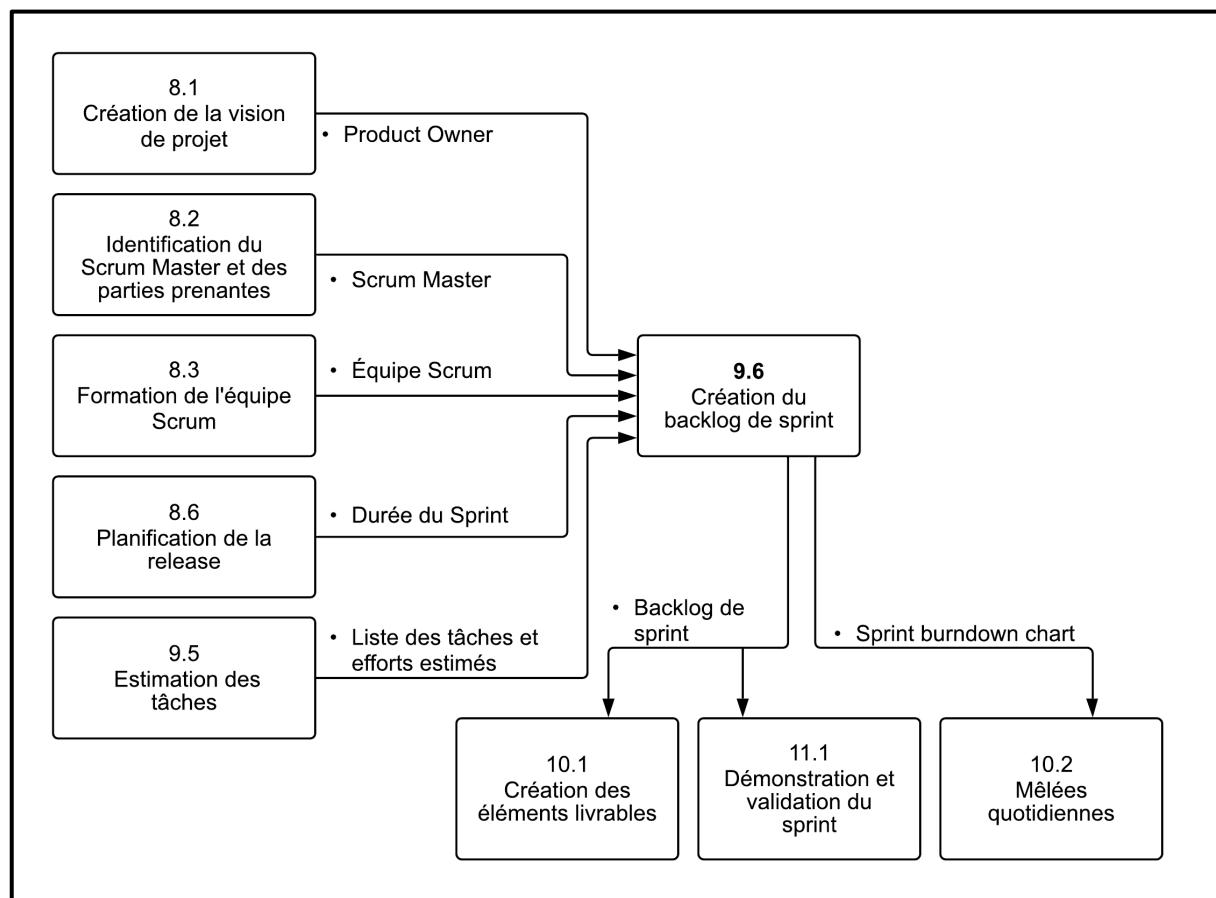


Figure 9-14 : Crédit à la Scrumstudy pour la création du backlog de sprint.

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*

Décrise dans la section 8.4.1.1.

9.6.1.2 Liste des tâches et efforts estimés*

Décrise dans la section 9.5.3.1.

9.6.1.3 Durée du sprint*

Décrise dans la section 8.6.3.2.

9.6.1.4 Dépendances

Décris dans la section 9.4.3.3.

9.6.1.5 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 évènements 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 vitesse, 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.

9.6.3 Résultats

9.6.3.1 Backlog de sprint*

La liste des tâches qui doivent être réalisées par l'équipe Scrum au cours du sprint à venir est appelée backlog de sprint.

Il est courant pour le backlog de sprint d'être représenté sur un scrumboard ou tableau des tâches. Il illustre de façon visuelle le statut de chaque user story dans le backlog. Le sprint backlog inclut également les

risques relatifs aux différentes tâches. Toute activité d'atténuation permettant de résoudre les risques identifiés doit également être incluse comme tâche dans le backlog de sprint.

Une fois que l'équipe Scrum a finalisé et s'engage sur le backlog de sprint, il ne doit pas y être ajouté de nouvelles user stories. Cependant, il pourrait être nécessaire d'ajouter des tâches, relatives aux user stories validées, qui auraient pu être oubliées ou négligées. Si de nouvelles exigences apparaissent durant un sprint, elles seront ajoutées au backlog de produit priorisé global et ajoutées à un prochain sprint.

9.6.3.2 Sprint burndown chart*

Le sprint burndown chart est un diagramme qui représente la quantité de travail restant à effectuer durant le sprint en cours. Le sprint burndown chart initial est accompagné d'un burndown planifié. Le sprint burndown chart doit être mis à jour à la fin de chaque journée au fur et à mesure que le travail est réalisé. Ce diagramme montre les progrès effectués par l'équipe Scrum et permet également de détecter les estimations qui pourraient être incorrectes. Si le sprint burndown chart montre que l'équipe Scrum n'est pas dans les temps pour finir les tâches du sprint, le Scrum Master doit identifier tout obstacle ou entrave à la réussite du sprint et tenter de l'éliminer.

Un diagramme associé est le sprint burnup chart. Contrairement au sprint burndown chart qui montre la quantité de travail restante, le sprint burnup chart illustre le travail réalisé durant le sprint.

9.7 Diagramme de flux de données de la phase de planification et d'estimation

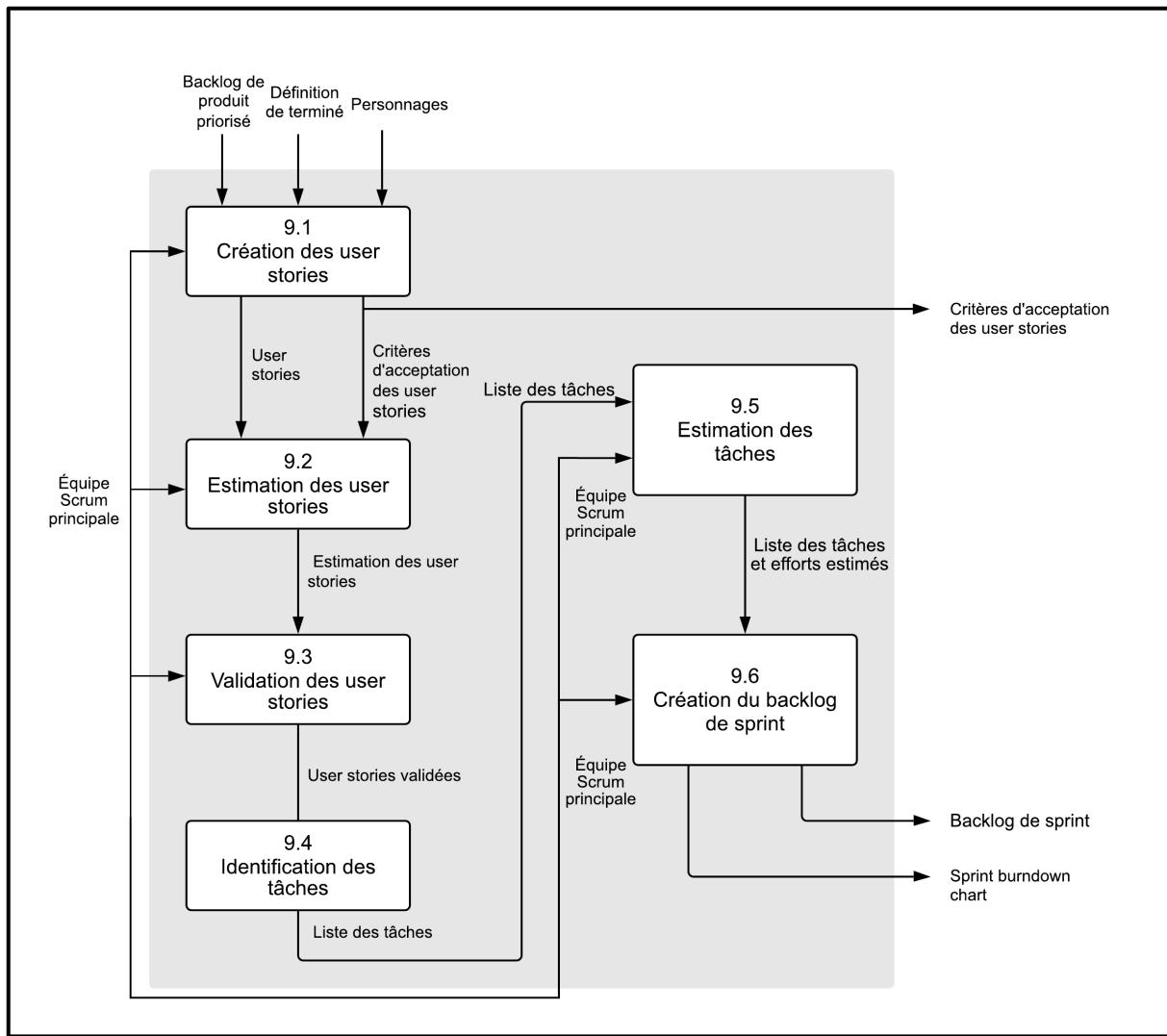


Figure 9-15 : Phase de planification et d'estimation — Diagramme de flux de données

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 mélées 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
- 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 figure 10-1 offre un aperçu des processus de la phase d'implémentation qui sont les suivants :

10.1 Création des incrémentés de produit — l'équipe Scrum travaille sur les tâches du backlog de sprint afin de créer des incrémentés 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.

10.2 Organisation des mélées quotidiennes — une réunion très ciblée et limitée dans le temps est organisée chaque jour. Elle est appelée « mélée 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.

10.3 Affinage du backlog de produit priorisé — c'est le processus au cours duquel le backlog de produit 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 à apporter au backlog, le cas échéant.

10.1 Création des incrémentés de produit	10.2 Organisation des mélées quotidiennes	10.3 Affinage du backlog de produit priorisé
<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum principale* 2. Backlog de sprint* 3. Scrumboard* 4. Registre des obstacles* 5. Calendrier de planification de la release 6. Dépendances 7. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ul style="list-style-type: none"> 1. Expertise de l'équipe* 2. Logiciel 3. Autres outils de développement 4. Expertise Scrum Guidance Body <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Éléments livrables du sprint* 2. Scrumboard mis à jour* 3. Registre des obstacles mis à jour* 4. Demandes de changement non-ratifiées 5. Risques identifiés 6. Risques atténuerés 7. Dépendances mises à jour 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum* 2. Scrum Master* 3. Sprint burndown chart* 4. Registre des obstacles* 5. Product Owner 6. Expérience de la journée de travail précédente 7. Scrumboard 8. Dépendances <p>OUTILS</p> <ul style="list-style-type: none"> 1. Mélée quotidienne* 2. Trois questions quotidiennes* 3. War room 4. Vidéoconférence <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Sprint burndown chart mis à jour* 2. Registre des obstacles mis à jour* 3. Équipe Scrum motivée 4. Scrumboard mis à jour 5. Demandes de changement non-ratifiées 6. Risques identifiés 7. Risques atténuerés 8. Dépendances mises à jour 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum principale* 2. Backlog de produit priorisé* 3. Incréments de produit rejetés 4. Demandes de changement approuvées 5. Demandes de changement non-ratifiées 6. Risques identifiés 7. Backlog de produit de programme mis à jour 8. Registres de rétrospective du sprint 9. Dépendances 10. Calendrier de planification de la release 11. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunions de revue du backlog de produit priorisé* 2. Techniques de communication 3. Autres techniques d'affinage du backlog de produit priorisé <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Backlog de produit priorisé mis à jour* 2. Calendrier de planification de la release mis à jour

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.

10.1 Création des incréments de produit	10.2 Organisation des mêlées quotidiennes	10.3 Affinage du backlog de produit priorisé
<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum principale* 2. Backlog de sprint* 3. Scrumboard* 4. Registre des obstacles* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Expertise de l'équipe* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Éléments livrables du sprint* 2. Scrumboard mis à jour* 3. Registre des obstacles mis à jour* 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum* 2. Scrum Master* 3. Sprint burndown chart* 4. Registre des obstacles* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Mêlée quotidienne* 2. Trois questions quotidiennes* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Sprint burndown chart mis à jour* 2. Registre des obstacles mis à jour* 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum principale* 2. Backlog de produit priorisé* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunion de revue du backlog de produit priorisé* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Backlog de produit priorisé mis à jour*

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 Crédation des éléments livrables

La figure 10-3 illustre les contributions, outils et résultats du processus de *création des éléments livrables*.

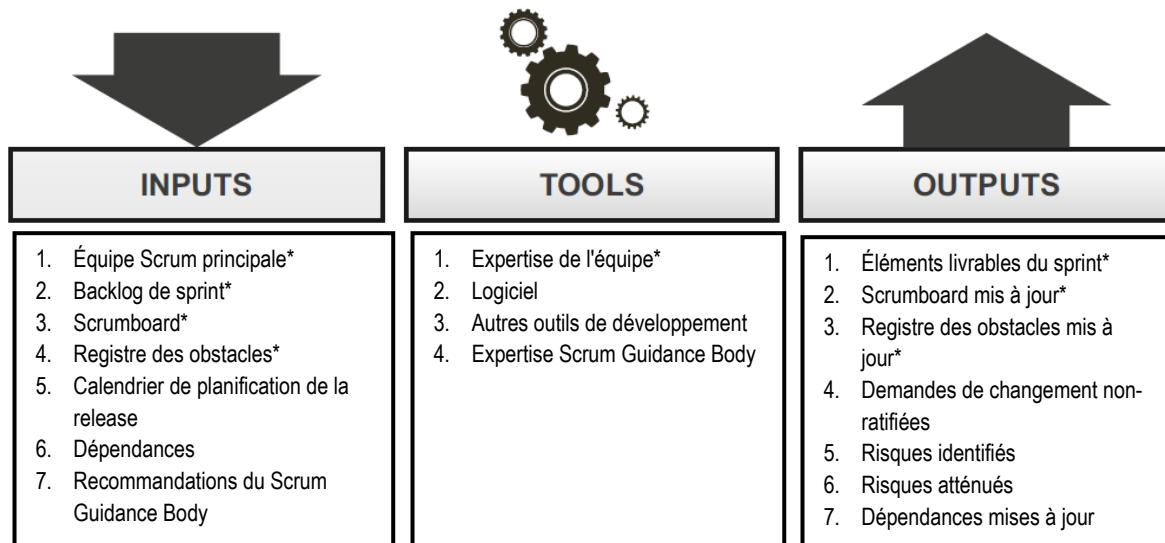


Figure 10-3 : Crédation des éléments livrables — Contributions, outils et résultats

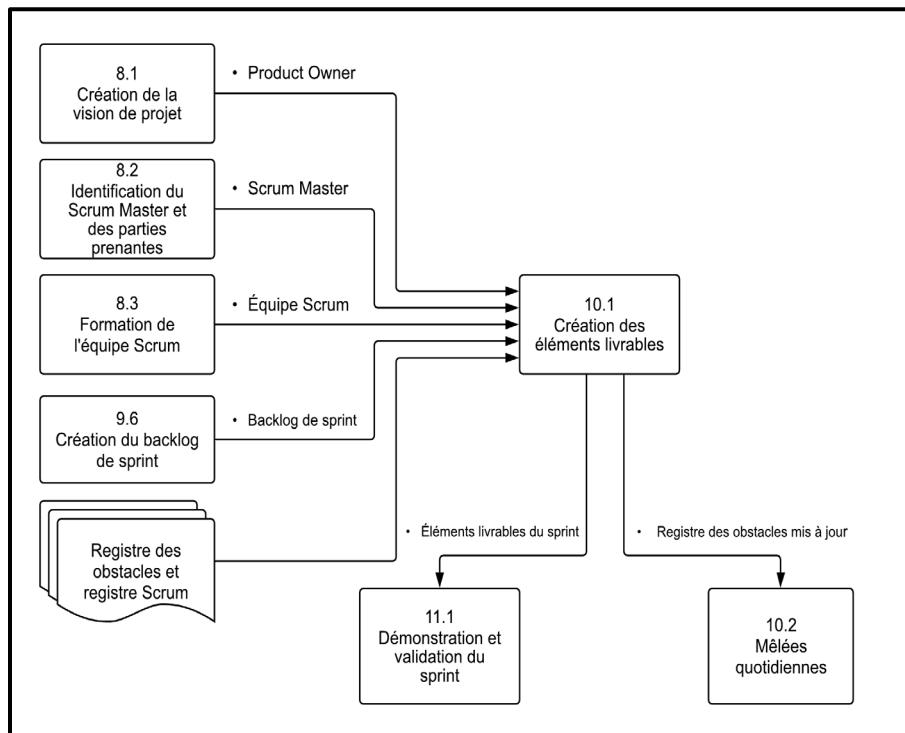


Figure 10-4 : Crédation des éléments livrables — Diagramme de flux de données

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*

Décrise 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*

Dans Scrum, la transparence provient d'outils d'information ouvertement accessibles, tels que le scrumboard, qui illustrent les progrès de l'équipe. L'équipe utilise un scrumboard afin de planifier et de suivre ses progrès au cours de chaque sprint. Le scrumboard est composé de quatre colonnes pour indiquer le 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. Au début d'un sprint, toutes les tâches pour ce sprint sont placées dans la colonne « à faire », elles sont ensuite déplacées en fonction de leurs progrès.

Stories	À faire	En cours	Test	Terminé
1			■ ■ ■	■ ■ ■
2		■ ■ ■ ■	■ ■ ■ ■	■
3	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	
4	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	

Figure 10-5 : Scrumboard

Le scrumboard doit préférablement être maintenu manuellement sur papier ou sur un tableau blanc, mais il peut également être maintenu sous format électronique sur un tableur.

L'équipe Scrum doit modifier ou ajouter des éléments au scrumboard lorsque cela est nécessaire. De cette façon le scrumboard offre des informations visuelles et un moyen de contrôler le travail auquel l'équipe Scrum a donné son accord et s'est engagée.

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.

10.1.1.5 **Calendrier de planification de la release**

Décrit dans la section 8.6.3.1.

10.1.1.6 **Dépendances**

Décrivées dans la section 9.4.3.3.

10.1.1.7 **Recommandations du Scrum Guidance Body**

Décrivées 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 incrémentés 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 les parties prenantes extérieures à l'équipe. Lorsque cela est nécessaire, le Scrum Guidance Body représente une source d'expertise supplémentaire.

10.1.2.2 Logiciel

10

Les outils informatiques automatisés peuvent être utilisés pour la planification, la collecte d'informations et la distribution. Les outils de collaboration virtuelle sont également essentiels pour les projets où l'équipe Scrum est dispersée. Un grand nombre d'outils informatiques automatisés différents sont disponibles. Ils permettent de suivre les progrès, de recueillir et de distribuer des données et contribuent à l'accélération des processus.

10.1.2.3 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.

10.1.2.4 Expertise du Scrum Guidance Body

Décrise 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.

10.1.3 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**

Décrivées dans la section 8.4.1.6.

10.1.3.5 **Risques identifiés**

Décris dans la section 8.4.3.4.

10.1.3.6 **Risques atténus**

Pendant que l'équipe Scrum effectue le travail consistant à créer les incrément 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**

Décrivées dans la section 9.4.3.3.

10.2 Organisation des mélées quotidiennes

La figure 10-6 illustre les contributions, outils et résultats des *mélées quotidiennes*.

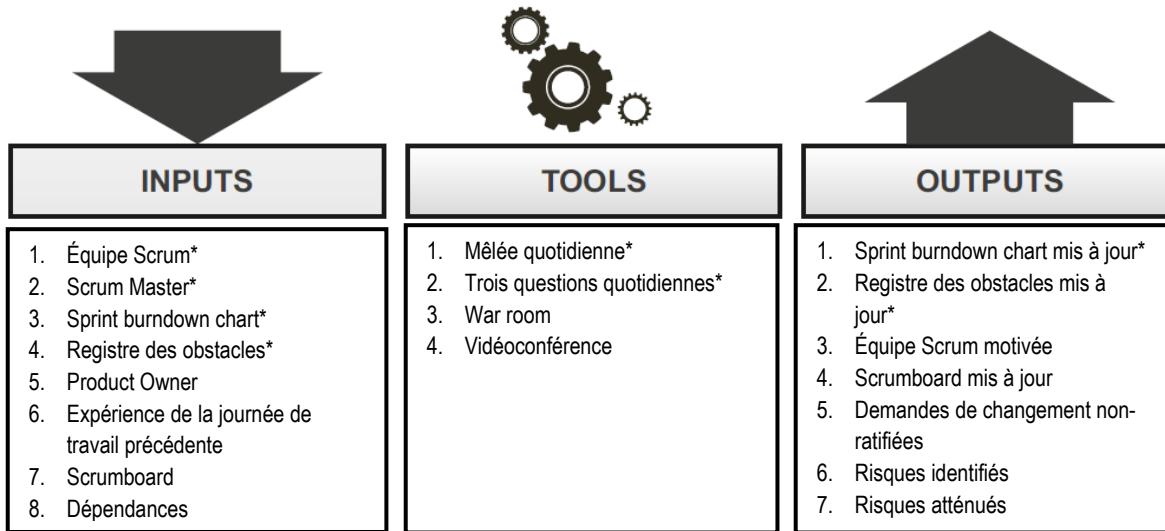


Figure 10-6 : Mélées quotidiennes — Contributions, outils et résultats

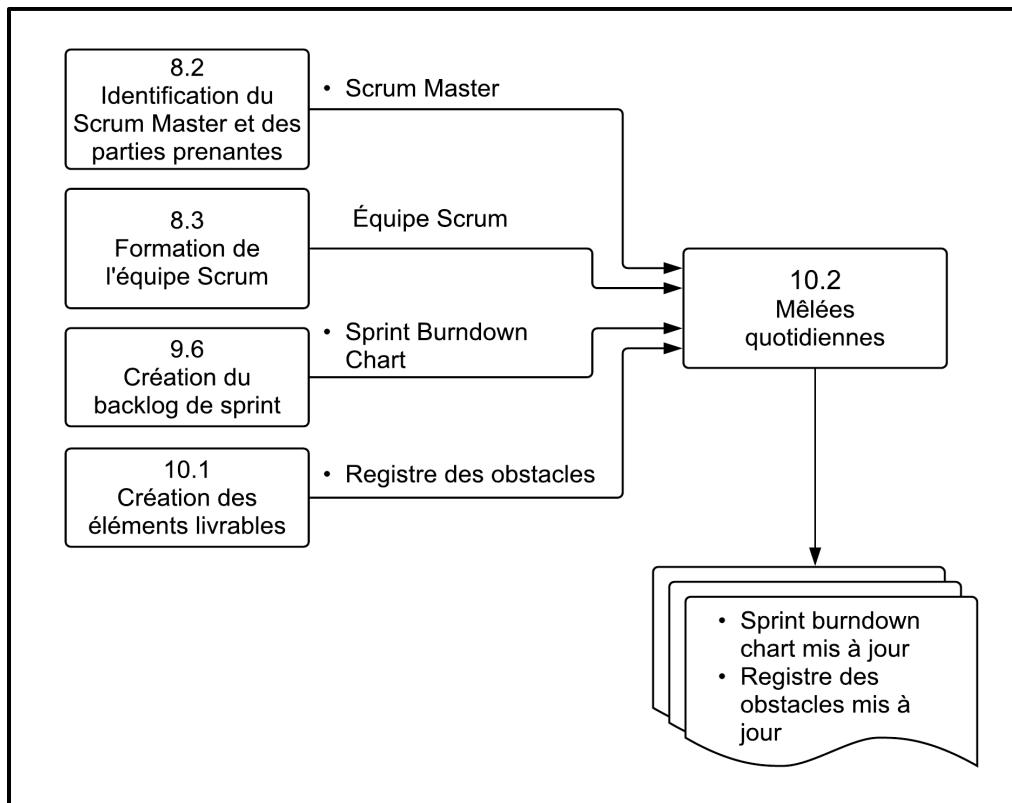


Figure 10-7 : Mélées quotidiennes — Diagramme de flux de données

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

10.2.1 Contributions

10.2.1.1 Équipe Scrum*

Décrise 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 Sprint burndown chart*

Décrit dans la section 9.5.3.2.

