

Les exemples à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **DP3SCRU** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

1. Objectif du livre	13
2. Notre démarche	13
3. Structure du livre	14
4. Remerciements	16

Chapitre 1

De la gestion de projet traditionnelle à l'Agilité

1. Introduction	17
2. Quelques faits et chiffres	17
3. Le modèle de gestion de projet "en cascade"	18
4. Le modèle (ou cycle) en V	20
4.1 La théorie	20
4.2 La mise en pratique du modèle en V	22
4.3 Les rôles	22
4.4 Notion d'effet tunnel	23
5. L'Agilité au cœur des projets	24
5.1 Un peu d'histoire	24
5.2 Les valeurs	25
5.3 Les 12 principes sous-jacents	25
5.4 L'Agilité, ce n'est pas l'anarchie	26
6. Scrum, un cadre de travail Agile	26
7. Information, formation et certifications	28
8. Pour conclure	29

Chapitre 2**Lean, Kanban et eXtreme Programming**

1. Un chapitre nécessaire	31
2. Liens de parenté entre les méthodes	32
3. Le Lean Management	32
3.1 Objectif du Lean	33
3.2 Les 14 principes du Lean.	33
4. Le Kanban	39
4.1 Principes du Kanban.	39
4.2 Le Kanban pour le développement de logiciel.	40
4.3 Kanban et Scrum	41
5. La méthode XP ou eXtreme Programming	42
5.1 Les principes de base.	42
5.2 Les pratiques d'eXtreme Programming	43
5.2.1 Livraisons fréquentes	43
5.2.2 Rythme durable	43
5.2.3 Client sur site	43
5.2.4 Conception simple	43
5.2.5 Mise en place de règles de codage	44
5.2.6 L'équipe est responsable du code	44
5.2.7 Utilisation de tests unitaires	44
5.2.8 Test de recette	45
5.2.9 Mise en place de l'intégration continue	45
5.2.10 Réaliser du refactoring de code	45
5.2.11 Programmation en binôme (Pair Programming)	45
5.2.12 Estimation à l'aide du Planning Poker	46
5.2.13 Utilisation de métaphores et analogies	46
5.3 Cycle d'eXtreme Programming	46
6. Scrum, un mix des méthodes.	47

Chapitre 3 Tour d'horizon de Scrum

1. Naissance de Scrum	49
2. Scrum en quelques mots	50
2.1 Les valeurs Scrum	51
2.2 L'équipe	52
2.3 Les trois piliers de Scrum	53
2.3.1 Transparence	54
2.3.2 Inspection	54
2.3.3 Adaptation	54
2.4 Les événements (cérémoniaux)	55
2.4.1 Le Sprint	55
2.4.2 La réunion de planification de Sprint	55
2.4.3 La mêlée quotidienne	56
2.4.4 La revue de Sprint	56
2.4.5 La rétrospective de Sprint	56
2.5 Les artefacts	56
2.5.1 Backlog Produit	57
2.5.2 Backlog de Sprint	57
2.5.3 Suivi de la progression	58
3. Cycle de vie de Scrum	60
4. Coût, délai et périmètre	61
5. Conclusion	62

Chapitre 4 L'équipe Scrum

1. L'équipe, point central de Scrum	63
1.1 Équipe auto-organisée	64
1.2 Équipe pluridisciplinaire	64
2. Le Scrum Master	65
2.1 Les responsabilités du Scrum Master	66

