

Contexte

1. Introduction	11
2. Enjeu de Talentsoft	13
3. Objectifs de Talentsoft	17
4. L'agilité comme remède miracle	18
4.1 Mise en place de l'agile	18
4.2 Les problématiques actuelles	19
5. La solution hybride	20

Passage à l'agile

1. Introduction	23
2. Cas des dirigeants	23
3. Cas des Product Owners	26
3.1 Rôle et missions du Product Owner	27
3.2 Caractéristiques du Product Owner	29

3.3 Disponibilité du Product Owner	30
4. Cas des architectes logiciels	33
5. Cas des équipes de développement	34
5.1 Deux types de réactions	34
5.2 Bénéfices de l'agile	35
5.2.1 Collaboration avec le client	35
5.2.2 Motivation des équipes	36
6. Cas des managers d'équipe	36
6.1 Rôle du leader technique R&D	39
6.2 Équilibrer la pression dans l'équipe	39
6.3 Équilibrer qualité et productivité	41
6.4 Accompagner le changement	42
6.5 Devenir un leader	43
7. Cas des clients	44
8. Contractualisation des livrables	45

Ce qu'il faut savoir pour lire la suite

1. Introduction	47
2. Le manifeste agile	47
2.1 Des individus et des interactions	49
2.2 Des logiciels opérationnels	50
2.3 Une collaboration avec les clients	51
2.4 Une adaptation au changement	52
3. Les méthodologies agiles	53
3.1 Scrum	54
3.2 Kanban	55
3.2.1 Value Stream Map	57
3.2.2 Limitation des travaux en cours	57
3.2.3 Mesure de Lead Time	58
3.2.4 En quoi Kanban peut-il aider une équipe ?	58
3.3 eXtreme Programming	59
3.4 Différences entre Scrum, Kanban et XP	61
3.5 Lean Startup	61
3.5.1 Lean Canvas	62
3.5.2 MVP et Build/Measure/Learn	63
3.6 Quelle méthodologie choisir ?	64

4. Le sprint (ou itération)	65
4.1 Priorités changeantes et sprint en cours	65
4.2 Le sprint est un incrément fonctionnel	66
5. Les backlogs	67
5.1 Backlog de produit	67
5.2 Backlog de sprint	68
6. Les personae	70
6.1 Donner du réalisme	70
6.2 Comment procéder pour créer le persona ?	71
7. Les User Stories	71
7.1 Format des User Stories	72
7.2 INVEST	73
7.3 Epic Stories et taille des stories	76
7.3.1 Taille des User Stories	77
7.3.2 Epic Stories et fonctionnalités	78
7.4 Visibilité des User Stories	78
7.5 Estimation des User Stories	79
7.6 Critères d'acceptation	83
7.6.1 Critères d'acceptation et Definition of Done	

7.6.2 Utilisation des critères d'acceptation	84
7.6.3 Format des critères d'acceptation	84
7.7 Représentation des différentes Stories	85
8. Les Technical Stories	86
8.1 Dette technique et architecture d'entreprise	87
8.2 Tâches communes à plusieurs User Stories	89
9. Les boards	91
9.1 Scrum Board	91
9.2 Kanban Board	93
9.3 Autres boards	96
10. Le Burndown Chart	97
10.1 Intérêt du Burndown Chart	97
10.2 Autres types de Burndown Chart	99
Mise en place des sprints	
1. Introduction	101
2. Définir des processus de développement	101

2.1 Utilité des processus	102
2.2 Définition des processus	103
3. Exemple de processus complet dans une méthodologie agile	105
4. Passage d'une étape à l'autre : la Definition of Done	110
4.1 Spécifier ses Definitions of Done	111
4.2 Types de Definition of Done	112
4.3 Exhaustivité et évolution	112
4.4 Bénéfices de la Definition of Done	114
4.4.1 Responsabilisation des développeurs	114
4.4.2 Qualité	115
5. Rôle des architectes logiciels	115
5.1 Impacts des méthodes agiles sur le rôle de l'architecte	116
5.2 Architectes (du) système	119
5.3 Architectes d'entreprise	122
6. Rôle du chargé de projet technique	124
6.1 Impacts des méthodes agiles	124
6.2 Rôle du chargé de projet	125
7. Documentation des projets	

	127
8. Gestion de la dette technique	130
8.1 Dette technique en environnement réel	130
8.1.1 Impacts sur la satisfaction client	131
8.1.2 Impacts sur la gestion de projet	131
8.1.3 Impacts sur la prévisibilité	134
8.2 Gestion de la dette technique au quotidien	135
8.2.1 Sensibilisation du management	135
8.2.2 Dédier des ressources à la dette technique	137
8.2.3 Résorber la dette technique dans tout développement	138
8.3 Incapacité à rembourser ses dettes	139
9. Durée d'un sprint	140