10

10.2.1.4 Registre des obstacles*

Décrit dans la section 10.1.1.4.

10.2.1.5 Product Owner

Décrit dans la section 8.1.3.1.

10.2.1.6 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.

10.2.1.7 Scrumboard

Décrit dans la section 10.1.1.3.

10.2.1.8 Dépendances

Décrivées 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.

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.

Les parties prenantes 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.

10.2.3 Résultats

10.2.3.1 Sprint burndown chart mis à jour*

Décrit dans la section 9.6.3.2.

10.2.3.2 Registre des obstacles mis à jour*

Décrit dans la section 10.1.1.4.

10.2.3.3 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'incrément produits.

L'équipe Scrum est décrite dans la section 8.3.3.1.

10.2.3.4 Scrumboard mis à jour

Décrit dans la section 10.1.1.3.

10.2.3.5 Demandes de changement non-ratifiées

Décrivées dans la section 8.4.1.6.

10.2.3.6 Risques identifiés

Décris dans la section 8.4.3.4.

10.2.3.7 Risques atténués

Décris dans la section 10.1.3.6.

10.2.3.8 Dépendances mises à jour

Décrivées dans la section 9.3.3.3.

10.3 Affinage du backlog de produit priorisé

La figure 10-8 illustre les contributions, outils et résultats du processus d'*affinage du backlog de produit priorisé*.

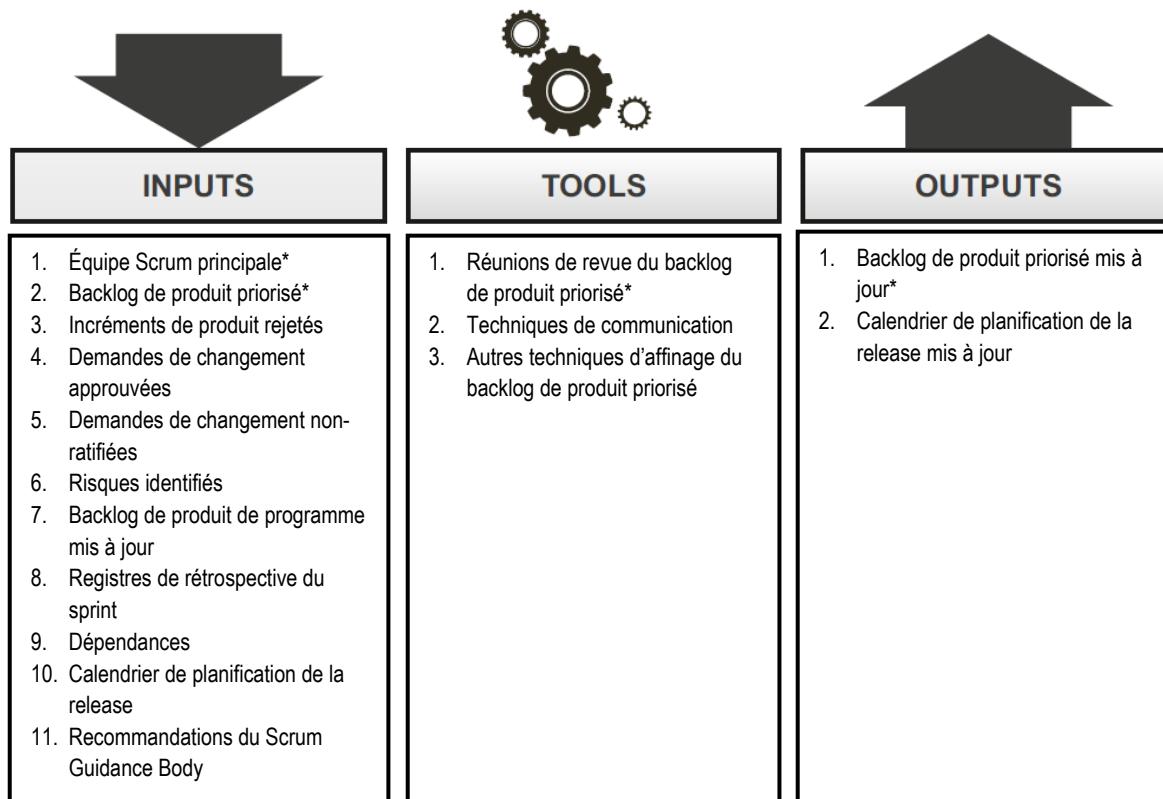


Figure 10-8 : Affinage du backlog de produit priorisé — Contributions, outils et résultats

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

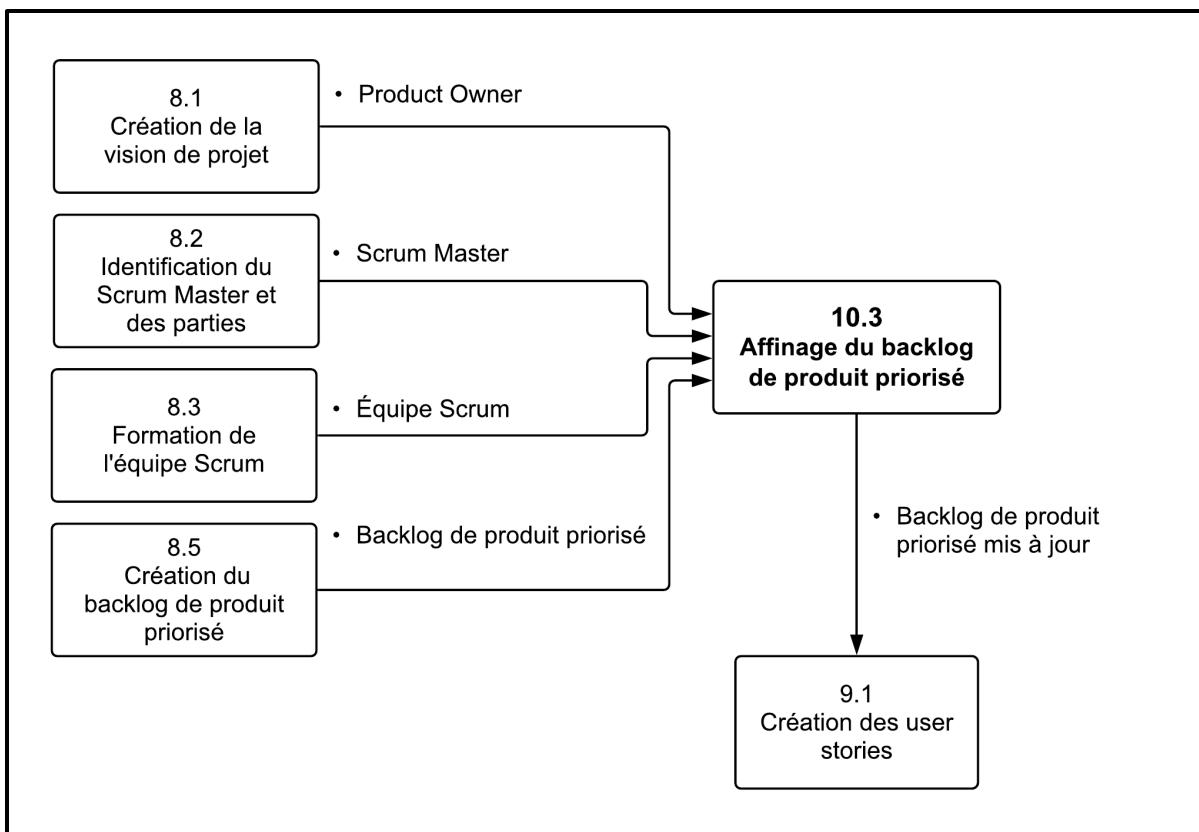


Figure 10-9 : Affinage du backlog de produit priorisé — Diagramme de flux de données

10.3.1 Contributions

10.3.1.1 Équipe Scrum principale*

Décrise 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 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'*affinage du backlog de produit priorisé* et pris en compte pour être développés lors du prochain sprint.

10.3.1.4 Demandes de changement approuvées

Décrivées dans la section 8.4.1.5.

10.3.1.5 Demandes de changement non-ratifiées

Décrivées dans la section 8.4.1.6.

10

10.3.1.6 Risques identifiés

Décrits dans la section 8.4.3.4.

10.3.1.7 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'*affinage du backlog de produit priorisé*.

10.3.1.8 Registres de rétrospective du sprint

Décris dans la section 11.3.3.4.

10.3.1.9 Dépendances

Décris dans la section 9.4.3.3.

10.3.1.10 Calendrier de planification de la release

Décrit dans la section 8.6.3.1.

10.3.1.11 Recommandations du Scrum Guidance Body

Décris dans la section 8.1.1.11.

Lors de l'*affinage du backlog de produit 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 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 les parties prenantes 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'*affinage du backlog de produit 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 (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'affinage du backlog de produit priorisé

D'autres outils d'affinage du backlog de produit 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

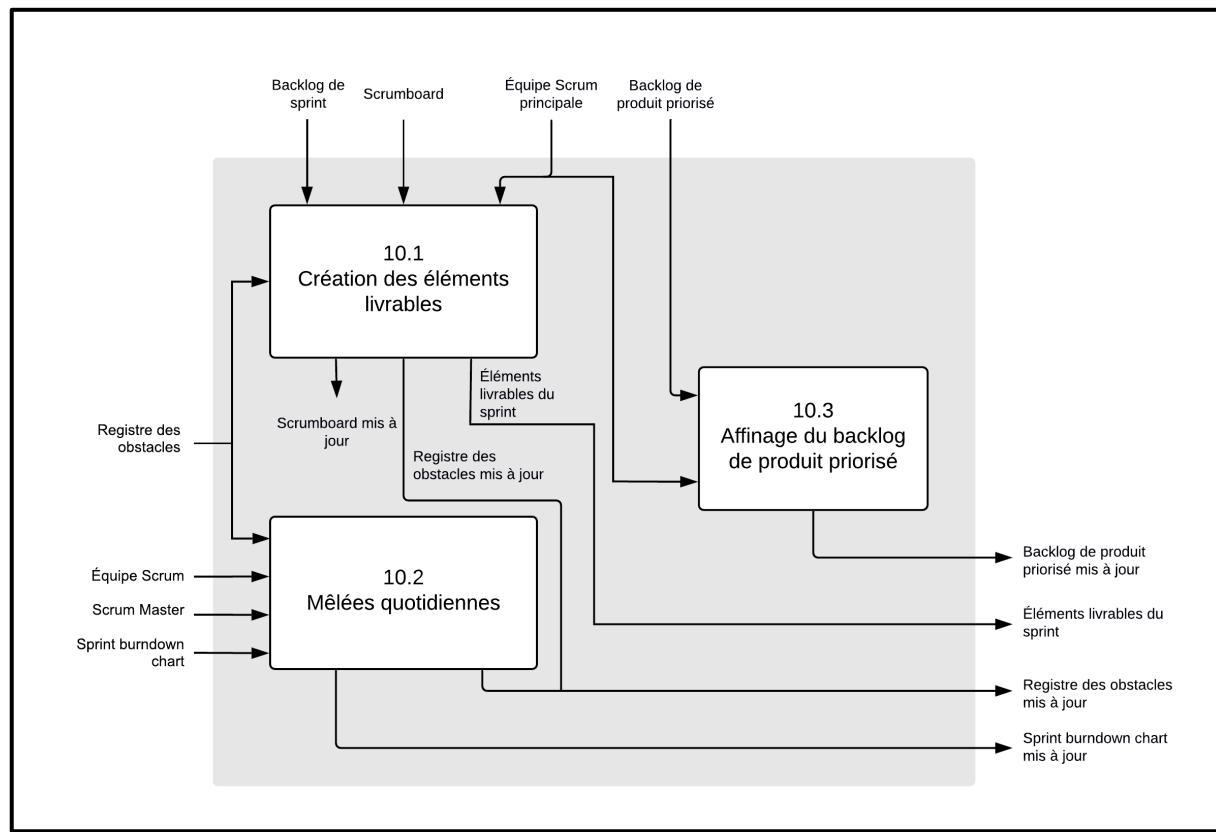


Figure 10-10 : Phase d'implémentation — Diagramme de flux de données

11. REVUE ET RÉTROSPECTIVE

La phase de revue et de rétrospective concerne la revue des incrément 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
- 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 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 incrémentés de produit au Product Owner et aux parties prenantes concernées lors d'une réunion de revue de sprint. L'objectif de cette réunion est d'obtenir l'approbation et l'acceptation du produit ou du service par le Product Owner.

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.

11.1 Démonstration et validation du sprint	11.2 Rétrospective du sprint
<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum principale* 2. Éléments livrables du sprint* 3. Backlog de sprint* 4. Définition de terminé* 5. Critères d'acceptation des user stories* 6. Parties prenantes 7. Calendrier de planification de la release 8. Risques identifiés 9. Dépendances 10. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunions de revue de sprint* 2. Analyse de valeur acquise 3. Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Produits acceptés* 2. Produits livrables rejetés 3. Risques mis à jour 4. Résultats de l'analyse de valeur acquise 5. Calendrier de planification de la release mis à jour* 6. Dépendances mises à jour 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Scrum Master* 2. Équipe Scrum* 3. Résultats de la <i>démonstration et validation du sprint</i>* 4. Product Owner 5. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunion de rétrospective du sprint* 2. EVSP 3. Speed Boat 4. Indicateurs et méthodes de mesures 5. Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Améliorations actionnables convenues* 2. Mesures de suivi et dates d'échéances 3. Éléments non-fonctionnels proposés pour le backlog de produit 4. Registres de rétrospective de sprint 5. Enseignements tirés par l'équipe Scrum 6. Recommandations mises à jour du Scrum Guidance Body

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.

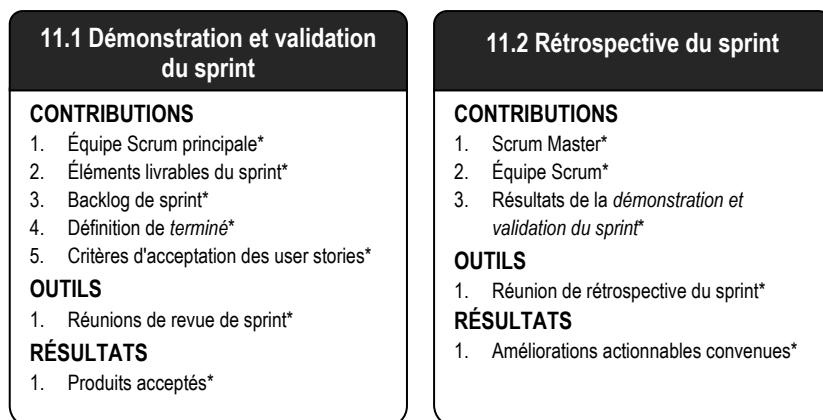


Figure 11-2 : Aperçu de la phase de revue et rétrospective (éléments essentiels)

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

11

11.1 Démonstration et validation du sprint

La figure 11-3 illustre les contributions, outils et résultats du processus de *démonstration et validation du sprint*.

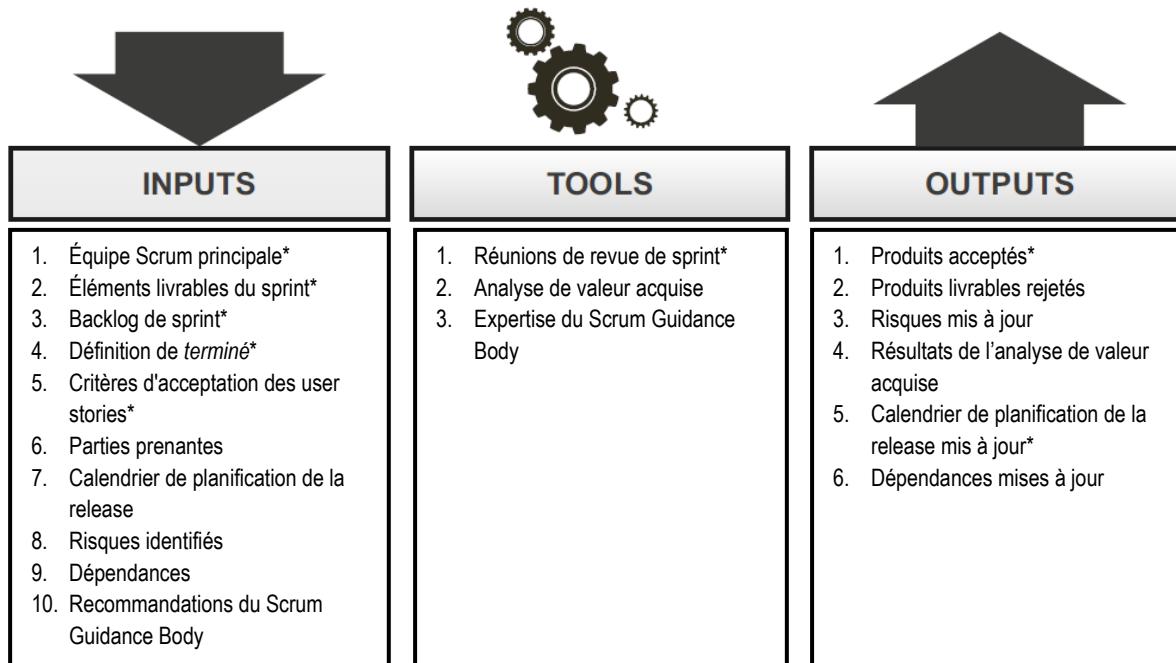


Figure 11-3 : Démonstration et validation du sprint — Contributions, outils et résultats

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

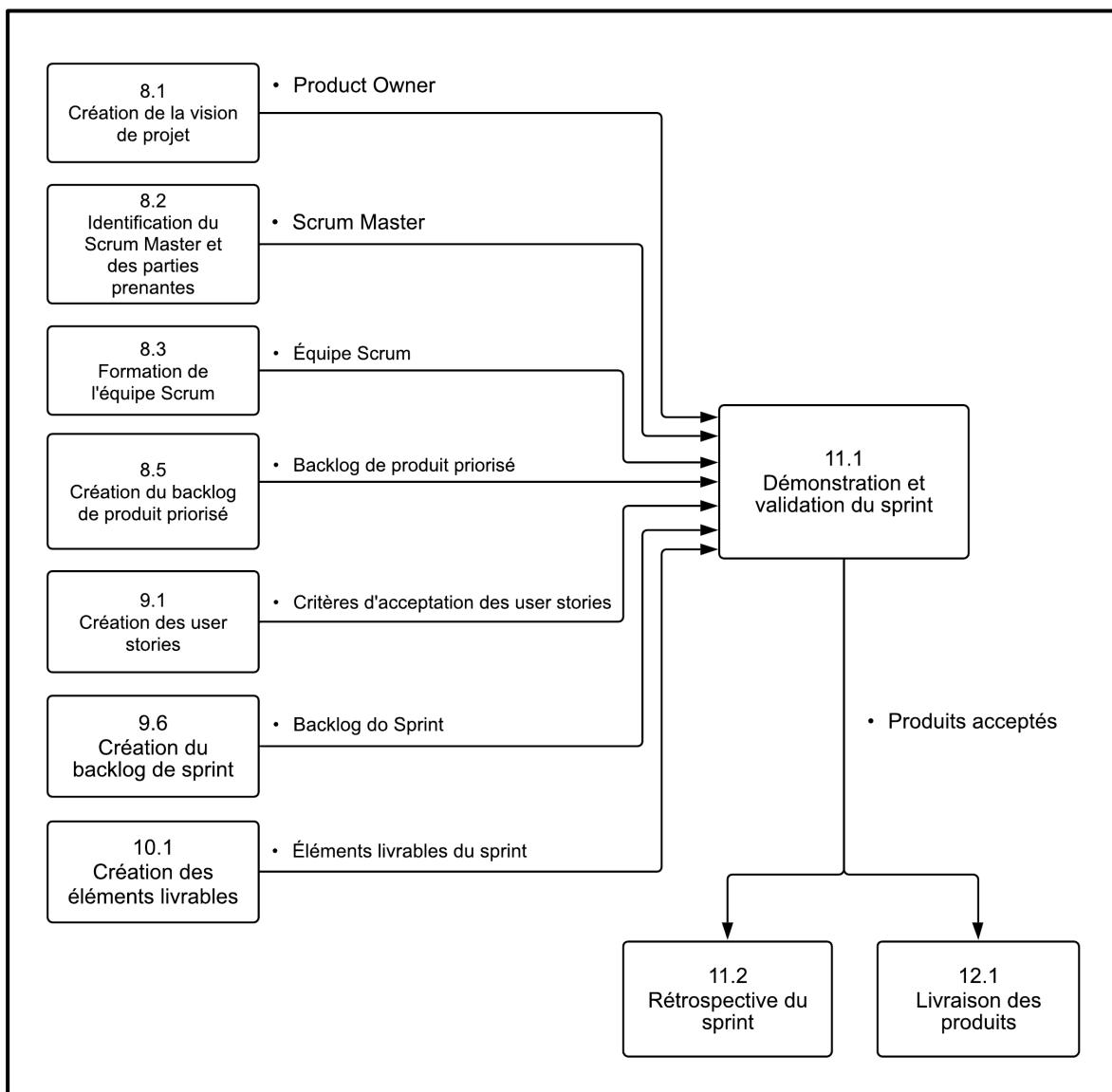


Figure 11-4 : Démonstration et validation du sprint — Diagramme de flux de données

11.1.1 Contributions

11.1.1.1 Équipe Scrum principale*

Décrise dans la section 8.4.1.1.

11.1.1.2 Éléments livrables du sprint*

Décris 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*

Décrise dans la section 8.5.3.2.

11.1.1.5 Critères d'acceptation des user stories*

Décris dans la section 9.4.1.3.

11.1.1.6 Parties prenantes

Décrises 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

Décris 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

Décrivées 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 Réunion de revue de sprint*

Les membres de l'équipe Scrum principale et les parties prenantes concernées participent aux réunions de revue de sprint afin d'accepter les éléments livrables qui remplissent les critères d'acceptation des user stories et de rejeter les incrémentés non acceptés. Ces réunions sont organisées à la fin de chaque sprint. L'équipe Scrum fait la démonstration des résultats du sprint, notamment des nouvelles fonctionnalités ou des produits créés. Cela donne au Product Owner et aux parties prenantes l'opportunité d'inspecter ce qui a été réalisé jusqu'à présent et de déterminer si des changements devraient être apportés au projet ou aux processus dans les sprints à venir.

11.1.2.2 Analyse de valeur acquise

Décrivée dans la section 4.6.1.

11.1.2.3 Expertise du Scrum Guidance Body

Décrivée 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.

11.1.3 Résultats

11.1.3.1 Produits acceptés*

Les produits qui remplissent les critères d'acceptation des user stories sont acceptés par le Product Owner. L'objectif d'un sprint est de créer des éléments potentiellement livrables ou des incrémentés de produit, qui remplissent les critères d'acceptation définis par le client et le Product Owner. Ceux-ci sont considérés comme des produits acceptées et peuvent être livrés au client s'il le désire. Une liste des éléments livrables est maintenue et mise à jour après chaque réunion de revue de sprint. Si un élément livrable ne remplit pas les critères d'acceptation définis, celui-ci n'est pas accepté et sera normalement reporté dans un prochain sprint de façon à rectifier tout problème. Cela n'est pas souhaitable car l'objectif de chaque sprint est de produire des incrémentés qui remplissent les critères d'acceptation.