2.1.1	Application de Scrum	66
2.1.2	Lever les obstacles	67
2.1.3	Optimiser les interactions	68
2.1.4	Leader du changement	68
2.2	La personnalité et les compétences du Scrum Master	69
2.2.1	Connaître Scrum	69
2.2.2	Être un leader	69
2.2.3	Être communicant	69
2.2.4	Avoir des capacités de médiation	70
2.2.5	Jouer la transparence	70
3.	Le Product Owner	70
3.1	Les responsabilités du Product Owner	71
3.1.1	Créer la vision du produit	71
3.1.2	Gérer le Product Backlog	71
3.1.3	Maximiser la valeur du produit et du travail de l'équipe	72
3.1.4	Définir le plan de Release	72
3.1.5	Implication dans le processus Scrum	72
3.1.6	Accepter ou non le résultat d'un Sprint	73
3.1.7	Ses pouvoirs et limites	73
3.2	La personnalité et les compétences du Product Owner	74
3.2.1	Posséder des connaissances fonctionnelles	74
3.2.2	Être organisé	74
3.2.3	Avoir des capacités de prise de décision	74
4.	L'équipe de réalisation	75
4.1	Généralités	75
4.2	Caractéristiques	75
4.2.1	Auto-organisée et multi-disciplinaire	75
4.2.2	Taille de l'équipe	76
5.	Et les autres rôles, alors ?	76
5.1	La disparition du chef de projet	76
5.2	Les autres rôles	77
6.	Bien constituer l'équipe : quelques pistes	77

7. Créer les conditions de la réussite	78
7.1 Rassembler pour gagner	78
7.2 Cas d'une équipe morcelée	79
8. Conclusion	80

Chapitre 5 **Construire et prioriser le Product Backlog**

1. Pourquoi investir dans le Product Backlog ?	81
2. La brique de base du Product Backlog : la User Story	82
3. Comment rédiger les User Stories et Epics ?	83
3.1 Règle des 3C	83
3.2 Rédiger une bonne User Story : le principe INVEST	84
3.3 Erreurs courantes	85
3.4 La Story technique : solution ou aveu d'échec ?	86
3.5 Identifier les fonctionnalités clés avec le Product Box	87
3.5.1 Objectifs	87
3.5.2 Mode opératoire	88
3.6 Une méthode efficace de découverte du Product Backlog : le Story Mapping.	91
3.6.1 Le Story Mapping, c'est quoi ?	91
3.6.2 Story Mapping illustré par un exemple	92
3.7 Principes de priorisation du Product Backlog	95
3.7.1 Pourquoi prioriser ?	95
3.7.2 Approche générale de la priorisation	96
3.7.3 Les facteurs qui influencent la priorisation.	96
3.7.4 Survol des méthodes de priorisation	97
3.8 Zoom sur la priorisation par les thèmes	98
3.8.1 Theme Screening (sondage des thèmes)	98
3.8.2 Theme Scoring (mesure des thèmes)	100
3.8.3 Priorisation des thèmes par l'utilisation de poids relatifs.	101
3.9 Zoom sur la priorisation par l'utilisation du modèle de Kano ..	103

3.10 Zoom sur la méthode MoSCoW	108
3.11 Zoom sur la méthode WSJF.	109
4. Gérer son Backlog en pratique	111
5. Conclusion	112

Chapitre 6

Planifier et estimer

1. Des pratiques à ne surtout pas négliger	113
2. Pourquoi la planification traditionnelle échoue	113
3. Horizons de planification.	115
4. Outils d'estimation.	115
4.1 T-Shirt sizing	115
4.2 Les story points.	116
4.3 Alors, story point ou jh ?	118
4.4 Notion de vélocité	119
4.5 Comment initialiser la vélocité ?	120
4.5.1 Mise en place d'un projet test	120
4.5.2 Choisir au feeling	121
4.5.3 Estimation de la vélocité à partir de l'historique	121
4.6 Qui estime ?	123
4.7 Une méthode pratique d'estimation : le Planning Poker	123
4.7.1 Le déroulement du Planning Poker	124
4.7.2 Bénéfices et risques.	125
4.7.3 Erreurs communes	125
4.7.4 Découper pour bien estimer : Atelier Carpaccio.	126
5. Planification de Release	128
5.1 Avoir un objectif clair.	128
5.2 Posséder un Product Backlog priorisé	128
5.3 Estimer le Product Backlog.	129
5.4 Connaître la vélocité de l'équipe	129
5.5 Définir la fin	130