11.1.3.2 Produits livrables rejetés

Si les incrémentés ne remplissent pas les critères d'acceptation ils sont alors rejetés. Les user stories associées aux incrémentés livrables rejetés sont ajoutées au backlog de produit priorisé de façon à ce que de tels incrémentés puissent être pris en compte au cours d'un prochain sprint.

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

Décrivées 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*.

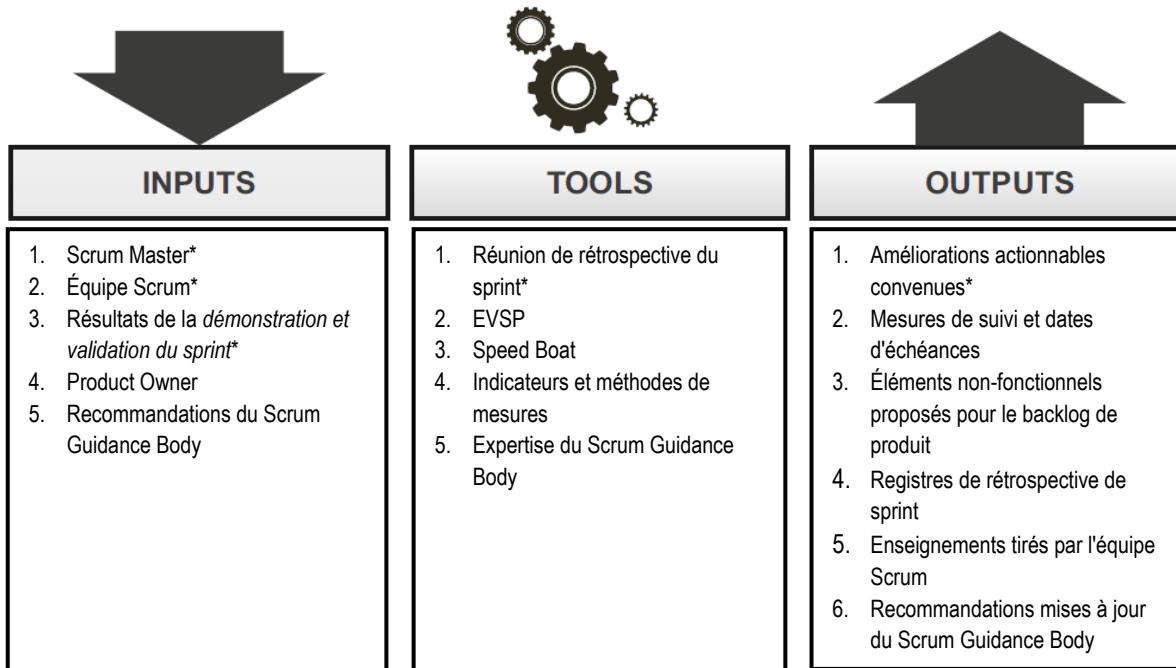


Figure 11-5 : Rétrospective de sprint — Contributions, outils et résultats

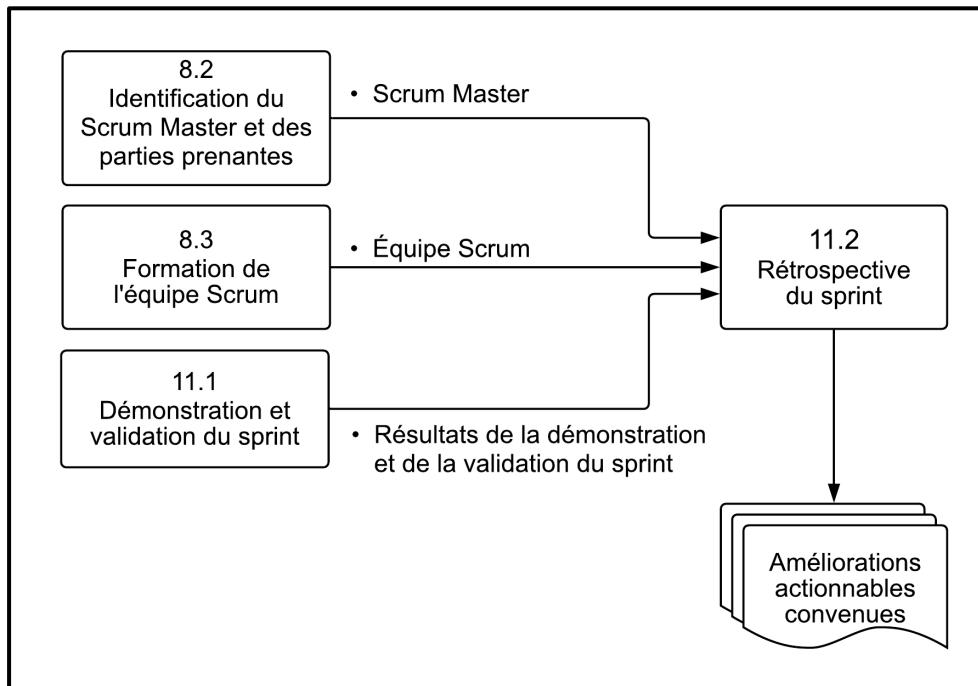


Figure 11-6 : Rétrospective de sprint — Diagramme de flux de données

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*

Décrise 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.

11

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 incrémentés 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 en 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 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

Décrise 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.

11.2.3 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

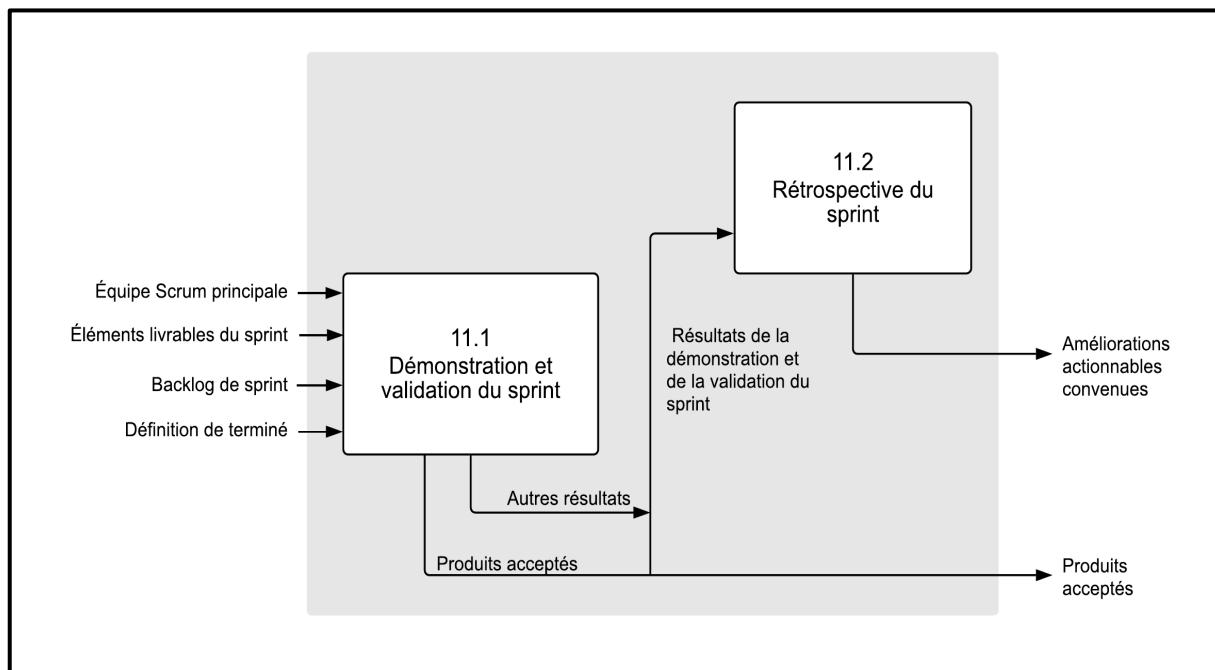


Figure 11-7 : Phase de revue et rétrospective — Diagramme de flux de données

12. RELEASE

La phase de release met l'accent sur la livraison des incrémentés de produits acceptées 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
- 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 figure 12-1 offre un aperçu des processus de la phase de release qui sont les suivants :

12.1 Livraison des produits — les produits acceptés sont livrés ou transférés aux parties prenantes concernées. Un accord officiel sur les produits finis documente la réussite du sprint.

12.2 Rétrospective du projet — ce processus conclue le projet. Les parties prenantes de l'organisation et l'équipe Scrum principale se rassemblent afin d'effectuer une rétrospective du projet 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.

12.1 Livraison des produits	12.2 Rétrospective du projet
<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Product Owner* 2. Parties prenantes* 3. Produits acceptés* 4. Calendrier de planification de la release* 5. Scrum Master 6. Équipe Scrum 7. Critères d'acceptation des user stories 8. Stratégie pilote 9. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ul style="list-style-type: none"> 1. Méthodes de déploiement organisationnel* 2. Stratégie de communication <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Accord sur les produits finis* 2. Produits finis* 3. Releases du produit* 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipe Scrum principale* 2. Parties prenantes 3. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunion de rétrospective du projet* 2. Autres outils pour la <i>réetrospective du projet</i> 3. Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Améliorations actionnables convenues* 2. Mesures de suivi et dates d'échéances* 3. Éléments non-fonctionnels proposés pour le backlog du programme et le backlog de produit priorisé 4. Recommandations mises à jour du Scrum Guidance Body

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.

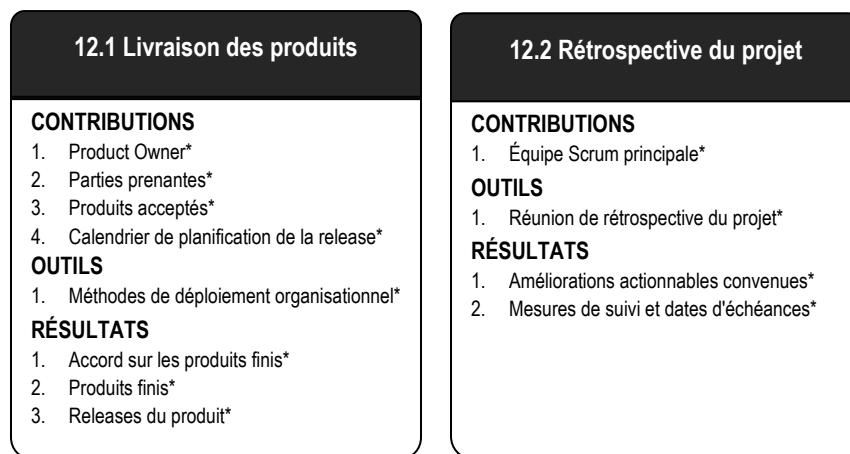


Figure 12-2 : Aperçu de la phase de release (éléments essentiels)

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

12

12.1 Livraison des produits

La figure 12-3 illustre les contributions, outils et résultats du processus de *livraison des produits*.

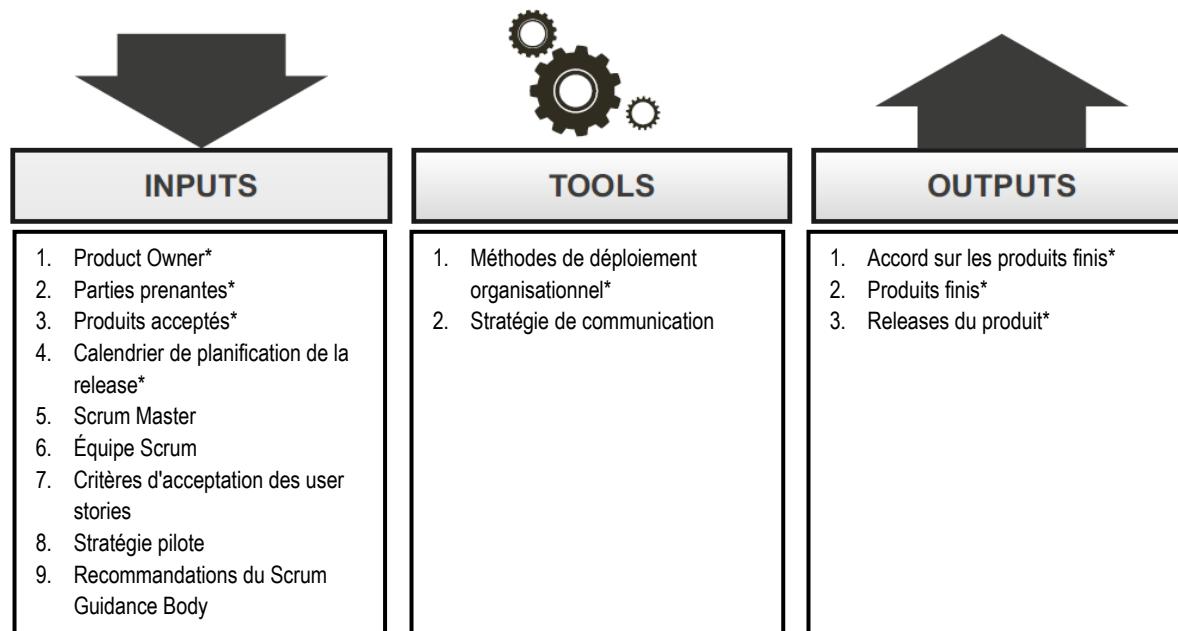


Figure 12-3 : Livraison des produits — Contributions, outils et résultats

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

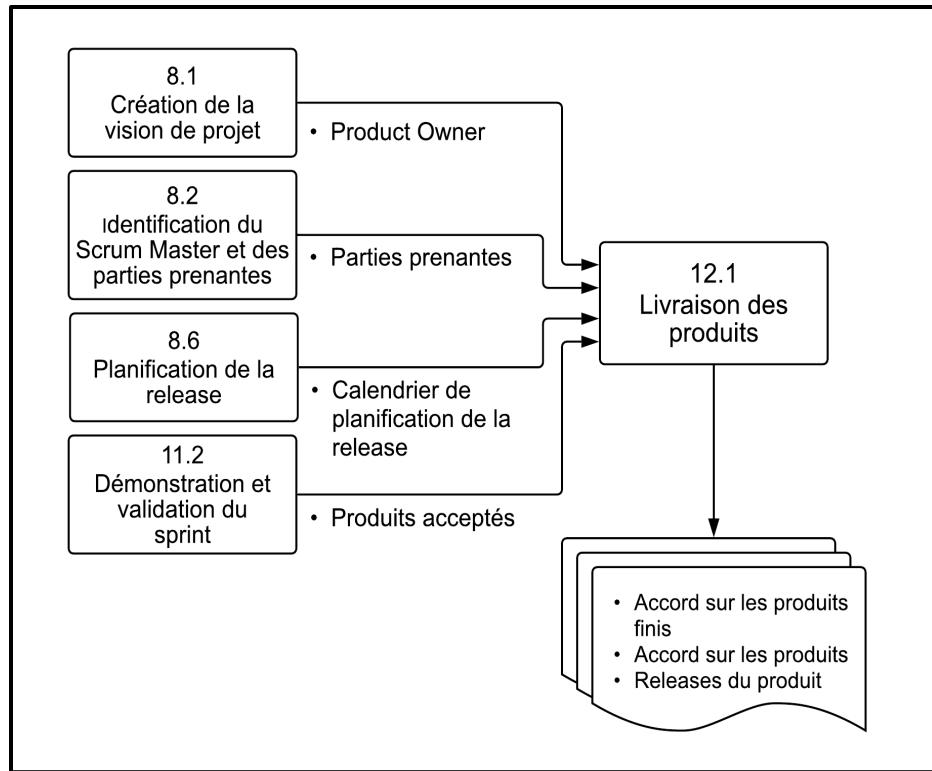


Figure 12-4 : Livraison des produits — Diagramme de flux de données

12.1.1 Contributions

12.1.1.1 Product Owner*

Décrit dans la section 8.1.3.1.

12.1.1.2 Parties prenantes*

Décrivées dans la section 8.2.3.2.

12.1.1.3 Produits acceptés*

Décris 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

Décrise 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.

12.1.1.8 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 les parties prenantes.

12.1.1.9 Recommandations du Scrum Guidance Body

Décris 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 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 incrément 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.

12.1.3 Résultats

12.1.3.1 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. L'approbation formelle du client est essentielle à la comptabilisation des revenus et la responsabilité de son obtention sera définie par les politiques de l'entreprise et ne relève pas nécessairement de la responsabilité du Product Owner.

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 incrémentés de produit sont créés, ils sont continuellement intégrés aux incrémentés 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 du projet

12

La figure 12-5 illustre les contributions, outils et résultats du processus de *rétrospective de projet*.

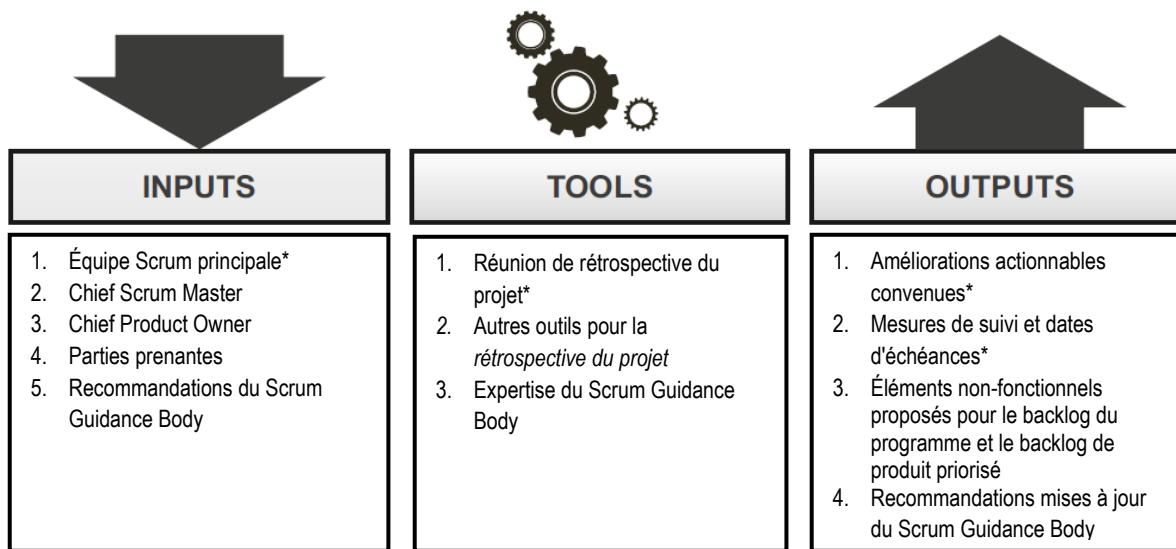


Figure 12-5 : Rétrospective de projet — Contributions, outils et résultats

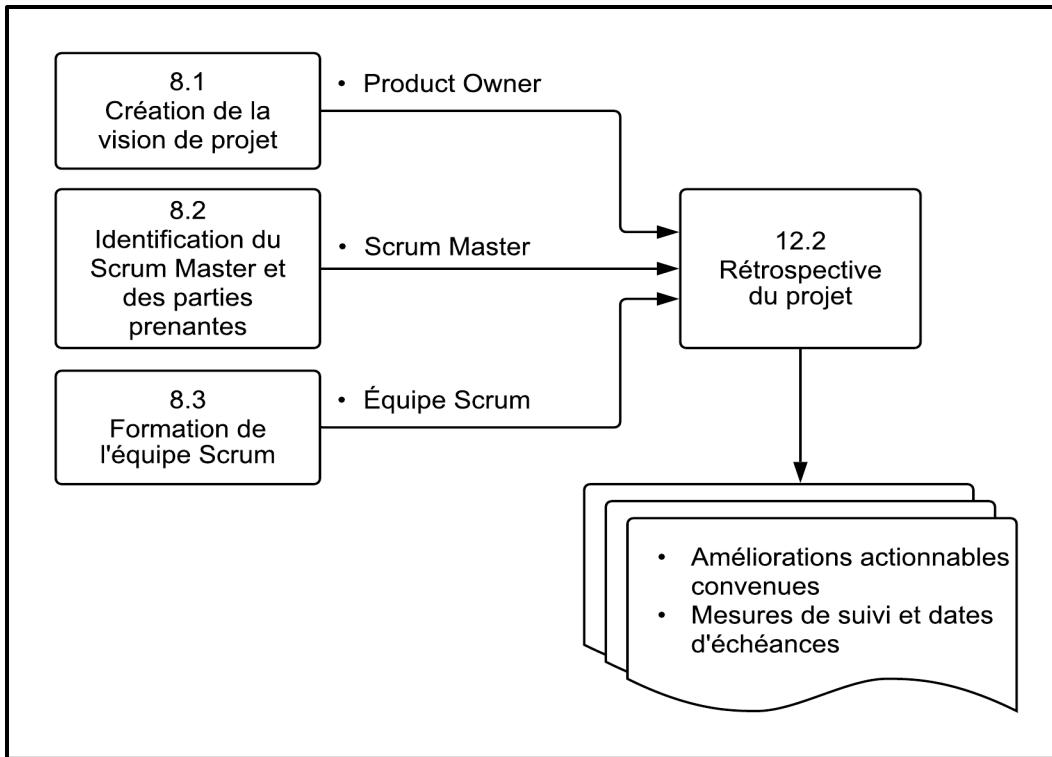


Figure 12-6 : Rétrospective de projet — Diagramme de flux de données

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*

Décrise dans la section 8.4.1.1.

12.2.1.2 Parties prenantes

Décrise dans la section 8.2.3.2.

12.2.1.3 Recommandations du Scrum Guidance Body

Décrivées dans la section 8.1.1.11.

Durant le processus de *rétrospective de projet*, 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 du projet. 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 du projet*

La réunion de rétrospective du projet 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 les parties prenantes. Durant la réunion, les enseignements sont documentés et les participants recherchent des opportunités pour améliorer les processus et résoudre les inefficacités.

12.2.2.2 Autres outils pour la rétrospective du projet

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 du projet*, 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 du projet*. Les recommandations du Scrum Guidance Body peuvent également inclure des suggestions relatives à la façon de mener la réunion de rétrospective du projet.

12.2.3 Résultats

12.2.3.1 Améliorations actionnables convenues*

Décrivées dans la section 11.3.3.1.

12.2.3.2 Mesures de suivi et dates d'échéances*

Décrivées 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 du projet. 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

Décrivées dans la section 11.2.3.6.

12.3 Diagramme de flux de données de la phase de release

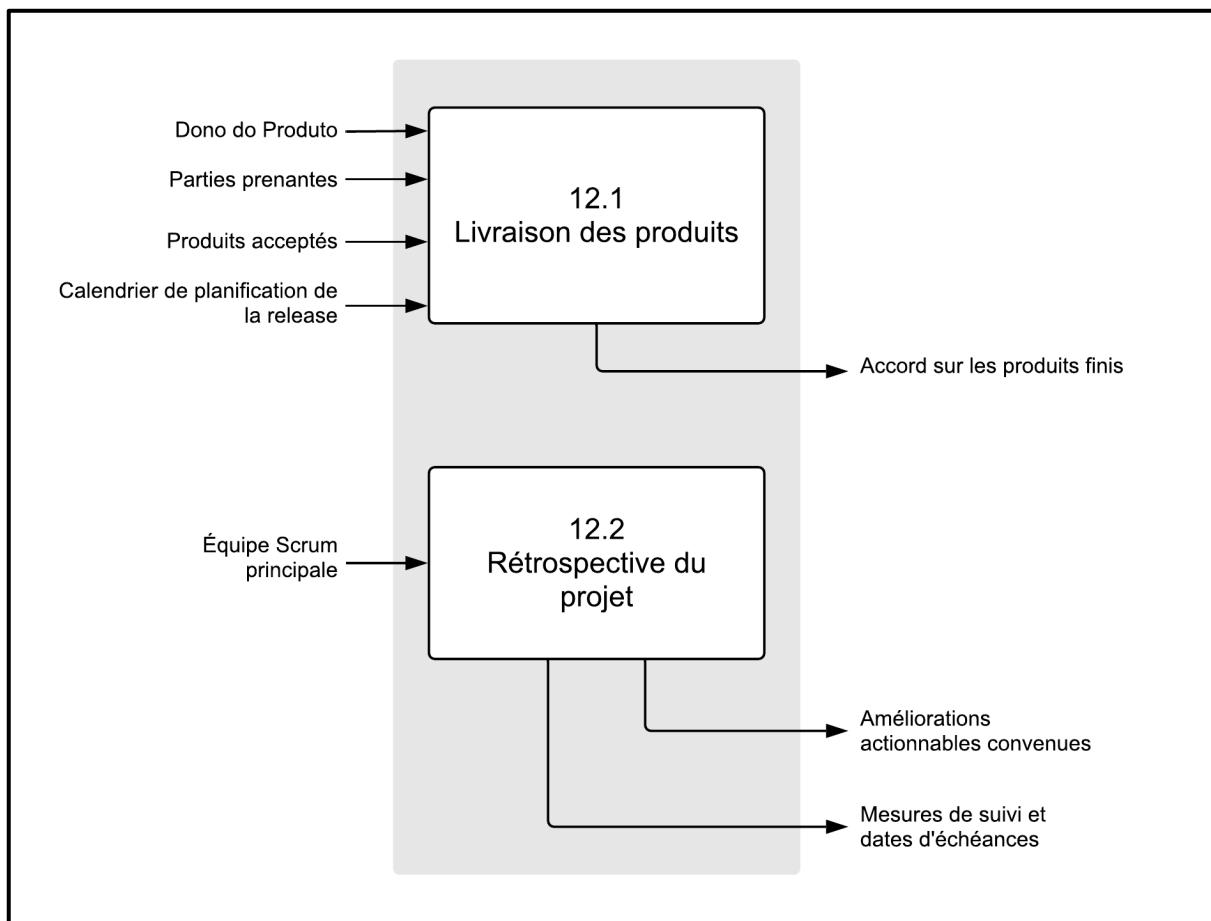


Figure 12-7 : Phase de release — Diagramme de flux de données

13. SCRUM POUR LES GROS PROJETS

Ce chapitre met en avant des aspects supplémentaires de l'approche Scrum qui s'appliquent uniquement aux gros projets. Tel qu'il est décrit dans *A Guide to the Scrum Body of Knowledge (Guide SBOK™)*, le passage à l'échelle de Scrum pour les gros projets s'applique aux éléments suivants :

- les gros projets quel que soit le secteur d'activité
- les produits, les services ou autres résultats destinés à être livrés aux parties prenantes

Le terme *produit* dans le *Guide SBOK™* fait référence à un produit, à un service ou à tout autre résultat. L'approche Scrum n'est pas seulement efficace pour les petits projets quel que soit le secteur d'activité, elle s'applique également aux projets larges et complexes impliquant des centaines de personnes et de nombreuses équipes.

En plus des impacts qu'un gros projet a sur les processus Scrum fondamentaux décrits dans les chapitres 8 à 12, ce chapitre introduit trois processus supplémentaires qui s'appliquent aux gros projets.

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 Chief Product Owners, Product Owners, Chief Scrum Masters, 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 d'une grosse équipe de projet qui coordonne les activités de plusieurs équipes Scrum dans un gros projet afin de concevoir des incrémentés de produit/éléments potentiellement livrables. Des informations supplémentaires relatives à l'utilisation de Scrum pour tout type de projet, gros ou petit, sont disponibles dans les chapitres 2 à 7 qui traitent des principes et aspects Scrum.

Gros projet comparé à un projet Scrum normal

Les processus Scrum fondamentaux définis dans les chapitres 8 à 12 sont valides pour tous les projets Scrum. Les concepts mentionnés dans ces chapitres permettent uniquement de gérer quelques équipes — normalement 1 à 3 équipes Scrum. Les impacts sur ces processus Scrum fondamentaux qui sont uniquement applicables aux plus gros projets sont décrits à la fin de ce chapitre.

Lors de la gestion de plus gros projets, impliquant plus de quatre équipes Scrum, en plus des processus définis dans les chapitres 8 à 12, des processus supplémentaires peuvent être nécessaires afin de répondre aux efforts supplémentaires de coordination et de synchronisation. La définition d'un gros projet peut varier en fonction de l'organisation et du niveau de complexité des projets entrepris. Le principal critère permettant de définir un gros projet par opposition à un petit projet est de compter plusieurs Scrum Masters et/ou Product Owners.

Les raisons qui justifient la nécessité de processus supplémentaires pour les gros projets incluent notamment :

- l'augmentation des interactions et dépendances entre les équipes Scrum au fur et à mesure que le niveau de complexité d'un gros projet augmente
- les besoins en termes de collaboration au sein d'une équipe de Product Owners
- le besoin de gérer des conflits, de résoudre des problèmes et d'établir des priorités au sein de toutes les équipes Scrum
- les besoins en termes de spécialisation puisque certaines équipes Scrum pourraient nécessiter des ressources spécialisées pour des tâches spécifiques — et que ces compétences particulières ne sont pas requises dans toutes les équipes Scrum.
- la nécessité de définir certains standards et directives qui doivent être respectés par toutes les équipes Scrum (p. ex. les normes en termes de sécurité au sein d'une entreprise ou les directives légales ou gouvernementales dans des secteurs d'activité spécifiques). Il est possible que ceux-ci doivent être définis par le Scrum Guidance Body.
- le besoin de créer un environnement ou espace de travail, pour un gros projet qui pourrait ensuite être utilisé par toutes les équipes Scrum.
- le besoin de coordonner les résultats fournis par différentes équipes Scrum afin de créer la release d'un gros projet.

Les processus supplémentaires relatifs au passage à l'échelle de Scrum pour les gros projets sont les suivants :

13.1 Création de composants pour les gros projets — ce processus sert à définir la façon dont plusieurs Product Owners ou plusieurs équipes Scrum travaillent ensemble. Les ressources spécialisées ou communes ainsi que les composants communs sont également identifiés.

13.2 Mener et coordonner les sprints — ce processus n'est normalement pertinent que pour les gros projets. Il aborde des aspects spécifiques qui doivent être pris en compte durant chaque sprint. Le cas

échéant, des réunions Scrum of Scrums sont organisées afin de coordonner les efforts entre plusieurs équipes Scrum.

13.3 Préparation de la release d'un gros projet — pour les gros projets, il est parfois commercialement nécessaire d'organiser un sprint spécial pour préparer la release du produit (l'équipe du projet en prend la décision selon les besoins commerciaux). Ce processus est un sprint de préparation de release.

La figure 13-1 illustre les contributions, outils et résultats du processus de passage à l'échelle de Scrum pour les gros projets.

13.1 Création de composants pour les gros projets	13.2 Mener et coordonner les sprints	13.3 Préparation de la release d'un gros projet
<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Énoncé de vision du projet* 2. Chief Product Owner* 3. Chief Scrum Master* 4. Identification des environnements* 5. Recommandations du Scrum Guidance Body* 6. Product Owners* 7. Scrum Masters* 8. Organisation de l'entreprise 9. Étude de cas 10. Scrum Master du programme 11. Product Owner du programme 12. Matrice d'organisation des ressources <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunion de planification des environnements* 2. Stratégies de communication 3. Planification des ressources des gros projets 4. Détermination des dépendances <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Plan de préparation de la release* 2. Définition minimum de <i>terminé</i> 3. Critères d'acceptation des user stories 4. Ressources partagées 5. Spécialisation de l'équipe 6. Améliorations recommandées du Scrum Guidance Body 7. Stratégie de collaboration des Product Owners 8. Stratégie de collaboration de l'équipe Scrum 9. Dépendances 10. Calendrier d'environnements 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipes principales* 2. Équipe principale élargie* 3. Définition de <i>terminé</i>* 4. Critères d'acceptation des user stories* 5. Dépendances 6. Calendrier d'environnements 7. Plan de préparation de la release 8. Stratégie de collaboration de l'équipe Scrum 9. Stratégie de collaboration des Product Owners 10. Ressources partagées <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunions Scrum of Scrums* 2. Expertise de l'équipe* 3. Réunion d'environnements <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Éléments livrables du sprint* 2. Plan de préparation de la release mis à jour 3. Dépendances résolues 4. Environnement 	<p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Équipes principales* 2. Équipe principale élargie* 3. Calendrier de planification de la release* 4. Plan de préparation de la release* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Stratégies de communication* 2. Sprint de préparation de la release 3. Méthodes de préparation de la release <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Produit livrable* 2. Notes de release 3. Environnement de release 4. Améliorations recommandées du Scrum Guidance Body

Figure 13-1 : Aperçu du passage à l'échelle de Scrum pour les gros projets

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

La figure 13-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de passage à l'échelle de Scrum pour les gros projets.

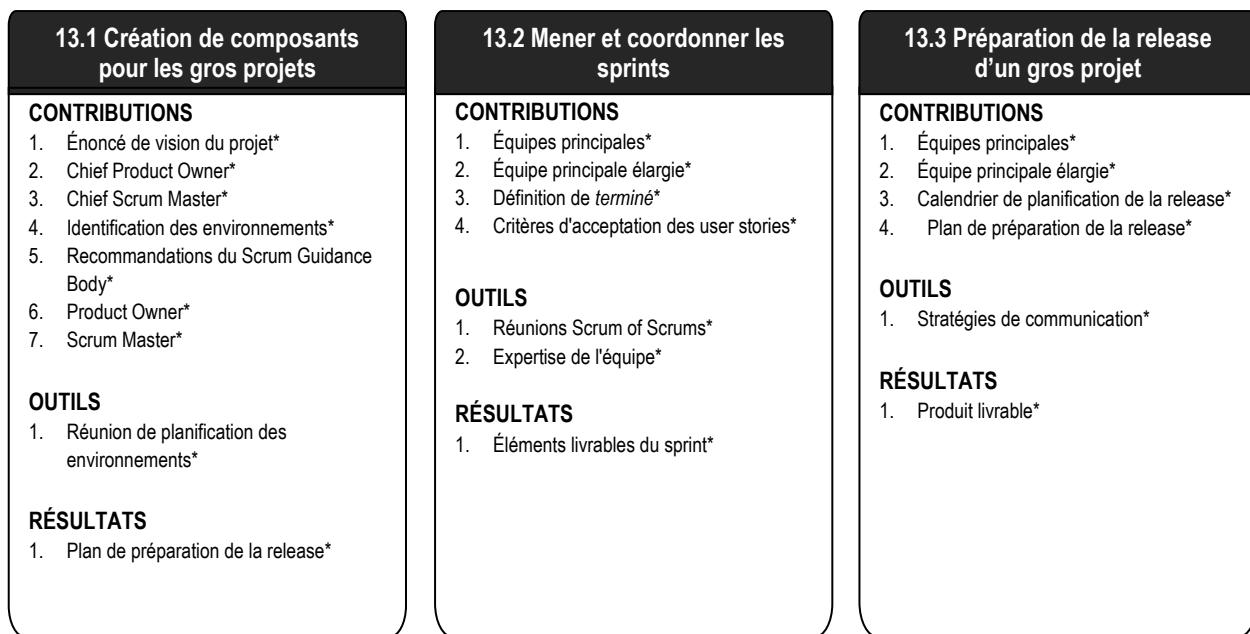


Figure 13-2 : Passage à l'échelle de Scrum pour les gros projets (éléments essentiels)

13.1 Création de composants pour les gros projets

La figure 13-3 illustre les contributions, outils et résultats du processus de *création de composants pour les gros projets*.

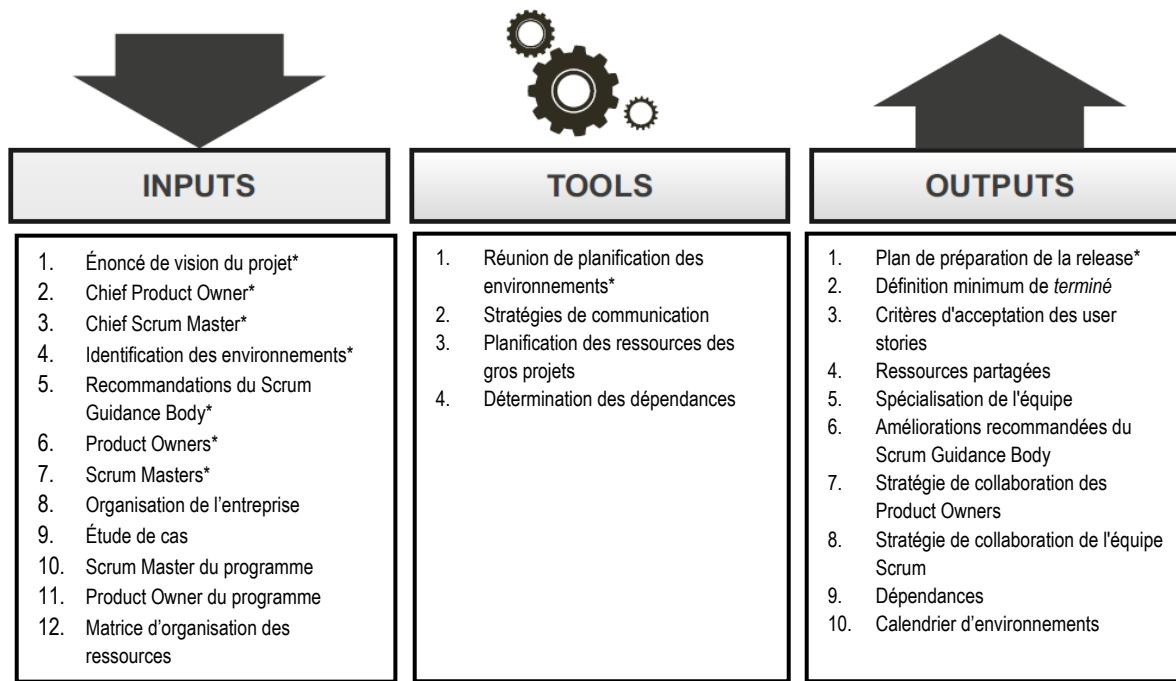


Figure 13-3 : Crédit à la création de composants pour les gros projets — Contributions, outils et résultats

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

La figure 13-4 illustre la relation du processus de *création de composants pour les gros projets* avec les processus Scrum fondamentaux dans la phase d'initiation.

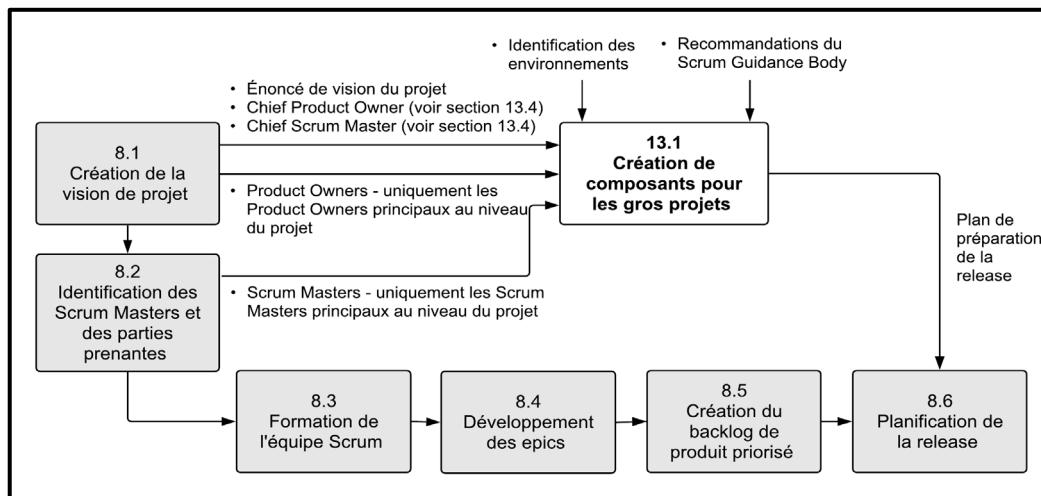


Figure 13-4 : Création de composants pour les gros projets — diagramme de flux de données

13.1.1 Contributions

13.1.1.1 Énoncé de vision du projet*

Décrit dans la section 8.1.3.2.

13.1.1.2 Chief Product Owner*

Décrit dans la section 3.4.2.

13.1.1.3 Chief Scrum Master*

Décrit dans la section 3.5.1.

13.1.1.4 Identification de l'environnement

Dans un gros projet, il est important d'identifier le nombre et le type d'environnements nécessaires puisque de nombreuses équipes Scrum travaillent simultanément afin de réaliser le travail de leurs sprints respectifs. Certains exemples d'environnements incluent notamment : les zones de développement/test logiciel, les zones ou ressources de travail physique ou encore les limites de processus pour chaque équipe Scrum.

13.1.1.5 Recommandations du Scrum Guidance Body*

Décrivées dans la section 8.1.1.11.

Pour les gros projets, le Scrum Guidance Body devient une importante source de référence. Il offre des conseils et promeut les meilleures pratiques de façon à augmenter les chances de succès.

13.1.1.6 Product Owners*

Le rôle de base d'un Product Owner est le même pour les petits comme pour les gros projets et est décrit dans la section 3.4. Cela inclut la collaboration des Product Owners avec leurs équipes Scrum respectives.

La principale différence dans le cas d'un gros projet est que le Product Owner ne prend pas les décisions quotidiennes en matière de priorité comme c'est le cas avec un petit projet. Il fournit simplement des informations et des recommandations au Chief Product Owner. De plus, les interactions avec les parties prenantes sont réparties entre les Product Owners. Chaque Product Owner travaille avec une/des équipe(s) spécifique(s) en fonction des rôles définis. Ces responsabilités et rôles sont définis dans la stratégie de collaboration des Product Owners.

Les Product Owners collaborent avec le Chief Scrum Master, le Chief Product Owner, les Scrum Masters et les autres Product Owners afin de développer la liste des composants et ressources nécessaires et communs à toutes les équipes sur l'ensemble du projet. Ils contribuent également à la révision et à la validation du plan de préparation de la release qui peut inclure un essai d'intégration final de bout en bout.

Seulement quelques Product Owners ou les principaux Product Owners participent à ce processus. Il n'est donc pas nécessaire que tous les Product Owners participent à la *création de composants pour les gros projets*.

13.1.1.7 Scrum Masters*

Le rôle de base d'un Scrum Master est le même pour les petits comme pour les gros projets et est décrit dans la section 3.5.

Un gros projet comprend normalement plusieurs Scrum Masters — chacun facilitant et garantissant un environnement de travail productif pour leur(s) équipe(s) Scrum respective(s) — un Scrum Master peut travailler avec plus d'une équipe Scrum. Les Scrum Masters sont identifiés lors du processus d'*identification du Scrum Master et des parties prenantes* (voir section 8.2).

La collaboration entre plusieurs équipes Scrum, facilitée par les Scrum Masters, est définie dans la stratégie de collaboration des équipes Scrum (voir section 13.1.3.8).

Les Scrum Masters collaborent avec le Chief Scrum Master, le Chief Product Owner, les autres Scrum Masters et les Product Owners afin de développer la liste des composants et ressources nécessaires et communs à toutes les équipes sur l'ensemble du projet. Ils fournissent également des informations pour la création du plan de préparation de la release.

Seulement quelques Scrum Masters ou les principaux Scrum Masters participent à ce processus. Il n'est donc pas nécessaire que tous les Scrum Masters participent à la *création de composants pour les gros projets*.

13.1.1.8 Organisation de l'entreprise

Il est recommandé que les organisations qui prévoient d'utiliser Scrum pour de gros projets adoptent l'approche Scrum. Pour un gros projet, l'organisation doit être capable de supporter l'effort en engageant les ressources nécessaires. Si elle en est incapable, des plans doivent être mis en place afin d'obtenir plus de ressources telles que du personnel, des outils ou des espaces de travail. Il est impératif que l'entreprise qui prévoit d'utiliser Scrum soit prête à modifier sa culture de travail et ses habitudes de façon à profiter pleinement des avantages associés à l'utilisation de l'approche Scrum.

13.1.1.9 Étude de cas

Décrise dans les sections 4.4.1, 4.4.2 et 8.1.1.1.

Bien qu'une étude de cas pour un gros projet ne soit pas différente d'une étude de cas pour un petit projet, une justification commerciale solide devient de plus en plus importante au fur et à mesure que la taille d'un projet augmente. Les bons, et plus particulièrement les mauvais, impacts commerciaux associés au choix d'un projet sont beaucoup plus élevés en raison de la somme d'argent et des autres ressources investies dans le projet.

13.1.1.10 Scrum Master du programme

Décrit dans la section 3.5.2.

13.1.1.11 Product Owner du programme

Décrit dans la section 3.4.3.

13.1.1.12 Matrice d'organisation des ressources

Dans le cas de gros projets Scrum avec des équipes Scrum en compétition pour l'allocation de ressources et la priorisation des tâches, il est important de gérer les ressources organisationnelles de façon optimale afin d'atteindre les objectifs globaux du projet. Les Product Owners doivent gérer la création de la valeur en mettant en place des plans organisationnels. Les plans organisationnels doivent inclure les composants qui seront développés ainsi que les compétences, les coûts et les autres ressources requises afin de les développer, les vélocités actuelles des équipes Scrum afin d'estimer la durée des projets, les exigences en termes de communication et les autres interfaces que les équipes Scrum doivent maintenir.

Décrise dans la section 8.2.1.8.

13

13.1.2 Outils

13.1.2.1 Réunion de planification des environnements*

Une réunion de planification des environnements sert à définir un calendrier de partage des environnements pour l'équipe Scrum. Par exemple, avec des équipes dispersées travaillant dans différents fuseaux horaires, il est possible de réaliser des tests 24h/24 afin d'optimiser l'utilisation des environnements. Il est donc impératif de créer un programme afin de déterminer les heures de test pour chaque équipe. Pour les projets logiciels, le plan des environnements peut également inclure des informations sur la façon dont le code est promu dans chaque environnement et par qui.

13.1.2.2 Stratégie de communication

Les stratégies de communication sont essentielles dans les gros projets puisqu'une mauvaise ou un manque de communication entre plusieurs équipes Scrum peut nuire aux efforts de collaboration et entraîner l'échec du projet. Les stratégies de communication doivent aborder les messages clés, les méthodes de communication, les canaux ou mécanismes permettant de communiquer les messages clés aux parties prenantes concernées (y compris la manière dont le Chief Product Owner et les autres Product Owners communiquent les priorités aux équipes Scrum), les responsabilités en termes de communication,

la classification des informations sensibles, le timing des activités de communication et les processus permettant d'évaluer l'efficacité des communications. Les stratégies de communication doivent également inclure le timing et la fréquence des réunions Scrum of Scrums (SoS) ainsi que la façon dont ces réunions se dérouleront.

Chaque équipe Scrum peut disposer de sa propre stratégie de communication qui décrit les documents à créer et maintenir tout au long du projet ainsi que les différentes méthodes utilisées pour transmettre les informations importantes aux parties prenantes. Elle spécifiera également les personnes responsables des différentes activités de communication. Voir section 12.1.2.2.

13.1.2.3 Planification des ressources des gros projets

La planification des ressources des gros projets est essentielle étant donnée la complexité de l'allocation des différents types de ressources aux différentes équipes Scrum travaillant en parallèle. Il existe des besoins concurrents pour les ressources rares et le Chief Product Owner ainsi que les autres Product Owners doivent planifier la livraison de la meilleure valeur dans un délai le plus court possible. La planification des ressources dans les gros projets doit prendre en compte les différents frais associés aux ressources telles que le personnel, les matériels et logiciels, les services externes et l'espace physique.

Le Chief Product Owner et les autres Product Owners peuvent avoir à coordonner des sources externes afin d'acquérir des ressources et d'augmenter le personnel (p. ex. il est possible d'avoir à employer des ressources externes afin de travailler avec les équipes permanentes, elles pourraient également être amenées à interagir avec l'équipe de gestion fournisseur au sein de l'entreprise). Lors de l'emploi de ressources externes, le Chief Product Owner et l'équipe doivent respecter les politiques de l'entreprise en termes de relation avec les ressources externes et les fournisseurs.

Dans les gros projets, le Chief Product Owner peut avoir à prendre en compte la planification de ressources supplémentaires afin d'aborder les besoins des équipes spécialisées et le besoin d'établir des environnements pour plusieurs équipes Scrum travaillant en parallèle. Le Chief Product Owner et les Product Owners peuvent collaborer avec les Scrum Masters et les équipes Scrum afin de définir les compétences spécialisées nécessaires pour un gros projet, le nombre de ressources requises, les équipes Scrum qui ont besoin de compétences spécialisées, et l'estimation des allocations.