5.6 Définir la durée des Sprints	130
5.7 Créer le plan de Release	130
6. Conclusion	131

Chapitre 7

La vie d'un Sprint

1. Introduction	133
2. Quelle durée pour les Sprints ?	133
3. Doit-il y avoir un Sprint 0 ?	134
4. Le rythme du Sprint : vue d'ensemble	135
5. Préparation du Sprint	136
5.1 Environnement de travail	136
5.2 Équipe	136
5.3 Définition de « Terminé »	137
6. Réunion de planification de Sprint	138
6.1 Pourquoi la présence du Product Owner est-elle importante ?	138
6.2 Definition of Ready	139
6.3 Première étape : présentation des User Stories	139
6.4 Deuxième étape : quel travail sera réalisé durant le Sprint ?	140
6.5 Troisième étape : comment réaliser le travail prévu ?	141
6.5.1 Estimation des tâches	142
6.5.2 Affectation des tâches	142
6.6 La gestion du temps	143
6.7 Et les corrections de bugs ?	144
6.8 Backlog Grooming	145
7. Mêlée quotidienne (Scrum Meeting/Daily Scrum)	145
7.1 Un protocole à respecter	145
7.2 Une mêlée efficace et utile	146
7.3 Le Scrum Master toujours à l'écoute !	148

7.4	Suivi de l'avancement.	148
7.5	Je n'ai plus rien à faire !	149
7.6	L'objectif du Sprint sera-t-il atteint ?	150
8.	La revue de Sprint (Sprint Review)	152
8.1	Qui, quoi, combien de temps ?	152
8.2	Un objectif, une motivation	154
8.3	Démontrer ce qui n'est pas démontrable...	154
9.	La rétrospective de Sprint	155
9.1	Une méthode pour vous aider	155
9.2	État d'esprit.	156
9.3	Environnement de la rétrospective	156
9.4	Méthode numéro 1 : Kick Drop Start	157
9.5	Méthode « classique »	158
9.6	Présentation « en étoile »	159
9.7	Le SpeedBoat	159
9.8	Autres méthodes.	161
10.	Laisser l'équipe se reposer	161
11.	Et si on recommençait ?	162

Chapitre 8

Tester en mode Agile

1.	Adopter Scrum : quel impact sur la stratégie de test ?	163
2.	Typologies de tests	164
2.1	Tests fonctionnels de validation	164
2.1.1	Critères de validation	165
2.1.2	Les données de tests et scénarios.	165
2.1.3	Les tests de validation et les User Stories	166
2.2	Tests de non-régression	166
2.3	Tests d'IHM (interface homme-machine)	167
2.4	Tests fonctionnels « de bout en bout »	167
2.5	Tests de composants	168

- 2.6 Tests unitaires. 168
- 2.7 Test-Driven Development 171
- 2.8 Acceptance Test-Driven Development 172
- 3. Anti-pattern : le cornet de glace 173
- 4. La pyramide de tests idéale 174
 - 4.1 Trouver du temps pour l'écriture. 176
 - 4.2 Faut-il écrire les tests de toutes les User Stories ? 176
 - 4.3 Comment écrire les tests ? 176
 - 4.4 Definition of Done et tests d'acceptation. 177
- 5. Les testeurs dans l'équipe Scrum 177
 - 5.1 Le test fait partie de l'équipe 177
 - 5.2 Testeur Agile : un métier en mutation. 178
- 6. En conclusion : écrivez des tests ! 178

Chapitre 9
Conseils pour déployer Scrum

- 1. Comment mener le changement vers Scrum ? 179
- 2. État des lieux. 180
 - 2.1 Adoption des méthodes Agiles. 180
 - 2.1.1 Scrum largement déployé. 180
 - 2.1.2 Les motivations pour l'adoption de Scrum. 181
 - 2.1.3 Comment Scrum est pratiqué ? 182
 - 2.1.4 Succès et challenges 182
 - 2.2 Un bilan positif 184
- 3. La motivation 184
- 4. Big-Bang ou déploiement progressif ? 185
- 5. Scrum et l'organisation en place 186
 - 5.1 Que faire des responsabilités existantes ? 186
 - 5.2 La structure 187