13.1.2.4 Détermination des dépendances

Décrise dans la section 9.4.2.3.

L'identification correcte des dépendances dans les gros projets permet aux équipes Scrum de déterminer lesquelles de leurs décisions et activités pourraient avoir un impact sur les autres équipes. Cela peut également influencer l'ordre relatif dans lequel une équipe Scrum exécute ses tâches respectives afin de créer les incrément de produit du sprint.

13.1.3 Résultats

13.1.3.1 Plan de préparation de la release*

Étant donné que chaque sprint crée un produit potentiellement livrable ou un autre incrément de produit dans un petit projet, une release peut être effectuée après n'importe quel sprint lorsque cela est commercialement justifiable. Dans un gros projet, certaines activités relatives à la préparation de la release doivent être réalisées pour chaque sprint. Par exemple, l'équipe d'un projet peut décider de conduire un ensemble complet de tests de performance chers et coûteux en temps ou de réaliser un ensemble spécial de tests d'intégration de bout en bout juste avant la release. Ces activités sont en dehors de la définition de *terminé* établie pour les sprints normaux.

Dans un tel cas, un sprint séparé de préparation de la release sera nécessaire (voir 13.3 *préparation de la release d'un gros projet*). Ces décisions commerciales et les justifications commerciales associées sont décrites dans un plan de préparation de la release. Le plan de préparation de la release détaille les étapes à suivre par chaque équipe Scrum et par le projet dans son ensemble afin de confirmer que les exigences minimales pour la release ont été atteintes et que le produit ou le composant est prêt pour la release.

13

13.1.3.2 Définition minimum de *terminé*

Décrise dans la section 5.4.3.

La définition de *terminé* est décrite dans la section 8.5.3.2.

Dans les gros projets, certains aspects peuvent être considérés comme définition minimum de *terminé*, tels que le fait de déménager dans un environnement spécifique ou d'atteindre certains critères internes avant de passer les tests de validation utilisateur (UAT).

En plus de la définition minimum de *terminé*, les Product Owners et leurs équipes Scrum peuvent établir des définitions supplémentaires de *terminé* qui s'appliquent à des user stories spécifiques (p. ex. toutes les user stories qui conduisent à la création ou à des modifications sur un site Web). Si des définitions supplémentaires sont définies au-delà de la définition minimum de *terminé*, celles-ci doivent être prises en compte par les équipes Scrum qui développent les produits. Des exemples de définition de *terminé* peuvent inclure des listes pouvant être utilisées au niveau du sprint ou de la release.

13.1.3.3 Critères d'acceptation des user stories

Décris dans la section 9.1.3.2.

13.1.3.4 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 projet important, les ressources partagées peuvent être limitées et requises par plusieurs 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 développer une méthode permettant de déterminer comment ces ressources partagées seront attribuées et utilisées. Dans les gros projets, les responsabilités concernant les éléments livrables du sprint sont partagées entre plusieurs équipes Scrum, chacune disposant de priorités et de disponibilités différentes. Un exemple de méthode pour l'attribution de ressources partagées peut être de garantir que les ressources sont d'abord attribuées aux fonctionnalités les plus importantes/à plus forte valeur et aux équipes qui y travaillent. Lorsque des demandes concurrentes présentent des priorités très similaires le Chief Product Owner doit décider de l'attribution en fonction des exigences commerciales actuelles.

13.1.3.5 Spécialisation de l'équipe

Dans le cas d'un projet important, la spécialisation de l'équipe peut être nécessaire. Il existe trois dimensions à la spécialisation d'une équipe.

La première est le besoin de réaliser des tâches spécifiques. Par exemple, une équipe spécialisée peut être une équipe d'intégration disposant de connaissances spécialisées en termes d'intégration continue. Ces connaissances peuvent être particulièrement importantes lors de l'exécution d'un sprint de préparation de la release.

La deuxième dimension à prendre en compte est le besoin en termes de compétences spécifiques pour des membres individuels de l'équipe. En théorie, tous les membres de l'équipe Scrum sont des généralistes et des spécialistes dans le sens où ils disposent de connaissances dans différents domaines et sont experts dans au moins l'un d'entre eux. Cependant, cela n'est peut-être pas le cas dans un gros projet. Il peut être nécessaire pour les membres d'équipes spécialisées de posséder des compétences spécifiques — telles que des connaissances spécifiques dans un domaine comme la sécurité — qui ne sont pas disponibles dans les équipes du gros projet pour lequel elles sont nécessaires. Pour un gros projet, il peut s'avérer très coûteux de former tout le monde dans tous les domaines. Ces experts aux compétences et connaissances spécialisées peuvent travailler dans différentes équipes en tant que membres temporaires d'une équipe. Autrement, il peut être parfois nécessaire de les embaucher à partir de sources externes. Il est important de se rappeler que l'addition d'un nouveau membre aura un impact sur la vitesse de l'équipe.

La troisième dimension est qu'il peut y avoir des limites à la flexibilité de l'équipe. Comme cela a été dit ci-dessus, dans un gros projet, il peut s'avérer très coûteux de former tout le monde dans tous les domaines. Cela signifie que chaque équipe disposera d'un niveau d'expertise important dans un ou plusieurs domaines. Elle pourra travailler dans quelques domaines différents avec de la formation et de l'aide. Mais il existe d'autres domaines dans lesquels elle sera incapable de travailler. Lorsqu'il est question de la

planification d'un sprint, chaque équipe doit disposer d'un sous-ensemble d'user stories qui sont logiquement les siennes en fonction de son degré d'expertise, certaines sur lesquelles elle peut travailler et certaines sur lesquelles elle ne peut pas travailler car elle ne possède pas les connaissances/compétences adéquates.

Cela résulte en un certain niveau de risque pour le projet. Il n'est pas toujours possible de réaliser toutes les user stories au plus fort niveau de priorité en un seul sprint. Les équipes peuvent avoir à travailler sur des user stories à plus faible niveau de priorité lorsqu'elles attendent la disponibilité de membres disposant de compétences spécialisées.

13.1.3.6 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.

13.1.3.7 Stratégie de collaboration des Product Owners

La stratégie de collaboration des Product Owners définit comment plusieurs Product Owners collaborent avec le Chief Product Owner. Au minimum, elle doit définir le nombre d'équipes Scrum qu'un Product Owner peut gérer (en fonction de son expérience, du temps et de sa connaissance du domaine), la façon dont les tâches consistant à collecter les exigences des parties prenantes sont distribuées entre les Product Owners, la façon dont le backlog de produit priorisé est mis à jour avec les nouvelles exigences ou avec les exigences modifiées et la façon dont les Product Owners collaborent avec plusieurs équipes Scrum. Il est important de noter que chaque équipe Scrum collabore avec seulement un Product Owner, cependant, un même Product Owner peut travailler avec plus d'une équipe Scrum si cela est nécessaire.

13.1.3.8 Stratégie de collaboration de l'équipe Scrum

La stratégie de collaboration de l'équipe Scrum définit comment les différentes équipes Scrum collaborent ensemble de façon à offrir le plus de valeur ajoutée en un délai le plus court possible. Cette stratégie doit inclure des informations sur les domaines spécifiques attribués aux équipes qualifiées, la façon dont les équipes participent à l'affinage et à l'estimation du backlog de produit priorisé (c-à-d. décider lequel des membres de l'équipe participe aux sessions d'affinage et aux exercices d'estimation de haut niveau) et la façon dont les équipes organisent les réunions Scrum of Scrums (SoS).

Cette stratégie peut également inclure des informations relatives à la façon dont chaque équipe Scrum est coachée (p. ex. présence d'un coach supplémentaire en plus du Scrum Master, présence d'un Scrum

Master à chaque emplacement dans le cas d'équipes dispersées, la façon dont les membres de l'équipe collaborent avec les Scrum Masters colocalisés et ceux qui ne le sont pas).

13.1.3.9 Dépendances

Décrivées dans la section 9.4.3.3.

13.1.3.10 Calendrier d'environnements

Le calendrier d'environnements est créé lors de la réunion de planification des environnements et est utilisé afin de coordonner les activités du processus de *déroulement et coordination des sprints* (13.2). Il définit les temps de test pour chaque équipe.

13.2 Déroulement et coordination des sprints

La figure 13-5 illustre les contributions, outils et résultats du processus de *déroulement et coordination des sprints*.

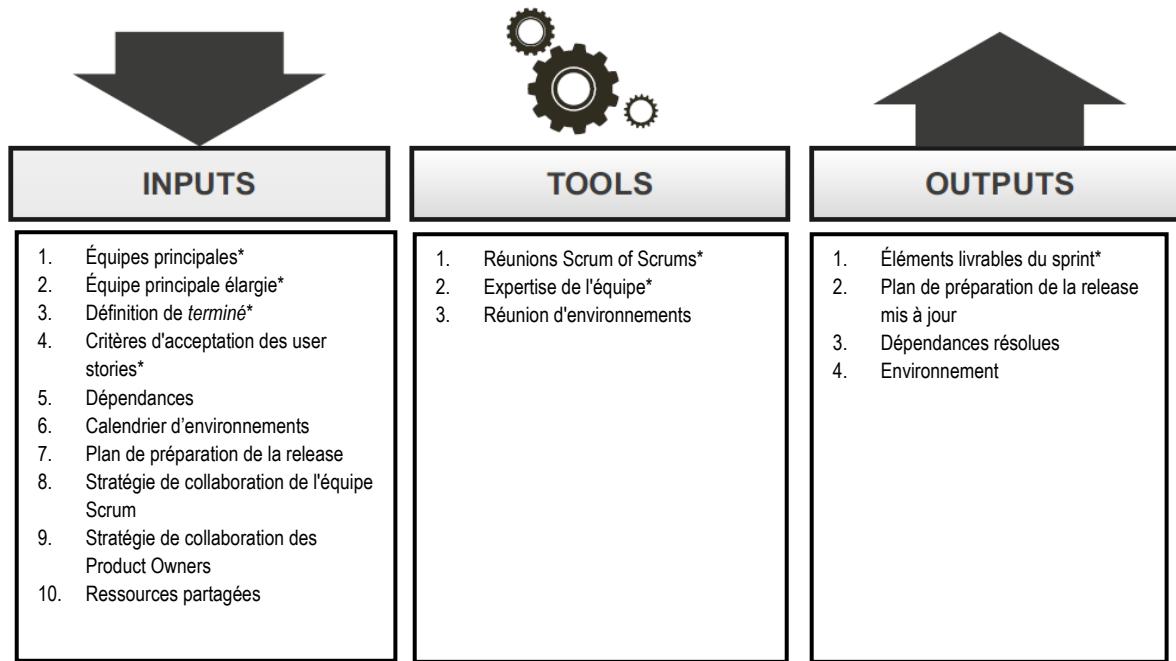


Figure 13-5 : Déroulement et coordination des sprints — Contributions, outils et résultats

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

La figure 13-6 illustre la relation du processus de *déroulement et coordination des sprints* avec les processus Scrum fondamentaux.

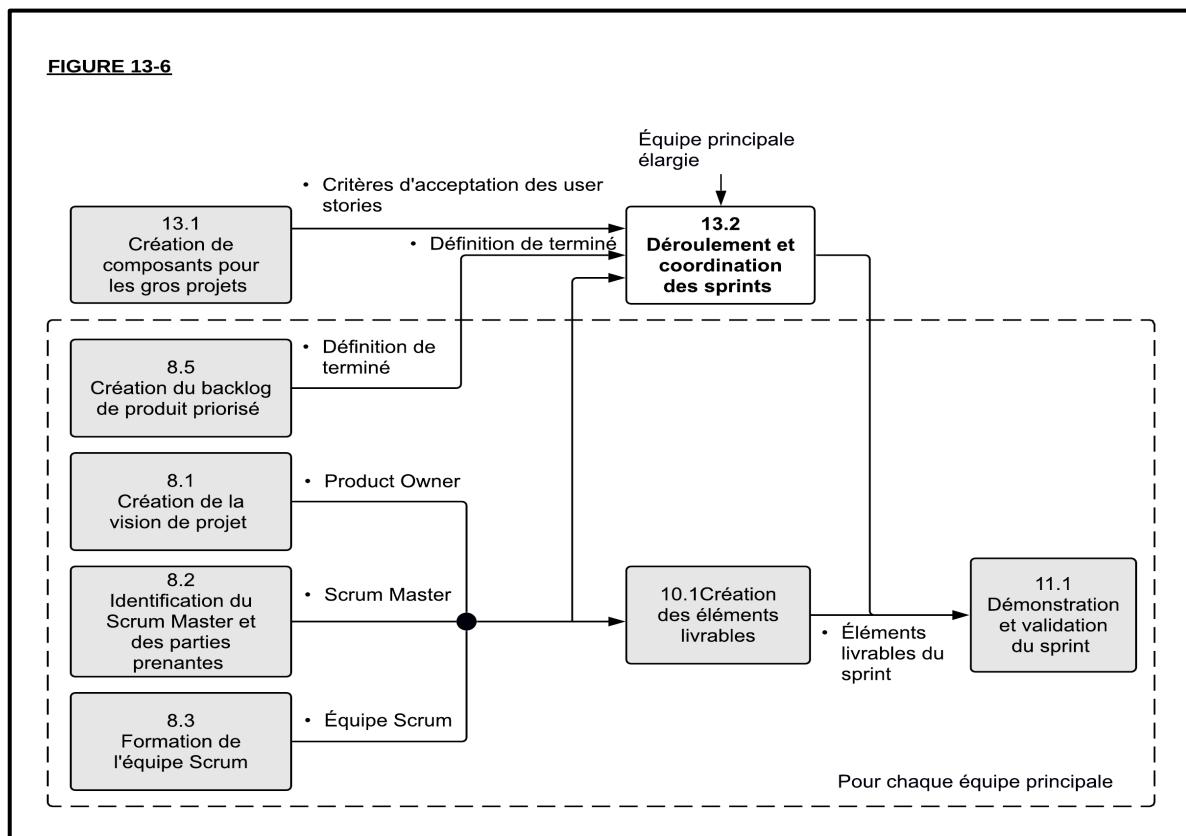


Figure 13-6 : Déroulement et coordination des sprints — Diagramme de flux de données

13.2.1 Contributions

13.2.1.1 Équipes principales*

Les équipes principales composées du Scrum Master, du Product Owner et de l'équipe Scrum sont décrites dans les sections 8.4.1.1 et 3.3.1.

13.2.1.2 Équipe principale élargie*

L'équipe principale élargie comprend le Chief Product Owner, Chief Scrum Master, les Scrum Masters, les Product Owners, et des membres spécifiques des équipes Scrum travaillant sur un gros projet. Le fait d'avoir une équipe principale élargie comprenant tous les membres de toutes les équipes ne serait pas

pratique. Chaque équipe doit donc choisir un membre qui la représentera au sein de l'équipe principale élargie.

Le Chief Scrum Master supprime les obstacles et garantit un environnement de projet productif pour toutes les équipes Scrum impliquées dans le projet. 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. Les Product Owners ont la responsabilité de décider du niveau de priorité des fonctionnalités et composants à développer par les équipes Scrum qui font partie du gros projet.

Le Chief Scrum Master, le Chief Product Owner, les Scrum Masters, les Product Owners et les membres spécifiques des équipes Scrum collaborent afin de développer une liste des ressources et composants communs nécessaires à toutes les équipes tout au long du projet.

13.2.1.3 Définition de *terminé**

Décrise dans les sections 8.5.3.2 et 5.4.3.

S'il y a un sprint de préparation de la release, sa définition de *terminé* est normalement unique et différente de la définition de *terminé* utilisée pour les autres sprints. Les définitions de *terminé* sont établies dans le but de garantir que les incrément de produit d'un sprint sont « potentiellement livrables ». Le sprint de préparation de la release résout tous les problèmes qui n'ont pas été abordés lors des sprints normaux en fonction des décisions commerciales délibérées et sont notées dans le plan de préparation de la release.

13.2.1.4 Critères d'acceptation des user stories*

Décris dans la section 9.1.3.2.

13.2.1.5 Dépendances

Décris dans les sections 9.4.2.3 et 9.4.3.3.

13.2.1.6 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.

13.2.1.7 Plan de préparation de la release

Décrit dans la section 13.1.3.1.

13.2.1.8 Stratégie de collaboration de l'équipe Scrum

Décrise dans la section 13.1.3.8.

13.2.1.9 Stratégie de collaboration des Product Owners

Décrise dans la section 13.1.3.7.

13.2.1.10 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. Voir section 13.1.3.4.

13.2.2 Outils

13.2.2.1 Réunion Scrum of Scrums*

Une réunion Scrum of Scrums est un élément important du passage à l'échelle de Scrum pour les gros projets. Généralement, un représentant de chaque équipe Scrum participe à la réunion — normalement le Scrum Master — mais il est également courant que d'autres membres d'une équipe Scrum participent à la réunion si cela est nécessaire. Normalement, cette réunion est facilitée par le Chief Scrum Master et est axée sur les points de coordination et d'intégration entre les différentes équipes Scrum.

Ces réunions sont préféablement courtes et les représentants de chaque équipe Scrum se rencontrent afin de partager le statut de leur équipe respective. Ces réunions ne sont normalement pas limitées dans le temps afin de permettre de partager plus d'informations entre les équipes. La réunion Scrum of Scrums est organisée à des intervalles prédéfinis ou lorsque les équipes Scrum en ont le besoin, de façon à faciliter le partage d'informations entre les différentes équipes Scrum. Les problèmes, dépendances et les risques qui impactent plusieurs équipes Scrum peuvent être surveillés de près ce qui permet aux différentes équipes travaillant sur un gros projet de mieux coordonner et intégrer leur travail. Le Chief Scrum Master (ou un

autre Scrum Master qui facilite les réunions SoS) est responsable de garantir que tous les représentants disposent d'un environnement conducteur au partage ouvert et honnête d'informations, notamment concernant le feedback fournit aux représentants des autres équipes. Pour les projets plus larges, impliquant un nombre important d'équipes, il est possible d'organiser ces réunions à plusieurs niveaux afin de partager le statut des équipes respectives.

A chacun leur tour, les représentants de chaque équipe Scrum offrent des mises à jour sur leur équipe. Ces mises à jour sont normalement offertes sous la forme de réponses 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 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 les informations qui permettent à chaque équipe de comprendre clairement le statut du travail de toutes les autres équipes.

13.2.2.2 Expertise de l'équipe*

Décrive dans la section 10.1.2.1.

En outre, pour les gros projets elle inclut également les connaissances du Chief Product Owner et du Chief Scrum Master en matière d'expertise de l'équipe. Cela contribue à évaluer les options disponibles pour développer les user stories de chaque sprint.

13.2.2.3 Réunion d'environnements

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.

13.2.3 Résultats

13.2.3.1 Éléments livrables du sprint*

Les incrémentums du sprint sont des produits potentiellement livrables développés par les équipes Scrum à la fin de chaque sprint. Les éléments livrables doivent posséder toutes les fonctionnalités définies dans les user stories incluses dans les sprints et doivent avoir été testés avec succès.

13.2.3.2 Plan de préparation de la release mis à jour

Le plan de préparation de la release détaille les étapes à suivre par chaque équipe Scrum et par le projet dans son ensemble afin de confirmer que les exigences minimales pour la release ont été atteintes. Il peut être mis à jour avec les changements identifiés lors de ce processus.

13.2.3.3 Dépendances résolues

Les dépendances entre les user stories, les tâches et les ressources requises afin de développer les user stories en incrément de produits exigent que les gros projets disposent d'un plan d'action pour les dépendances. Ce plan recouvre les actions nécessaires afin de gérer tous les types de dépendances — obligatoires, facultatives, externes et internes.

13.2.3.4 Environnements

Fait référence à l'identification et à la documentation de tous les environnements nécessaires pour développer et tester les produits du projet.

13

13.3 Préparation de la release d'un gros projet

La figure 13-7 illustre les contributions, outils et résultats du processus de *préparation de la release d'un gros projet*.



Figure 13-7 : Préparation de la release d'un gros projet — Contributions, outils et résultats

La figure 13-8 illustre la relation du processus de préparation de la release d'un gros projet avec les processus Scrum fondamentaux.

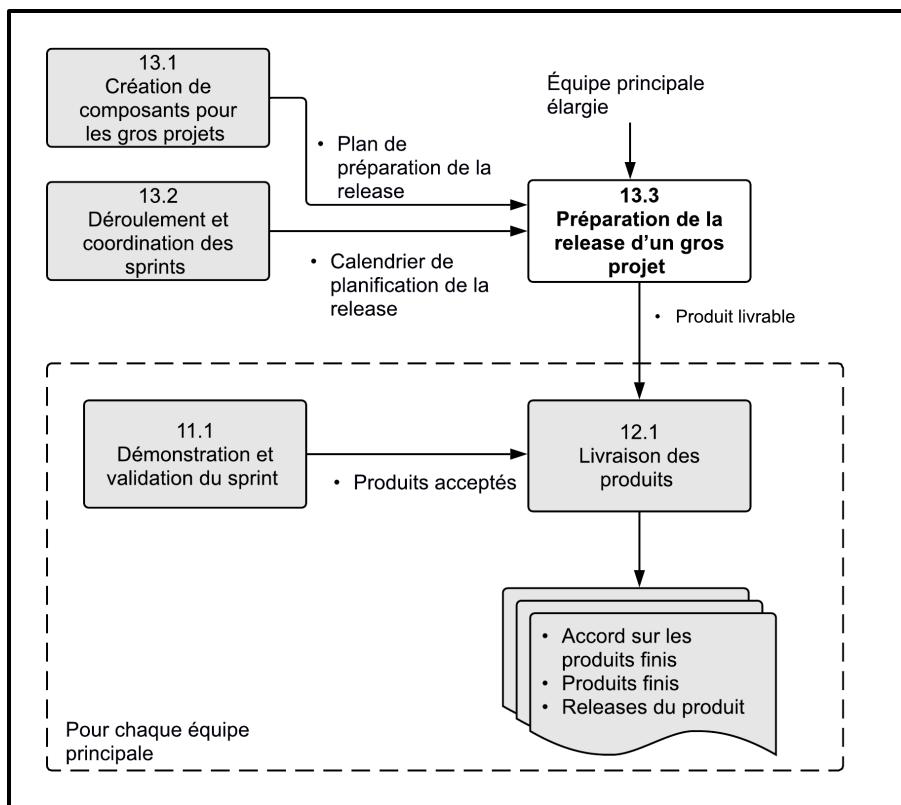


Figure 13-8 : Préparation de la release d'un gros projet — diagramme de flux de données

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

13.3.1 Contributions

13.3.1.1 Équipes principales*

Décrivées dans les sections 8.4.1.1 et 3.3.1.

13.3.1.2 Équipe principale élargie*

Décrivée dans la section 13.2.1.2.

13.3.1.3 Calendrier de planification de la release*

Décrit dans la section 8.6.3.1.

13.3.1.4 Plan de préparation de la release*

Décrit dans la section 13.1.3.1.

13.3.2 Outils

13.3.2.1 Stratégie de communication*

Décrise dans les sections 12.1.2.2 et 13.1.2.2.

13.3.2.2 Sprint de préparation de la 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. 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. Dans un sprint de préparation de la release aucune user story du backlog de produit priorisé n'est développée. Au lieu de ça, les tâches identifiées dans le plan de préparation de la release (voir 13.1.3.1) sont réalisées.

Il est important de noter que les tâches réalisées dans un sprint de préparation de la release découlent d'une décision commerciale. Cela ne change en rien le mandat qui veut que les critères d'acceptation et la définition de *terminé* soient respectés et atteints à la fin de chaque sprint.

Un sprint de préparation de la release **n'est pas** obligatoire. Bien qu'il ne soit pas obligatoire, dans de nombreux gros projets, la décision commerciale est prise d'organiser un sprint de préparation de la release.