6.	En finir avec les idées reçues	188
6.1	Scrum n'est pas une approche structurée	188
6.2	Il n'existe pas de notion de planification	189
6.3	Scrum bannit la documentation	189
6.4	Avec Scrum, nous passons trop de temps en réunion	189
7.	Le soutien du management	190
8.	Vaincre les résistances à la conduite de changement	190
8.1	Résistance d'intérêt ou politique	190
8.2	Résistance de confort	191
8.3	Résistance d'incapacité ou affective	192
9.	Utiliser les Serious Games pour faciliter le déploiement	193
9.1	Pour briser la glace... ..	193
9.1.1	Le secret du succès	193
9.1.2	On se classe par âge	194
9.1.3	Le réseau social papier	194
9.2	Le Marshmallow Challenge	195
10.	Se faire accompagner	196
11.	Nos conseils en guise de conclusion	196

Chapitre 10

Scrum à l'aide d'un logiciel

1.	Faut-il obligatoirement utiliser un logiciel ?	199
2.	Tour d'horizon des outils de gestion de projets Scrum	200
2.1	Jira	201
2.2	Axosoft	204
2.3	iceScrum	207
2.4	Tuleap	209
3.	Autres outils utiles	213
3.1	Story Mapping	213
3.2	Pour les rétrospectives d'équipes distribuées : IdeaBoardz. . .	215
3.3	Outils de collaboration d'équipe	215

4. Conclusion	216
---------------------	-----

Chapitre 11 **Aller plus loin**

1. Avant-propos	217
2. Mise à l'échelle de Scrum	218
2.1 LeSS	218
2.1.1 Les principes de base	218
2.1.2 Qu'est-ce qui est différent de Scrum dans LeSS ?	219
2.1.3 Notre avis sur LeSS	222
2.2 SAFe	222
2.2.1 Introduction	222
2.2.2 Les fondations	223
2.2.3 Le niveau « Équipe »	224
2.2.4 Le niveau « Programme »	224
2.2.5 Le niveau « Gestion de portefeuille »	226
2.2.6 Notre avis sur SAFe	226
2.3 La méthode Spotify	226
2.3.1 Description du modèle	226
2.3.2 L'équipe	227
2.3.3 Les tribus	228
2.3.4 Les guildes et chapitres	228
2.3.5 Notre avis sur le modèle Spotify	229
3. Scrum et Kanban	229
3.1 Différences entre les deux méthodes	229
3.2 Mixer ou non les approches	230
3.2.1 Quelques considérations générales	230
3.2.2 Un hybride de plus en plus utilisé : ScrumBan	231
3.2.3 Faire son choix	234
4. Scrum et DevOps	234

5. Scrum et contractualisation	236
5.1 Contradictions	237
5.2 Créer les conditions de la confiance	238
5.3 Répondre à un appel d'offres	239
5.4 Mise en place d'un plan d'assurance qualité (PAQ)	240
5.5 Contractualisation par Sprint	241
5.5.1 Comment calculer le coût d'un Sprint ?	241
5.5.2 Quid des User Stories non livrées ?	242
5.5.3 Gestion des bugs et « non-validation » de User Stories	242
5.6 Différentes formes de contrats envisageables	243
5.6.1 Coûts variables	243
5.6.2 Coûts fixes, périmètre variable	244
5.6.3 Coûts fixes, périmètre fixe	244
5.6.4 Coûts fixes, périmètre fixe mais avec ajustement	244
5.6.5 Budget par itération	245
5.6.6 Utilisation d'une marge de profit	245
5.6.7 Mise en place de pénalités	245
5.6.8 Travail collaboratif	246
5.6.9 Money For Nothing (payer pour rien) et Change For Free (changement offert)	246
5.7 Exemple de contrat type	247

Chapitre 12

Vérifiez vos connaissances

1. Pourquoi ce questionnaire ?	249
2. Les questions	250
3. Les réponses	258
4. L'heure du résultat	268
5. Il est temps de se quitter	268