13

13.3.2.3 Méthodes de préparation de la release

Les méthodes de préparation de la release sont les méthodes utilisées afin d'exécuter les tâches identifiées dans le plan de préparation de la release de façon à ce que les incrément de produit soient prêts à être expédiés/mis à disposition. Ces méthodes peuvent être spécifiques au projet mais sont plus probablement valides au niveau du portefeuille ou, au moins, du programme. Il est possible que celles-ci soient définies par le Scrum Guidance Body.

13.3.3 Résultats

13.3.3.1 Produit livrable*

Un produit livrable est un élément ou un incrément de produit qui remplit les critères d'acceptation définis par le client et le Product Owner. Il sera prêt à être livré ou mis à disposition à la fin du sprint de préparation de la release.

13.3.3.2 Notes de release

Les notes de release sont les documents fournis au client avec le produit de la release. Elles incluent les critères d'expédition externes ou relatifs au marché pour le produit à livrer.

13.3.3.3 Environnement de release

L'environnement de release est nécessaire afin de supporter la release dans/sur le terrain.

13.3.3.4 Améliorations recommandées du Scrum Guidance Body

Décrivées dans la section 13.1.3.6.

13.4 Impact des gros projets sur les processus Scrum fondamentaux

Bien que les processus Scrum fondamentaux décrits dans les chapitres 8 à 12 demeurent valides pour les gros projets, certains impacts spécifiques anticipés doivent être pris en compte. Le tableau 13-1 résume les impacts d'un gros projet sur les processus Scrum fondamentaux.

Processus	Description des impacts d'un gros projet sur le processus
8.1 Création de la vision de projet	<p>Résultat supplémentaire : Chief Product Owner identifié* Pour un gros projet un résultat de ce processus est l'identification du Chief Product Owner. Plusieurs Product Owners peuvent également être identifiés, tandis que les petits projets ne nécessitent qu'un seul Product Owner.</p> <p>Résultat supplémentaire : Chief Scrum Master identifié* De la même façon que pour le Chief Product Owner, le Chief Scrum Master doit également être identifié dans le cas d'un gros projet.</p>

Processus	Description des impacts d'un gros projet sur le processus
8.2 Identification du Scrum Master et des parties prenantes	<p>Résultat : Parties prenantes identifiées</p> <p>En plus d'identifier les parties prenantes du projet durant ce processus, pour un gros projet certains membres clés des différentes équipes Scrum doivent également être identifiés comme parties prenantes clés. Ces parties prenantes contribuent au processus de <i>création de composants pour les gros projets</i>.</p>
8.3 Formation de l'équipe Scrum	<p>Contribution supplémentaire : Chief Product Owner</p> <p>Décrit dans la section 3.4.2. Pour un gros projet, le Chief Product Owner est impliqué dans la formation des équipes Scrum et contribue au choix des membres des équipes. Le Chief Product Owner sert les intérêts du gros projet dans son ensemble tandis que les Product Owners se concentrent sur les équipes individuelles.</p> <p>Contribution supplémentaire : Chief Scrum Master</p> <p>Décrit dans la section 3.5.1. Pour un gros projet, le Chief Scrum Master est impliqué dans la formation des équipes Scrum et contribue au choix des membres des équipes. Le Chief Scrum Master sert les intérêts du gros projet dans son ensemble tandis que les Scrum Masters se concentrent sur les équipes individuelles.</p> <p>Contribution supplémentaire : Ressources partagées</p> <p>Décrivées dans la section 13.1.3.4. La connaissance de toutes les ressources partagées disponibles pour les équipes Scrum est une contribution nécessaire à la formation de chaque équipe Scrum.</p> <p>Contribution supplémentaire : Spécialisation de l'équipe</p> <p>Décrive dans la section 13.1.3.5.</p>
8.4 Développement des épiccs	<p>Contribution supplémentaire : Stratégie de collaboration des Product Owners</p> <p>Décrise dans la section 13.1.3.7. C'est un des principaux résultats du processus de <i>création de composants pour les gros projets</i>. Elle définit la façon dont plusieurs Product Owners travaillent ensemble avec le Chief Product Owner. Elle aborde la façon dont ils travaillent avec les parties prenantes afin de collecter les exigences, mettent à jour le backlog de produit priorisé et travaillent avec plusieurs équipes Scrum. Seul un Product Owner interagit directement avec chaque équipe Scrum. Cependant, des décisions doivent être prises concernant la façon dont les équipes Scrum sont attribuées aux différents Product Owners et le nombre d'équipes Scrum avec lesquelles chaque Product Owner travaille.</p>

Processus	Description des impacts d'un gros projet sur le processus
8.5 Création du backlog de produit priorisé	<p>Contribution supplémentaire : Stratégie de collaboration des Product Owners Décrise dans la section 13.1.3.7. Étant donné que la stratégie de collaboration des Product Owners définit la façon dont les Product Owners mettent à jour le backlog de produit priorisé, cela est une contribution importante dans ce processus.</p> <p>Contribution supplémentaire : Dépendances Décrites dans la section 9.4.3.3.</p>
8.6 Planification de la release	<p>Contribution supplémentaire : Chief Scrum Master Décrit dans la section 3.5.1.</p> <p>Contribution supplémentaire : Plan de préparation de la release Décrit dans la section 13.1.3.1. Le plan de préparation de la release comporte des informations importantes qui impactent la planification globale de la release.</p>
9.1 Création des user stories	<p>Contribution supplémentaire : Stratégie de collaboration des Product Owners Décrise dans la section 13.1.3.7.</p> <p>Contribution supplémentaire : Stratégie de collaboration de l'équipe Scrum Décrise dans la section 13.1.3.8.</p>
9.2 Estimation des user stories	<p>Contribution supplémentaire : Stratégie de collaboration des Product Owners Décrise dans la section 13.1.3.7.</p> <p>Contribution supplémentaire : Stratégie de collaboration de l'équipe Scrum Décrise dans la section 13.1.3.8.</p>
9.3 Validation des user stories	Pas de changement
9.4 Identification des tâches	Pas de changement
9.5 Estimation des tâches	Pas de changement
9.6 Création du backlog de sprint	Pas de changement
10.1 Création des éléments livrables	Pas de changement
10.2 Mélées quotidiennes	Pas de changement

Processus	Description des impacts d'un gros projet sur le processus
10.3 Affinage du backlog de produit priorisé	<p>Contribution supplémentaire : Stratégie de collaboration des Product Owners Décrise dans la section 13.1.3.7. La stratégie de collaboration des Product Owners définit comment plusieurs Product Owners mettent à jour le backlog de produit priorisé.</p> <p>Contribution supplémentaire : Stratégie de collaboration de l'équipe Scrum Décrise dans la section 13.1.3.8. La stratégie de collaboration de l'équipe Scrum définit comment les équipes participent à l'affinage du backlog de produit priorisé. Cette stratégie définit également les représentants d'équipe impliqués dans le processus d'affinage et la manière dont ils sont sélectionnés.</p> <p>Contribution supplémentaire : Spécialisation de l'équipe Décrise dans la section 13.1.3.5.</p> <p>Résultat supplémentaire : Plan de préparation de la release mis à jour Les changements apportés au backlog durant l'affinage du backlog de produit peuvent avoir un impact sur la plan de préparation de la release (voir section 13.1.3.1).</p>
11.1 Démonstration et validation du sprint	Ce processus est réalisé individuellement par chaque équipe Scrum. Le Product Owner respectif de chaque équipe approuve les user stories. Cependant, ce processus peut être un peu complexe en raison des interdépendances. Il se peut que le Product Owner participe à toutes les réunions de revue de sprint pour chaque équipe. Une validation de bout en bout peut souvent être nécessaire. En effet, certains éléments peuvent sembler <i>terminés</i> lorsqu'ils sont validés au sein et par un équipe individuelle, mais peuvent ne pas respecter les critères d'acceptation finaux lorsqu'ils sont évalués lors de la validation de bout en bout.

Processus	Description des impacts d'un gros projet sur le processus
11.2 Rétrospective du sprint	<p>Contribution supplémentaire : Stratégie de collaboration des Product Owners Décrise dans la section 13.1.3.7. L'affinage du backlog de produit peut être particulièrement difficile pour les gros projet. S'il n'est pas fait efficacement, l'affinage peut causer des problèmes et gaspiller les efforts entre les équipes. Il est donc recommandé de discuter de l'affinage durant la rétrospective, en se concentrant spécifiquement sur la façon dont les nombreux Product Owners interagissent entre eux et avec les équipes Scrum afin d'affiner le backlog de façon efficace.</p> <p>Contribution supplémentaire : Stratégie de collaboration de l'équipe Scrum Décrise dans la section 13.1.3.8. L'affinage du backlog de produit peut être particulièrement difficile pour les gros projet. S'il n'est pas fait efficacement, l'affinage peut créer des problèmes et gaspiller les efforts entre les équipes. Il est donc recommandé de discuter de l'affinage durant la rétrospective, en se concentrant spécifiquement sur l'interopérabilité entre les différentes équipes Scrum et la façon dont elles interagissent avec les Product Owners lors des activités d'affinage.</p>
12.1 Livraison des produits	<p>Contribution supplémentaire : Chief Product Owner Décrit dans la section 3.4.2.</p> <p>Contribution supplémentaire : Chief Scrum Master Décrit dans la section 3.5.1.</p> <p>Contribution supplémentaire : Notes de release Décris dans la section 13.3.3.2.</p> <p>Pour les gros projets, les notes de release issues du processus de <i>préparation de la release d'un gros projet</i> deviennent une contribution dans ce processus.</p>
12.2 Rétrospective du projet	<p>Contribution supplémentaire : Chief Product Owner Décrit dans la section 3.4.2.</p> <p>Contribution supplémentaire : Chief Scrum Master Décrit dans la section 3.5.1.</p>

Tableau 13-1 : Résumé de l'impact des gros projets sur les processus Scrum fondamentaux

14. SCRUM POUR L'ENTREPRISE

Ce chapitre détaille les processus relatifs au passage à l'échelle de Scrum pour l'entreprise : *Création de composants pour les programmes ou les portefeuilles, revue et mise à jour du Scrum Guidance Body, création et affinage du backlog de programme ou de portefeuille, coordination des composants pour les programmes ou les portefeuilles, et rétrospective de release de programme ou de portefeuille.*

Tel qu'il est décrit dans *A Guide to the Scrum Body of Knowledge (Guide SBOK™)*, le passage à l'échelle de Scrum pour l'entreprise 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 autres résultats destinés à être livrés aux parties prenantes
- 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 les personnes qui découvrent l'approche et les processus Scrum se concentrent principalement sur les contributions, outils et résultats obligatoires. En revanche, les autres praticiens Scrum expérimentés peuvent aspirer à améliorer leurs connaissances grâce à l'ensemble des informations contenues dans ce chapitre.

La figure 14-1 offre un aperçu des processus relatifs au passage à l'échelle de Scrum pour l'entreprise, qui sont les suivants :

14.1 Création de composants pour les programmes ou les portefeuilles — le Product Owner du programme ou du portefeuille et les parties prenantes principales identifient les ressources et composants communs requis pour le programme ou le portefeuille. La définition minimum de *terminé* est établie et toutes les parties prenantes sont identifiées.

14.2 Revue et mise à jour du Scrum Guidance Body — les recommandations du Scrum Guidance Body sont régulièrement passées en revue par les membres du Scrum Guidance Body et celles-ci sont mises à jour le cas échéant. Le changement des membres du Scrum Guidance Body est également abordé durant ce processus.

14.3 Création et affinage du backlog de programme ou de portefeuille — le backlog de programme ou de portefeuille est créé, mis à jour et entretenu. Il est possible de formuler des suggestions pour l'amélioration des recommandations du Scrum Guidance Body et d'ajuster les échéances de mise en œuvre en fonction du changement des exigences et/ou des progrès des projets du programme ou du portefeuille.

14.4 Coordination des composants pour les programmes ou les portefeuilles — les composants du programme ou du portefeuille sont coordonnés. Les dépendances entre les projets sont abordées, les obstacles communs sont discutés et les meilleures pratiques sont partagées. Des recommandations pour les améliorations du Scrum Guidance Body sont parfois formulées.

14.5 Rétrospective de release de programme ou de portefeuille — le Product Owner du programme ou du portefeuille et les parties prenantes principales se réunissent afin de faire la rétrospective de la release d'un programme ou d'un portefeuille et d'en assimiler les enseignements. Généralement, ces enseignements mènent à la mise en œuvre des améliorations actionnables convenues lors des futures releases. Il est parfois possible de formuler des recommandations pour l'amélioration du Scrum Guidance Body.

<p>14.1 Création de composants pour les programmes ou les portefeuilles</p> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Vision et mission de l'entreprise* 2. Product Owner du portefeuille* 3. Scrum Master du portefeuille* 4. Product Owner du programme* 5. Scrum Master du programme* 6. Matrice d'organisation des ressources 7. Recommandations du Scrum Guidance Body 8. Parties prenantes clés <p>OUTILS</p> <ol style="list-style-type: none"> 1. Stratégies de communication* 2. Stratégies de ressources humaines de l'entreprise* 3. Analyse des parties prenantes <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Définition minimum de <i>terminé</i>* 2. Critères d'acceptation des user stories* 3. Ressources partagées* 4. Parties prenantes identifiées* 5. Améliorations recommandées du Scrum Guidance Body 	<p>14.2 Revue et mise à jour du Scrum Guidance Body</p> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Législations* 2. Améliorations recommandées du Scrum Guidance Body* 3. Membres du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Critères de sélection des membres* 2. Benchmarking 3. Réunions du Scrum Guidance Body <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Recommandations mises à jour du Scrum Guidance Body* 2. Actionable Escalations 3. Liste mise à jour des membres du Scrum Guidance Body 4. Mises à jour rejetées des recommandations du Scrum Guidance Body 	<p>14.3 Création et affinage du backlog de programme ou de portefeuille</p> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Vision et mission de l'entreprise* 2. Backlog de portefeuille priorisé* 3. Backlog de programme priorisé* 4. Product Owner du portefeuille* 5. Scrum Master du portefeuille* 6. Product Owner du programme* 7. Scrum Master du programme* 8. Recommandations du Scrum Guidance Body 9. Politiques de l'entreprise 10. Normes du secteur 11. Résultats des évaluations/du benchmarking <p>OUTILS</p> <ol style="list-style-type: none"> 1. Réunions de revue du backlog de programme ou de portefeuille priorisé* 2. Techniques de communication* 3. Méthodes de priorisation des user stories 4. Atelier d'user stories 5. Entretiens d'utilisateurs ou de clients 6. Questionnaires <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Backlog de programme ou de portefeuille priorisé mis à jour* 2. Améliorations recommandées du Scrum Guidance Body* 3. Échéances mises à jour d'implémentation des projets* 4. Personnages 5. Risques identifiés
<p>14.4 Coordination des composants pour les programmes ou les portefeuilles</p> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Définition de <i>terminé</i>* 2. Dépendances connues* 3. Backlog de programme ou de portefeuille priorisé* 4. Product Owner du portefeuille* 5. Scrum Master du portefeuille* 6. Product Owner du programme* 7. Scrum Master du programme* 8. Produits du projet potentiellement livrables 9. Registres des obstacles 10. Backlogs de produit priorisés 11. Enseignements tirés par l'équipe Scrum 12. Calendriers de planification de la release <p>OUTILS</p> <ol style="list-style-type: none"> 1. Réunion Scrum of Scrums (SoS)* 2. Réunion Scrum of Scrum of Scrums (SoSoS) 3. Techniques de communication <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Registres des obstacles mis à jour* 2. Dépendances mises à jour* 3. Améliorations recommandées du Scrum Guidance Body 	<p>14.5 Rétrospective des releases de programme ou de portefeuille</p> <p>CONTRIBUTIONS</p> <ol style="list-style-type: none"> 1. Product Owner du portefeuille* 2. Scrum Master du portefeuille* 3. Product Owner du programme* 4. Scrum Master du programme* 5. Parties prenantes 6. Recommandations du Scrum Guidance Body <p>OUTILS</p> <ol style="list-style-type: none"> 1. Réunion de rétrospective de programme ou de portefeuille* 2. Expertise du Scrum Guidance Body <p>RÉSULTATS</p> <ol style="list-style-type: none"> 1. Améliorations actionnables convenues* 2. Mesures de suivi et dates d'échéances* 3. Améliorations recommandées du Scrum Guidance Body 	

Figure 14-1 : Aperçu du passage à l'échelle de Scrum pour l'entreprise

La figure 14-2 ci-dessous illustre les contributions, outils et résultats obligatoires pour les processus de passage à l'échelle de Scrum pour l'entreprise.

<p>14.1 Création de composants pour les programmes ou les portefeuilles</p> <p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Vision et mission de l'entreprise* 2. Product Owner du portefeuille* 3. Scrum Master du portefeuille* 4. Product Owner du programme* 5. Scrum Master du programme* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Stratégies de communication* 2. Stratégies de ressources humaines de l'entreprise* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Définition minimum de <i>terminé</i>* 2. Critères d'acceptation des user stories* 3. Ressources partagées* 4. Parties prenantes identifiées* 	<p>14.2 Revue et mise à jour du Scrum Guidance Body</p> <p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Législations* 2. Améliorations recommandées du Scrum Guidance Body* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Critères de sélection des membres* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Recommandations mises à jour du Scrum Guidance Body* 	<p>14.3 Création et affinage du backlog de programme ou de portefeuille</p> <p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Vision et mission de l'entreprise* 2. Backlog de portefeuille priorisé* 3. Backlog de programme priorisé* 4. Scrum Master du portefeuille* 5. Product Owner du portefeuille* 6. Product Owner du programme* 7. Scrum Master du programme* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunions de revue du backlog de programme ou de portefeuille priorisé* 2. Techniques de communication* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Backlog de programme ou de portefeuille mis à jour* 2. Recommandations mises à jour du Scrum Guidance Body* 3. Échéances mises à jour
<p>14.4 Coordination des composants pour les programmes ou les portefeuilles</p> <p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Définition de <i>terminé</i>* 2. Dépendances connues* 3. Backlog de programme ou de portefeuille priorisé* 4. Product Owner du portefeuille* 5. Scrum Master du portefeuille* 6. Product Owner du programme* 7. Scrum Master du programme* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunion Scrum of Scrums (SoS)* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Registres des obstacles mis à jour* 2. Dépendances mises à jour* 	<p>14.5 Rétrospective des releases de programme ou de portefeuille</p> <p>CONTRIBUTIONS</p> <ul style="list-style-type: none"> 1. Product Owner du portefeuille* 2. Product Owner du programme* 3. Scrum Master du portefeuille* 4. Scrum Master du programme* <p>OUTILS</p> <ul style="list-style-type: none"> 1. Réunion de rétrospective de programme ou de portefeuille* <p>RÉSULTATS</p> <ul style="list-style-type: none"> 1. Améliorations actionnables convenues* 2. Mesures de suivi et dates d'échéances* 	

Figure 14-2 : Passage à l'échelle de Scrum pour l'entreprise (éléments essentiels)

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

14.1 Crédit de composants pour les programmes ou les portefeuilles

La figure 14-3 illustre les contributions, outils et résultats du processus de *création de composants pour les programmes ou les portefeuilles*.

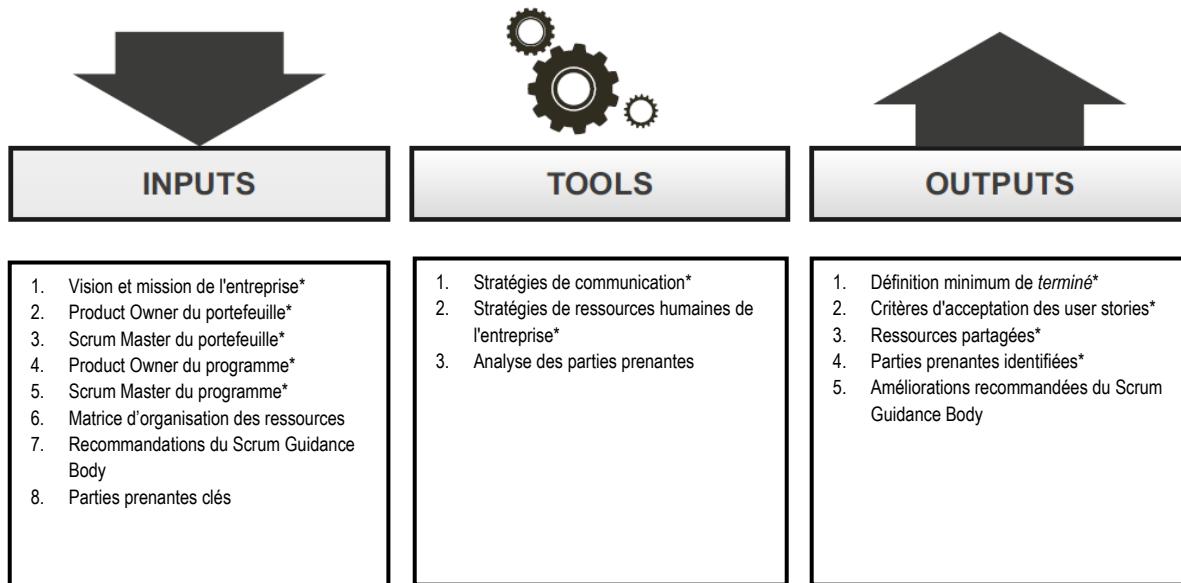


Figure 14-3 : Crédit de composants pour les programmes ou les portefeuilles — Contributions, outils et résultats

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

14

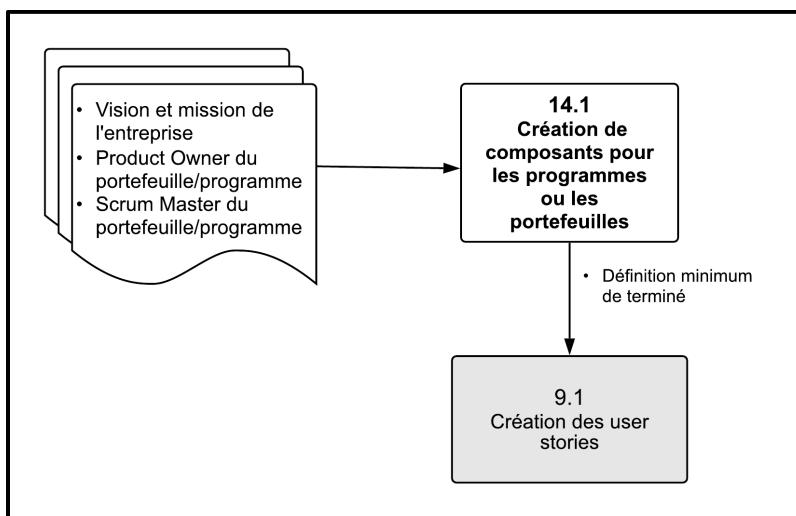


Figure 14-4 : Crédit de composants pour les programmes ou les portefeuilles — diagramme de flux de données

14.1.1 Contributions

14.1.1.1 Vision et mission de l'entreprise*

Les énoncés de vision et de mission de l'entreprise sont décrits respectivement dans les sections 8.1.1.8 et 8.1.1.9.

La vision et la mission de l'entreprise sont importantes quel que soit le projet mais elles le sont d'autant plus au niveau des programmes et plus particulièrement encore au niveau des portefeuilles. Les programmes et les portefeuilles doivent être motivés par la vision et la mission globales de l'entreprise. Cela garantit l'unité des efforts au travers de l'organisation.

14.1.1.2 Product Owner du portefeuille*

Décrit dans la section 3.4.4.

14.1.1.3 Scrum Master du portefeuille*

Décrit dans la section 3.5.3.

14.1.1.4 Product Owner du programme*

Décrit dans la section 3.4.3.

Le Product Owner du programme est l'une des nombreuses parties prenantes d'un projet. Au niveau du programme, le Product Owner du programme joue un rôle similaire à celui du Product Owner dans un projet. Il est responsable de et gère la création et l'affinage des composants du programme.

14.1.1.5 Scrum Master du programme*

Décrit dans la section 3.5.2.

Le Scrum Master du programme est l'une des nombreuses parties prenantes d'un projet. Au niveau du programme, le Scrum Master du programme joue un rôle similaire à celui du Scrum Master dans un projet. Il est un facilitateur, résout les problèmes et supprime les obstacles au niveau du programme.