Index	271
-----------------	-----

Introduction

Première partie : Scrum

- 1. Les origines 25
- 2. Scrum 26

**Chapitre 1-1
Le cadre SCRUM**

- 1. Introduction 43
- 2. Les valeurs Scrum 43
- 3. Les rôles 51
 - 3.1 Le Product Owner 52
 - 3.2 L'équipe de développement 58
 - 3.3 Le Scrum Master 83
 - 3.4 Les stakeholders 85
- 4. Les outils Scrum 87
 - 4.1 Technique pour recueillir les besoins 88
 - 4.2 Le Story Mapping 96
 - 4.3 Design Thinking 113
 - 4.4 Les Users Stories 134
 - 4.4.1 INVEST 138
 - 4.4.2 Bien écrire les Users Stories 141
 - 4.4.3 Les critères d'acceptation 144
 - 4.5 Les Epics 147
 - 4.6 Le sprint 154
 - 4.7 Definition of Ready (DoR) 157
 - 4.8 Definition of Done (DoD) 161
 - 4.9 Les Backlogs 163
 - 4.10 Le Sprint Planning 168

2 De Scrum à SAFe

Au cœur de l'agilité

4.11 La Daily Scrum	170
4.12 La Sprint Review	172
4.13 La Sprint Retrospective	174
4.14 Le Backlog Refinement	175
4.15 Le Burndown Chart	177
4.16 Le Burnup Chart	183
4.17 Autres mesures	189
4.18 Les Personas	192
4.19 La Release Planning	193
4.20 eXtreme quotation	198
4.21 Les tableaux d'avancement	200

Chapitre 1-2 L'esprit agile

1. Introduction	205
2. Le Manifeste	208
3. Les quatre valeurs	211
3.1 Préférence sur les individus et leurs interactions	211
3.2 Préférence à des applications qui fonctionnent	212
3.3 Préférence pour une collaboration avec le client	215
3.4 Préférence pour l'adaptabilité au changement	217
4. Le cadre Scrum	219

Chapitre 1-3 Les outils techniques

1. Code versioning	223
2. Intégration continue	229
3. Déploiement continu	233
4. Les microservices	235

Chapitre 1-4
Rétrospective sur la première partie

- 1. Ce que nous avons vu..... 251
- 2. Quelques feedbacks 253
- 3. Origine de Scrum 260

Conclusion de la première partie 263

Deuxième partie : SAFe

Chapitre 2-1
Prenons un train SAFe en marche

- 1. Introduction 273
- 2. Le Program Increment (PI)..... 273
- 3. Les trains 278
- 4. Les niveaux SAFe 285
 - 4.1 Le Program Level..... 287
 - 4.2 Le Large Solution Level..... 298
 - 4.3 Le Portfolio Level 307
 - 4.3.1 Le ROI, c'est quoi ou c'est qui ? 308
 - 4.3.2 Autres techniques d'évaluation 312
 - 4.3.3 Lean Portfolio Management 315
 - 4.3.4 WSJF ou comment prioriser un Portfolio Backlog 319
 - 4.3.5 Application de la budgétisation participative..... 323

Chapitre 2-2 PI Planning

1. Introduction	325
2. Rendez-vous au PI Planning	327
3. Le COVID-19 ne nous arrêtera pas !	337
4. Bien préparer le PI Planning (PIP)	339
4.1 Au niveau Team	341
4.2 Au niveau Program	344
4.3 Au niveau Large Solution	348
4.4 Au niveau Portfolio	352

Chapitre 2-3 Exécution du PI

1. Introduction	355
2. System Demo	356
3. Le PO Sync	361
4. Scrum of Scrums	363
5. IP Itération	365
5.1 L'atelier Inspect & Adapt	366
5.1.1 Le PI System Demo	367
5.1.2 Mesurons, sortons les KPI	370
5.1.3 Rétrospective et atelier de résolution des problèmes	378
5.2 Le reste à faire (RAF)	379
5.3 Le PI Planning et sa préparation	379
5.4 Innovation	380
5.5 Veille technique	381
5.6 Mise en production (MEP)	381
5.7 Backlog Refinement	382
6. Le mécanisme bien huilé	382

Chapitre 2-4
Les principes de SAFe

- 1. Introduction385
- 2. À vous de jouer maintenant !389

Glossaire395

Index415