14.1.1.6 Matrice d'organisation des ressources

Décrise dans la section 8.2.1.8.

14.1.1.7 Recommandations du Scrum Guidance Body

Décrise dans la section 8.1.1.11.

Les recommandations du Scrum Guidance Body sont particulièrement importantes au niveau des programmes et des portefeuilles puisque les conseils adéquats sont nécessaires à un nombre potentiellement important de projets.

14.1.1.8 Parties prenantes clés

Bien que de nombreuses parties prenantes soient identifiées durant ce processus, plusieurs parties prenantes clés sont déjà connues. Par exemple, au niveau du portefeuille, les parties prenantes clés incluent des membres du conseil d'administration d'une entreprise et des organisations gouvernementales. Les parties prenantes clés au niveau du programme incluent les sponsors du programme ou des projets associés et les hauts dirigeants.

Voir également les sections associées 1.4.3.1, 3.3.2 et 6.4.2.1.

14

14.1.2 Outils

14.1.2.1 Stratégies de communication*

Décrise dans la section 12.1.2.2.

Les stratégies de communication définissent la façon dont les informations sont disséminées aux parties prenantes et au travers de l'organisation, du portefeuille et des programmes. Elles établissent également comment et quand communiquer ainsi que les modes de communication à utiliser. Le portefeuille fournit des conseils et contribue à la stratégie de communication pour les programmes au sein du portefeuille. De la même façon, le programme fournit des conseils et contribue à la stratégie de communication pour les projets au sein du programme.

14.1.2.2 Stratégies de ressources humaines de l'entreprise*

De façon générale, les stratégies de ressources humaines de l'entreprise fournissent des informations sur la disponibilité du personnel pour différents projets, programmes et portefeuilles. Elles fournissent aussi des informations relatives à l'embauche du personnel requis pour les efforts à venir.

14.1.2.3 Analyse des parties prenantes

Une analyse standard des parties prenantes permet d'identifier les parties prenantes au niveau du programme ou du portefeuille. Des détails supplémentaires sur les parties prenantes du programme ou du portefeuille peuvent être identifiés en tant que personnages lors du processus de *création et d'affinage du backlog de programme ou de portefeuille*.

14.1.3 Résultats

14.1.3.1 Définition minimum de terminé*

Décrise dans la section 5.4.3.

14.1.3.2 Critères d'acceptation des user stories*

Décris dans la section 9.1.3.2.

14.1.3.3 Ressources partagées*

Décris dans la section 13.1.3.4.

14.1.3.4 Parties prenantes identifiées*

Décris dans la section 8.2.3.2.

Les parties prenantes clés au niveau du portefeuille ou du programme contribuent à ce processus. Les parties prenantes supplémentaires sont identifiées durant ce processus.

14.1.3.5 Améliorations recommandées du Scrum Guidance Body

Décrivées dans la section 13.1.3.6.

Des suggestions ou du feedback, résultant du processus de *création de composants pour les programmes ou les portefeuilles*, peuvent être formulés en vue d'apporter des améliorations potentielles au Scrum Guidance Body (SGB). Ces améliorations recommandées sont discutées et acceptées ou rejetées par le Scrum Guidance Body (voir section 14.2 processus de *revue et mise à jour du Scrum Guidance Body*). Si le SBG accepte ces suggestions, celles-ci sont ajoutées en tant que mises à jour dans la documentation du Scrum Guidance Body.

14.2 Revue et mise à jour du Scrum Guidance Body

La figure 14-5 illustre les contributions, outils et résultats du processus de *revue et mise à jour du Scrum Guidance Body*.

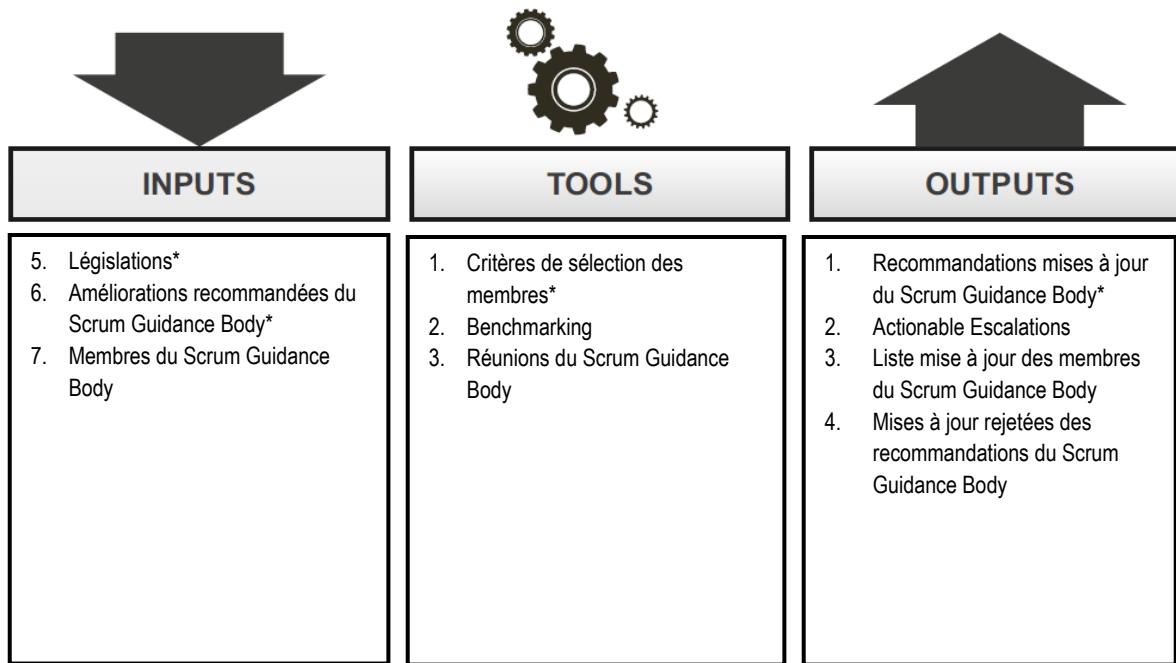


Figure 14-5 : Revue et mise à jour du Scrum Guidance Body — Contributions, outils et résultats

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

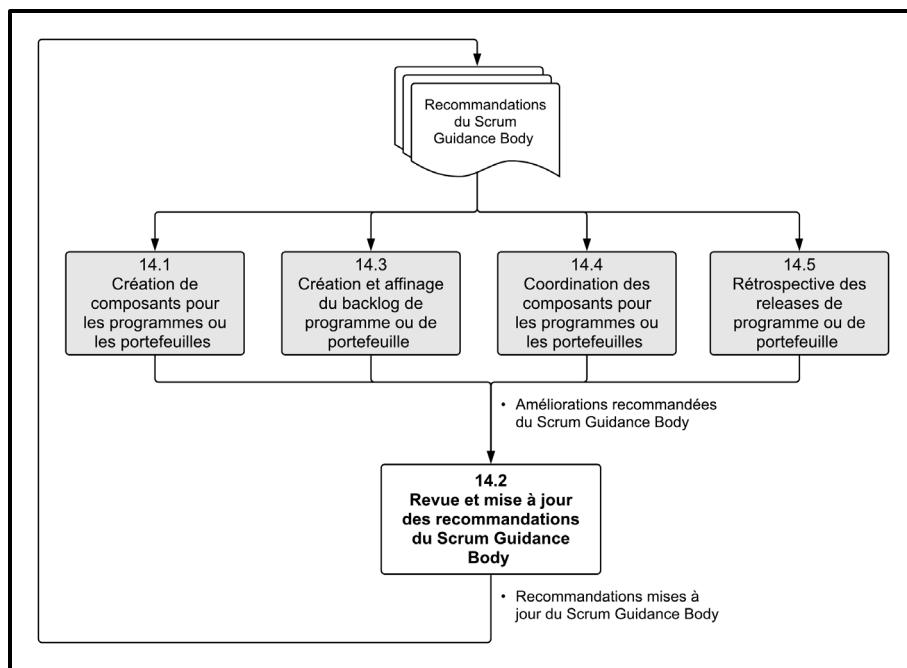


Figure 14-6 : Revue et mise à jour du Scrum Guidance Body — Diagramme de flux de données

14.2.1 Contributions

14.2.1.1 Législations*

Les législations incluent les législations fédérales, locales et étatiques ou les règlements de secteur auxquels le programme ou le portefeuille doivent adhérer. Les user stories créées afin de respecter les législations gouvernementales dans un délai défini sont ajoutées au backlog du programme ou du portefeuille.

Parfois, les recommandations du Scrum Guidance Body doivent être mises à jour afin de prendre en compte de nouvelles législations.

14.2.1.2 Améliorations recommandées du Scrum Guidance Body*

Décrivées dans la section 13.1.3.6.

Après les rétrospectives et autres processus Scrum, des suggestions et feedbacks peuvent être formulés afin de réviser ou d'améliorer les recommandations du Scrum Guidance Body. Si le Scrum Guidance Body accepte ces suggestions ou feedbacks, ceux-ci seront ajoutés en tant que mises à jour aux recommandations du Scrum Guidance Body.

14.2.1.3 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.

14.2.2 Outils

14.2.2.1 Critères de sélection des membres*

Les critères de sélection des membres sont déterminés par les parties prenantes pour définir les membres du Scrum Guidance Body, leurs rôles et responsabilités, le nombre de membres ainsi que l'expertise et les compétences requises.

Chaque entreprise possède ses propres critères de sélection pour les membres du Scrum Guidance Body. Cependant, il est recommandé que chaque membre dispose d'un certain degré d'expertise Scrum et de limiter le nombre de membres du SGB afin de garantir que celui-ci demeure adéquat et ne devient pas normatif.

14

14.2.2.2 Benchmarking

Une entreprise se doit de régulièrement comparer ses propres pratiques à celles des entreprises concurrentes (benchmarking) 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.

14.2.2.3 Réunions du Scrum Guidance Body

Le Scrum Guidance Body se réunit régulièrement afin d'aborder le besoin potentiel de 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.

14.2.3 Résultats

14.2.3.1 Recommandations mises à jour du Scrum Guidance Body*

Décrivées dans la section 11.2.3.6.

La revue du Scrum Guidance Body peut résulter en des changements nécessaires et conduire à la mise à jour des recommandations du Scrum Guidance Body.

14.2.3.2 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 accélérer l'approbation d'un changement de politique.

14.2.3.3 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.

14.2.3.4 Mises à jour rejetées des recommandations du Scrum Guidance Body

Les améliorations recommandées du Scrum Guidance Body ne sont pas toujours acceptées. Si l'amélioration recommandée n'est pas acceptée par les membres du Scrum Guidance Body, le feedback expliquant la raison du rejet est fourni aux parties concernées.

14.3 Cration et affinage du backlog de programme ou de portefeuille

La figure 14-7 illustre les contributions, outils et résultats du processus de *création et d'affinage du backlog de programme ou de portefeuille*.

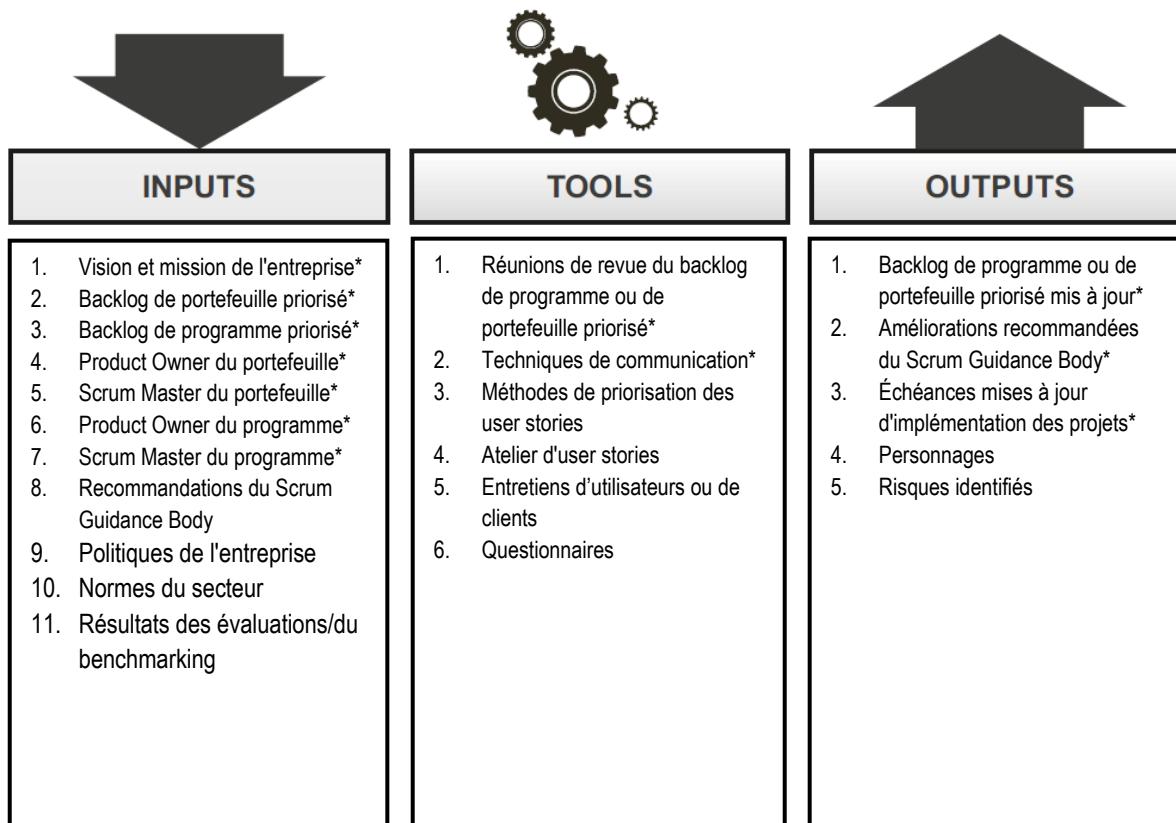


Figure 14-7 : Crédit et affinage du backlog de programme ou de portefeuille — Contributions, outils et résultats

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

La figure 14-8 illustre la relation entre le processus de *création et d'affinage du backlog de programme ou de portefeuille* et les processus Scrum fondamentaux.

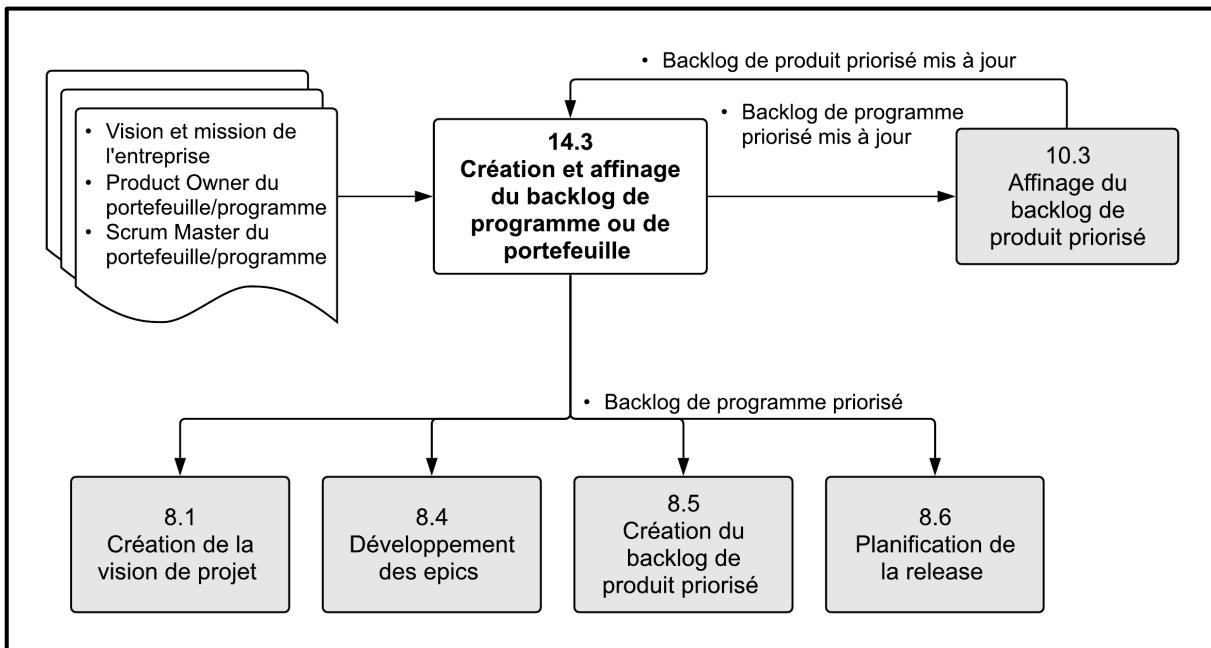


Figure 14-8 : Crédit et affinage du backlog de programme ou de portefeuille — Diagramme de flux de données

14.3.1 Contributions

14.3.1.1 Vision et mission de l'entreprise*

Décrivées dans les sections 8.1.1.8 et 8.1.1.9.

14.3.1.2 Backlog de portefeuille priorisé*

Le backlog de portefeuille priorisé joue, au niveau du portefeuille, le même rôle que le backlog de programme priorisé au niveau du programme. Les éléments du backlog de portefeuille priorisé contribuent aux différents backlogs de programme priorisés et à terme aux backlogs de produit priorisés des projets individuels. Comme pour les backlogs de programme priorisés (voir la section 8.1.1.5), les user stories ne sont que peu ou pas affinées à ce niveau. L'affinage est effectué au sein même des projets au niveau de chaque backlog de produit priorisé spécifique.

14.3.1.3 Backlog de programme priorisé*

Décrit dans la section 8.1.1.5.

Le backlog de programme priorisé joue, au niveau du programme, un rôle très similaire à celui du backlog de produit priorisé au niveau du projet. Il identifie les exigences d'un programme et ses priorités.

Il existe cependant quelques différences :

La création des incrémentés respectifs et leur validation sont effectuées au sein des projets d'un programme. Les critères d'acceptation et la définition de *terminé* pour chaque élément/user story du backlog de produit peuvent être définis au niveau du programme. Les projets doivent respecter ces critères mais peuvent y ajouter leurs propres critères si cela s'avère nécessaire.

La durée d'un sprint est spécifique au projet et varie normalement entre les différents projets d'un programme. De plus, la vitesse varie d'équipe en équipe. Il est donc nécessaire d'avoir des user stories très granulaires au niveau du programme. L'affinage des user stories au niveau du programme permet uniquement de garantir que la story est clairement comprise et que des critères d'acceptation tangibles peuvent être définis au niveau du programme.

14.3.1.4 Product Owner du portefeuille*

14

Décrit dans la section 3.4.4.

Le rôle de Product Owner de portefeuille est similaire à celui du Product Owner d'un programme. Il est responsable de et encourage la création et l'affinage du backlog de produit du portefeuille.

14.3.1.5 Scrum Master du portefeuille*

Décrit dans la section 3.5.3.

Le Scrum Master de portefeuille est l'une des nombreuses parties prenantes d'un projet. Au niveau du portefeuille, le Scrum Master du portefeuille joue un rôle similaire à celui du Scrum Master d'un programme.

14.3.1.6 Product Owner du programme*

Décrit dans la section 3.4.3.

Le Product Owner de programme est l'une des nombreuses parties prenantes d'un projet. Au niveau du programme, le Product Owner du programme joue un rôle similaire à celui du Product Owner d'un projet. Il est responsable de et encourage la création et l'affinage du backlog de produit du programme.

14.3.1.7 Scrum Master du programme*

Décrit dans la section 3.5.2.

Le Scrum Master de programme est l'une des nombreuses parties prenantes d'un projet. Au niveau du programme, le Scrum Master du programme joue un rôle similaire à celui du Scrum Master d'un projet. Il est un facilitateur, résout les problèmes et supprime les obstacles au niveau du programme.

14.3.1.8 Recommandations du Scrum Guidance Body

Décris dans les sections 8.1.1.11 et 10.3.1.11.

Lors de la création et de l'affinage du backlog de programme ou de portefeuille priorisé, les recommandations du Scrum Guidance Body indiquent les meilleures pratiques à prendre en compte au niveau du programme ou du portefeuille.

14.3.1.9 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 existantes puisque celles-ci ont été créées suivant les politiques existantes.

14.3.1.10 Normes du secteur

Les nouvelles normes de secteur, ou la modifications de normes existantes doivent être mises en œuvre de façon à maintenir un produit ou service viable. Les user stories relatives au respect de ces normes doivent donc être incluses au backlog de programme et/ou de portefeuille et priorisées en conséquence.

Il peut arriver que les recommandations du Scrum Guidance Body doivent être modifiées afin de refléter les nouvelles normes ou celles qui ont été modifiées.

14.3.1.11 Résultats des évaluations/du benchmarking

Tout d'abord, les résultats des évaluations/du benchmarking nécessitent la mise à jour des meilleures pratiques recommandées par le Scrum Guidance Body. Ces résultats peuvent également contribuer à l'établissement d'une norme minimale lors de la conception d'un produit ou d'un service et conduire 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.

14.3.2 Outils

14.3.2.1 Réunion de revue du backlog de programme ou de portefeuille priorisé*

La participation aux réunions de revue du backlog de programme ou de portefeuille diffère de la participation aux réunions de revue au niveau du projet. Les équipes Scrum participent aux sessions d'affinage au niveau du projet. 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.

Voir également les sections associées 10.3.2.1 et 6.5.1.2.

14.3.2.2 Techniques de communication*

Décrivées dans la section 10.3.2.2.

14.3.2.3 Méthodes de priorisation des user stories

Décrivées dans la section 8.5.2.1.

Au niveau d'un programme ou d'un portefeuille, il y a normalement moins d'exigences/user stories qu'au niveau d'un projet. Le pourcentage d'user stories disposant d'une valeur, d'une exigence commerciale ou d'un impact utilisateur tangible important est normalement beaucoup plus bas qu'au niveau du projet. Par conséquent, un nombre limité de techniques seront pertinentes et utiles au niveau du programme ou du portefeuille.

Par exemple, l'analyse Kano a ses limites étant donné qu'il n'y aura pas d'éléments insatisfaisants ou attrayants. Sans un nombre important de parties prenantes, et plus particulièrement d'utilisateurs, la méthode des 100 points offre également une valeur limitée. La technique MoSCow présente aussi un certain nombre de limitations étant donné qu'il n'y aura pas de fonctionnalités qui « devraient être réalisées » ou « impossibles » au niveau d'un programme ou d'un portefeuille.

La comparaison par paire est une technique qui fonctionne bien au niveau d'un programme ou d'un portefeuille.

14.3.2.4 Atelier d'user stories

Décrit dans la section 8.4.2.2.

Comparés aux projets, les ateliers d'user stories pour les programmes ou portefeuilles ne produisent comme résultat que des user stories de haut niveau, il y aura donc un nombre considérablement moins important d'user stories. Cependant, les réunions offrent toujours de la valeur grâce à la participation de représentants issus des différents projets d'un programme ou des différents programmes d'un portefeuille. Cela permet de garantir que les exigences sont bien définies et comprises.

14.3.2.5 Entretiens d'utilisateurs ou de clients

Décris dans la section 8.4.2.4.

14.3.2.6 Questionnaires

Décris dans la section 8.4.2.5.

14.3.3 Résultats

14.3.3.1 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 afin de refléter une nouvelle priorisation des user stories existantes.

Voir également la section associée 10.3.3.1.

14.3.3.2 Améliorations recommandées du Scrum Guidance Body*

Décris dans la section 13.1.3.6.

Des suggestions ou du feedback, résultant du processus de *création et d'affinage du backlog de programme ou de portefeuille*, peuvent être formulés en vue d'apporter des améliorations potentielles au Scrum

Guidance Body. Ces améliorations recommandées seront discutées et acceptées ou rejetées par le Scrum Guidance Body (voir section 14.2 processus de *revue et mise à jour 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.

14.3.3.3 Échéances mises à jour d'implémentation des projets*

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.

14.3.3.4 Personnages

Décris dans la section 8.4.3.2.

14.3.3.5 Risques identifiés

Décris dans la section 8.4.3.4.

14.4 Coordination des composants pour les programmes ou les portefeuilles

14

La figure 14-9 illustre les contributions, outils et résultats du processus de *coordination des composants pour les programmes ou les portefeuilles*.

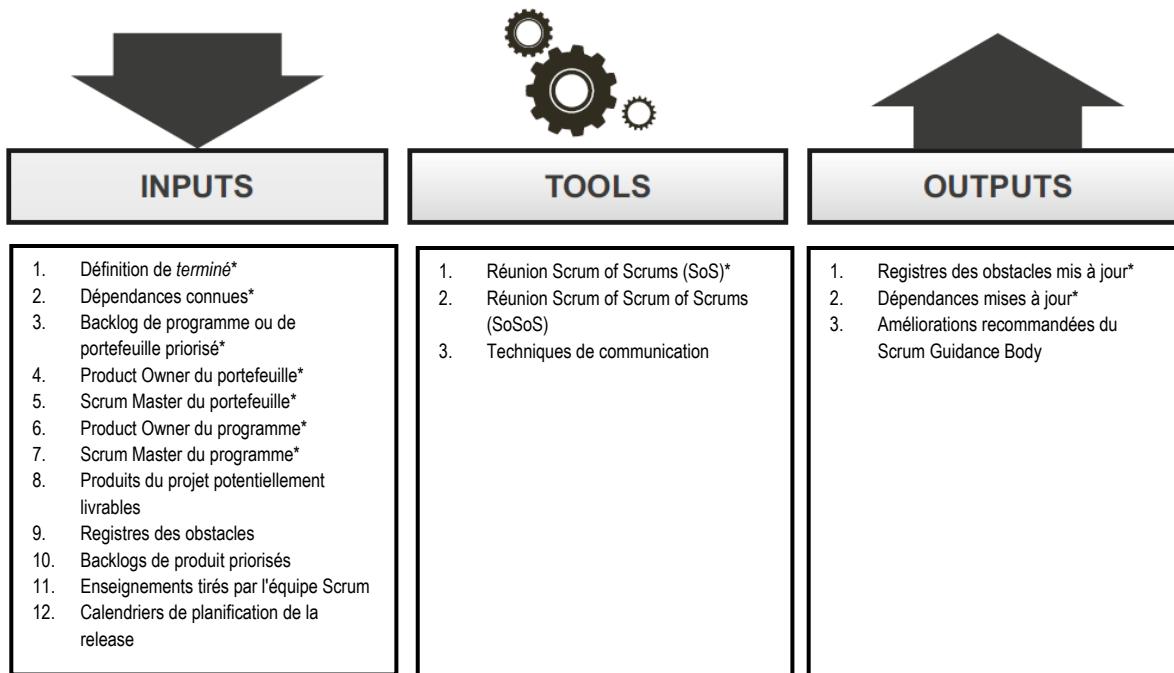


Figure 14-9 : Coordination des composants pour les programmes ou les portefeuilles — Contributions, outils et résultats

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

La figure 14-10 illustre les contributions, outils et résultats du processus de coordination des composants pour les programmes ou les portefeuilles.

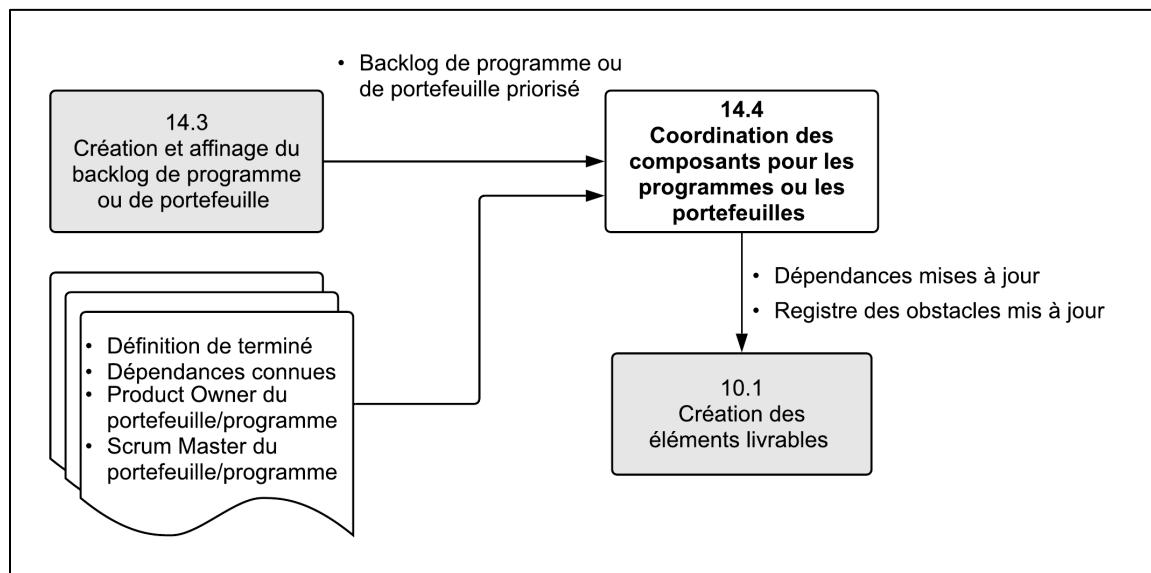


Figure 14-10 : Coordination des composants pour les programmes ou les portefeuilles — diagramme de flux de données

14.4.1 Contributions

14.4.1.1 Définition de terminé*

Décrise dans la section 5.4.2.

La définition de *terminé* établie au niveau du programme ou du portefeuille peut être utilisée comme définition minimum de *terminé* pour les projets à l'échelle de l'entreprise.

14.4.1.2 Dépendances connues*

En cas de projets et/ou produits étroitement liés au sein de l'entreprise, il peut être possible d'identifier des dépendances. Par conséquent, les projets doivent être coordonnés entre eux afin de gérer les dépendances. Ces dépendances peuvent inclure :

- des dates de release identiques pour des projets interdépendants
- des dépendances entre les releases
- des dépendances entre des fonctionnalités étroitement liées

Voir également les sections associées 9.4.2.3 et 13.1.2.4.

14

14.4.1.3 Backlog de programme ou de portefeuille priorisé*

Décrit dans les sections 14.3.1.2 et 14.3.1.3.

14.4.1.4 Product Owner du portefeuille*

Décrit dans les sections 3.4.4 et 14.3.1.4.

14.4.1.5 Scrum Master du portefeuille*

Décrit dans les sections 3.5.3 et 14.3.1.5.

14.4.1.6 Product Owner du programme*

Décrit dans les sections 3.4.3 et 14.3.1.6.

14.4.1.7 Scrum Master du programme*

Décrit dans les sections 3.5.2 et 14.3.1.7.

14.4.1.8 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 incrément de produit sont terminés. Les user stories incluses dans ces incrément sont conformes à la définition de *terminé* ainsi qu'à leurs critères d'acceptation respectifs.

Au fur et à mesure que les sprints des projets associés sont complétés, la façon dont chaque projet progresse devient de plus en plus claire. Ces connaissances permettent non seulement de continuellement mettre à jour les prévisions quant au respect ou non des échéances par tous les projets pour certaines exigences, mais cela fournit également des informations essentielles afin de gérer les dépendances entre les projets.

14.4.1.9 Registres des obstacles

Les obstacles auxquels les projets individuels doivent faire face peuvent s'appliquer à d'autres projets. Il est donc possible d'avoir à partager les registres des obstacles entre les projets et/ou programmes.

Décris dans la section 10.1.1.4. Voir également la section associée 13.2.1.2 concernant le partage des registres des obstacles entre les équipes dans un gros projet.

14.4.1.10 Backlogs de produit priorisés

Décris dans la section 8.5.3.1.

Ce sont les backlogs de produit priorisés au niveau d'un projet.

14.4.1.11 Enseignements tirés par les équipes Scrum

Décris dans la section 11.2.3.5.

14.4.1.12 Calendriers de planification de la release

Décrits dans la section 8.6.3.1.

Bien qu'ils soient hypothétiques et sujets à modifications, ces calendriers sont essentiels afin de juger si les projets respectifs sont à même de respecter les échéances requises, particulièrement en ce qui concerne les dépendances.

14.4.2 Outils

14.4.2.1 Réunion Scrum of Scrums (SoS)*

Décrise dans la section 13.2.2.1.

L'objectif de cette réunion est similaire à celui des gros projets. Au niveau d'un programme, les représentants de tous les projets du programme se réunissent à des intervalles réguliers lors des réunions Scrum of Scrums (SoS).

14.4.2.2 Réunion Scrum of Scrum of Scrums (SoSoS)

Au niveau d'un portefeuille, les représentants de tous les programmes et de projets « autonomes » dans le portefeuille se réunissent à des intervalles réguliers. Les représentants participant aux réunions Scrum of Scrums devraient être présents. Ce niveau supplémentaire de réunions est donc appelé Scrum of Scrum of Scrums (SoSoS). La figure 14-11 illustre le concept des réunions Scrum of Scrum (SoS) et Scrum of Scrum of Scrums.

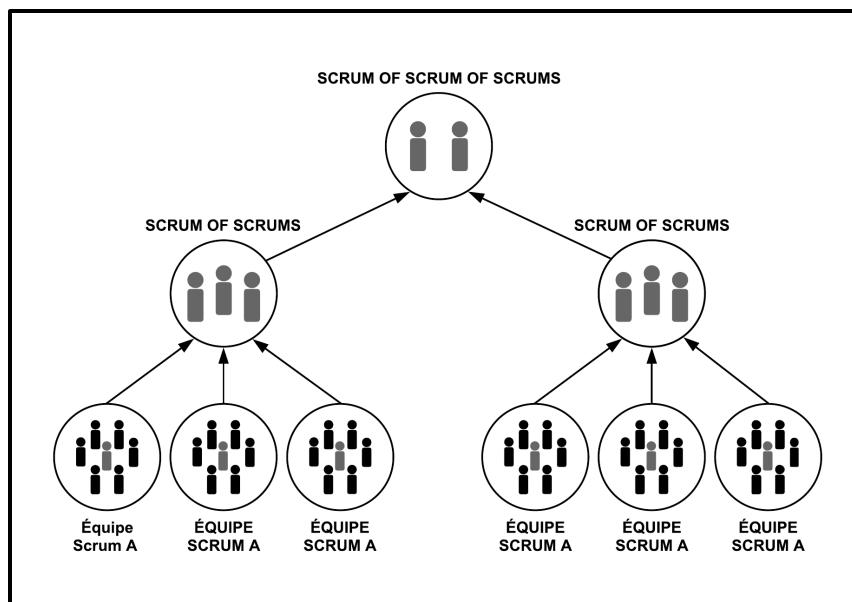


Figure 14-11 : Réunion Scrum of Scrums (SoS)

Dans cet exemple, six équipes Scrum travaillent en parallèle. Les équipes Scrum A, B et C travaillent sur des parties d'un programme tandis que les équipes Scrum D, E et F travaillent sur des parties d'un autre programme. Une réunion Scrum of Scrums sera organisée afin de coordonner les interdépendances entre les projets d'un programme. Une réunion Scrum of Scrums peut être organisée afin de gérer les interdépendances entre plusieurs ou tous les programmes.

14.4.2.3 Techniques de communication

Décrivées dans la section 10.3.2.2.

Des exemples de techniques de communication qui peuvent être utilisées au niveau d'un programme ou d'un portefeuille sont les tableaux d'affichage et la messagerie instantanée.

14.4.3 Résultats

14.4.3.1 Registres des obstacles mis à jour*

À la suite d'une réunion Scrum of Scrums (SoS) ou Scrum of Scrum of Scrums (SoSoS), il peut être nécessaire de mettre à jour les registres des obstacles.

Voir également la section associée 10.1.1.4.

14.4.3.2 Dépendances mises à jour*

À la suite du processus de coordination des composants pour les programmes ou les portefeuilles, il peut être nécessaire de mettre à jour les dépendances connues pour inclure de nouvelles dépendances ou modifier des dépendances existantes.

Voir également la section associée 9.4.3.3.

14.4.3.3 Améliorations recommandées du Scrum Guidance Body

14

Décrivées dans la section 13.1.3.6.

Des suggestions ou du feedback, résultant du processus de *coordination des composants pour les programmes ou les portefeuilles*, peuvent être formulés en vue d'apporter des améliorations potentielles au Scrum Guidance Body (SGB). Ces améliorations recommandées sont discutées et acceptées ou rejetées par le Scrum Guidance Body (voir section 14.2 processus de *revue et mise à jour 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.

14.5 Réunion de rétrospective de release de programme ou de portefeuille

La figure 14-12 illustre les contributions, outils et résultats du processus de *rétrospective des releases de programme ou de portefeuille*.

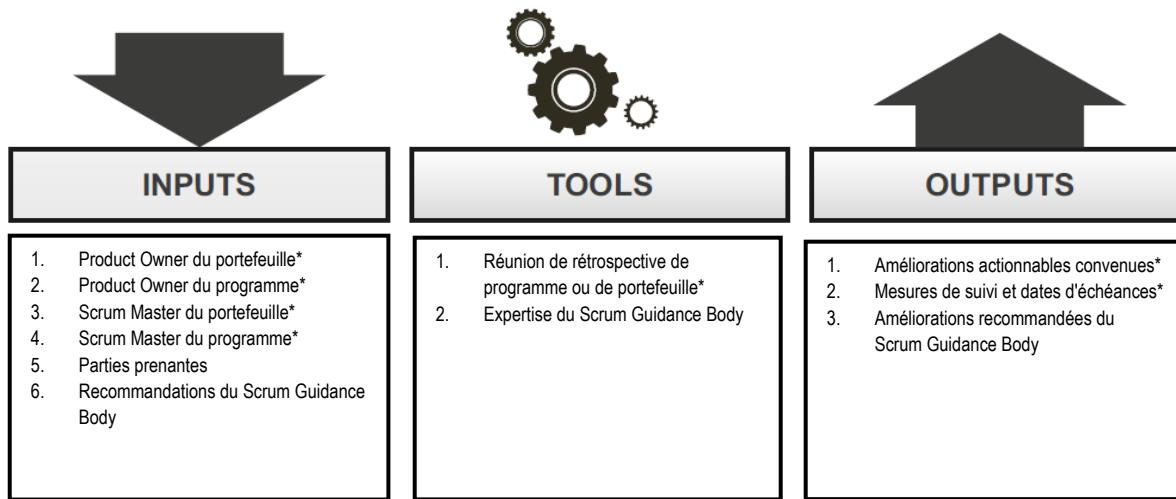


Figure 14-12 : Rétrospective des releases de programme ou de portefeuille — Contributions, outils et résultats

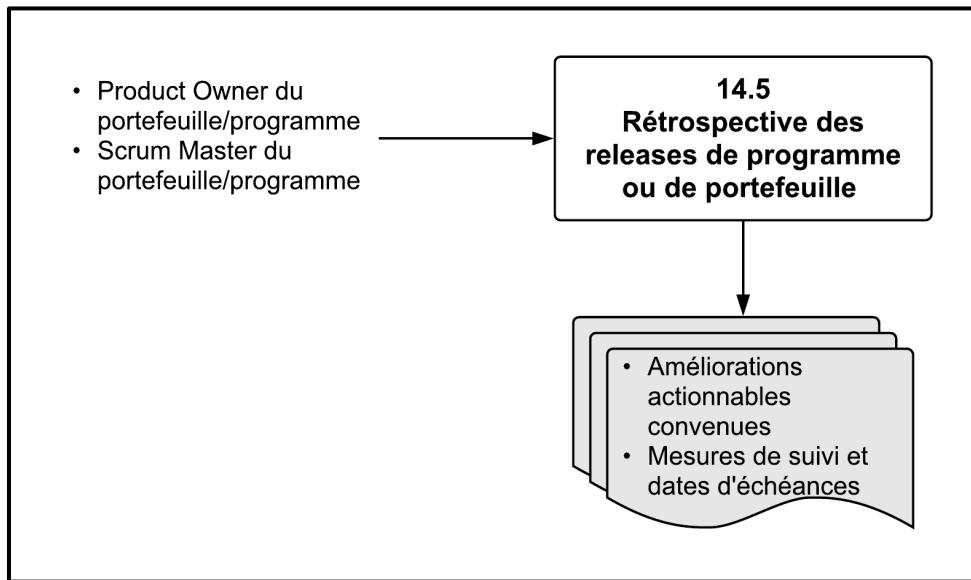


Figure 14-13 : Rétrospective des releases de programme ou de portefeuille — diagramme de flux de données

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

14.5.1 Contributions

14.5.1.1 Product Owner du portefeuille*

Décrit dans les sections 3.4.4 et 14.3.1.4.

14.5.1.2 Scrum Master du portefeuille*

Décrit dans les sections 3.5.3 et 14.3.1.5.

14.5.1.3 Product Owner du programme*

Décrit dans les sections 3.4.3 et 14.3.1.6.

14.5.1.4 Scrum Master du programme*

Décrit dans les sections 3.5.2 et 14.3.1.7.

14

14.5.1.5 Parties prenantes

Décrivées dans la section 3.3.2.

14.5.1.6 Recommandations du Scrum Guidance Body

Décrivées dans les sections 8.1.1.11 et 12.2.1.5.

Durant la rétrospective des releases d'un programme ou d'un portefeuille, les recommandations du Scrum Guidance Body fournissent des meilleures pratiques pertinentes, notamment des informations concernant les procédures administratives, audits, évaluations et critères de transition d'un projet. Le Scrum Guidance Body joue un rôle similaire à celui qu'il remplit lors des rétrospectives au niveau d'un projet (décrit dans la section 12.2.1.5).

14.5.2 Outils

14.5.2.1 Réunion de rétrospective de programme ou de portefeuille*

La réunion de rétrospective de programme ou de portefeuille est similaire à la réunion de rétrospective du projet, décrite dans la section 12.2.2.1, mais est organisée au niveau d'un programme ou d'un portefeuille. La principale différence est que les réunions de rétrospective d'un programme ou d'un portefeuille sont organisées beaucoup moins fréquemment que les réunions de rétrospective d'un projet.

14.5.2.2 Expertise du Scrum Guidance Body

Décrise dans la section 8.4.2.7.

14.5.3 Résultats

14.5.3.1 Améliorations actionnables convenues*

Décrise dans la section 11.2.3.1.

14.5.3.2 Mesures de suivi et dates d'échéances*

Décrise dans la section 11.2.3.2.

14.5.3.3 Améliorations recommandées du Scrum Guidance Body

Décrise dans la section 13.1.3.6.

Des suggestions ou du feedback, résultant du processus de *rétrospective des releases de programme ou de portefeuille*, peuvent être formulés en vue d'apporter des améliorations potentielles au Scrum Guidance Body (SGB). Ces améliorations recommandées sont discutées et acceptées ou rejetées par le Scrum Guidance Body (voir section 14.2 processus de *revue et mise à jour 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.

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 adaptives, il est également nécessaire d'intégrer des éléments de stabilité au sein de leurs processus.

A.2.1 Démocratisation du concept agile

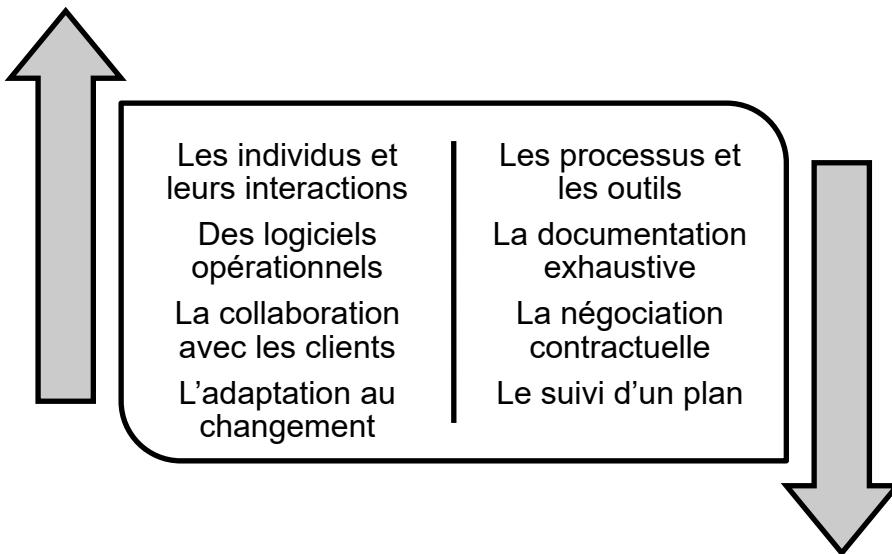
Les changements rapides en termes de technologie, de demandes de marché et d'attentes ont résulté en un nombre grandissant de défis associés au développement de produits et services en utilisant les modèles de gestion de projet traditionnels. Cela a mené à la conceptualisation ainsi qu'à l'implémentation 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 adaptive 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 adaptives 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 privilégiions 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 incrémentations 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'incrémentations 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 adaptive 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 évènements 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 incrément du projet afin de respecter des critères définis en priorisant les incrément 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 incrément 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 spéulation (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 Lyncock
Jaime 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 TROISIÈME ÉDITION

Cette annexe fournit un résumé des mises à jour apportées au *Guide SBOK™ — troisième édition* comparées à l'édition précédente.

C.1 Résumé des changements

Les mises à jour apportées au *Guide SBOK™ — troisième 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 troisième 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 « mélées 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 « mélées 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

Une analyse standard des parties prenantes permet d'identifier les parties prenantes au niveau du programme ou du portefeuille. Des détails supplémentaires sur les parties prenantes 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.

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 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.

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 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 incrémentations doivent être mises 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 évènements 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. Les parties prenantes 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 les parties prenantes 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 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 les parties prenantes 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 incrément 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 les parties prenantes.

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.

Création du backlog de sprint

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.

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 les parties prenantes 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/Swehart (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 incrément 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 incrément de produit au Product Owner et aux parties prenantes 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 incrément 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.

Dette technique

Lorsque l'équipe travaille à la création des principaux incrément 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 épics

L'énoncé de vision du projet sert de base au développement des épics. Des réunions de groupes d'utilisateurs peuvent être organisées pour développer les épics.

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 rôle de l'équipe Scrum principale. L'équipe Scrum travaille à la création des incrément de produit du projet et contribue à la création de la valeur commerciale pour toutes les parties prenantes 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 valeur 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. 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 incrément 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 les parties prenantes, 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

Le Scrum Master et les parties prenantes 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 délégant

Un leader délégant 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 serviteurs 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 serviteurs 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 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 incrément 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.

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'utility function 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 même 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

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 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, 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 création du backlog de sprint.

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 les parties prenantes. 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 (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 incrément 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 acceptées 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 incrément de produit sont terminés. Les user stories incluses dans ces incrément 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 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 du projet

Ce processus conclue le projet. Les parties prenantes de l'organisation et l'équipe Scrum principale se rassemblent afin d'effectuer une rétrospective du projet 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 création du backlog de sprint. 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 du projet 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 du projet.

Réunion de rétrospective du projet

La réunion de rétrospective du projet 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 incrémentés 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 backlog de produit 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 les parties prenantes 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 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 les parties prenantes 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 épics.

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 évènement ou une série d'évènements 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 mélées 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 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 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 six semaines) durant laquelle l'équipe Scrum travaille à la création des incrément 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, les parties prenantes 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 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.

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 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 mélées 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 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 face aux risques. Il définit le niveau ou la propension des parties prenantes 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
 Affinage du backlog de produit 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
 Attristant, 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 bénéfices, 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
- Création du backlog de sprint, 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 incrémentés 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, 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 serviteur, 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
- Mélée 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, 55
- Parties prenantes, 42, 104
- Parties prenantes du programme, 143
- Parties prenantes externes, 92
- Parties prenantes identifiées, 154
- Parties prenantes internes, 92
- Percent complete, 79
- Perdant-gagnant, 60
- Perdant-perdant, 60
- Personnages, 167
- Personnages mis à jour ou affinés, 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 du projet, 250
 - contributions, 256
 - outils, 257
 - résultats, 258
- Réunion de planification de sprint, 211
- Réunion de rétrospective du projet, 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 atténus, 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 principles, 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