Méthodes utilisées

1. Introduction	145
2. Communiquer	145
2.1 Favoriser la communication	146
2.2 Identifier les problèmes de communication	148
2.3 Résoudre les problèmes de communication	

	149
3. Taille d'équipe	151
4. Mise en situation du développeur	153
4.1 Comprendre son projet	153
4.2 Connaître les attentes du client	154
4.3 Motiver en impliquant	155
5. Travail en binôme	156
5.1 Niveau 1 : répartition des tâches	157
5.2 Niveau 2 : Pair Programming	158
6. Règles de codage	159
6.1 Référentiel de règles de codage	160
6.2 Champ d'application des règles de codage	160
7. Domain-Driven Design	161
8. Test-Driven Development	162
8.1 Bénéfices du TDD	164
8.2 TDD et dette technique	164
9. Behavior-Driven Development	165

9.1 Réduire la documentation produit	165
9.2 Périmètre et coût	166
10. Revues de code	168
10.1 Pair Review	172
10.2 Validation technique continue	172
10.3 Assurance qualité technique	173
11. Démonstrations régulières au Product Owner	173
12. Maquettage de produit	174
13. Amélioration continue	174
14. Intégration continue	175
14.1 Couverture du code par les tests	178
14.2 Analyse statique du code	178
14.3 Automatisation des livraisons	179
14.4 Fréquence des livraisons et déploiement continu	179
14.5 Intégration continue et dette technique	181
15. Décliner le feedback pour en profiter au maximum	182
15.1 Early Adopters	183

15.2 Tests A/B	183
15.3 Utilité des réseaux sociaux	184
15.4 Fail Fast	184
15.5 Rétrospectives	185
15.6 Encourager la communication	185
16. 20 % de « Free Time »	186
16.1 Essayer ses propres idées	186
16.2 Origine des idées	188
16.3 Organisation du Free Time	189
16.3.1 Organisation du travail	189
16.3.2 Gestion du temps	192
16.4 Distinguer Free Time et Roadmap	192
 Réunions agiles	
1. Introduction	195
2. Backlog Refinement Meeting	195
2.1 Utilité	196
2.2 Déroulement de la réunion	197
2.3 Effets potentiels sur la priorisation	198

3. Sprint Planning Meeting	199
4. Daily Scrum Meeting	200
4.1 Organisation des Scrum Meetings dans une grande équipe	202
4.2 Scrum Meetings de Scrum Meetings	203
5. Revue de sprint	204
6. Rétrospective de sprint	205
7. Rétrospective d'anomalies (ou Post-Mortem)	206
8. Présentations R&D	207
9. Big Code Review	209
10. One on One Meeting	210
11. Synthèse des réunions	211
11.1 Backlog Refinement Meeting	211
11.2 Sprint Planning Meeting	212
11.3 Daily Scrum Meeting	212
11.4 Revue de sprint	212
11.5 Rétrospective de sprint	213

11.6 Rétrospective d'anomalies	214
11.7 Présentations R&D	214
11.8 Big Code Review	215
11.9 One on One Meeting	215
	216

Gérer son projet agile

1. Introduction	217
2. Gérer son équipe	218
2.1 Organisation fonctionnelle	219
2.2 Organisation en projets/produits	220
2.3 Organisation matricielle	224
3. Équipes et culture d'entreprise	225
4. Gérer son backlog de produit	227
4.1 Prioriser son backlog de produit	227
4.2 Prioriser son backlog de sprint	229
5. Gérer son backlog d'architecture	231
5.1 Effort d'architecture dans un projet	232

5.2 Organisation des travaux d'architecture	233
5.3 Organisation des équipes d'architecture	234
6. Minimum Viable Product	236
7. Gérer la qualité et les anomalies	236
7.1 Développer sur un socle instable	236
7.1.1 Différents cas de figure	237
7.1.2 Réduire la difficulté	239
7.2 C'est l'auteur de l'anomalie qui la corrige	240
7.2.1 Impliquer la chaîne de développement	240
7.2.2 Root Cause Analysis	241
7.3 Privilégier la qualité	241
8. Estimer	242
8.1 Unité d'estimation	242
8.1.1 Que représente un Story Point ?	243
8.1.2 Estimation en unités de temps	243
8.1.3 Estimation en Story Points	246
8.2 Définir des Stories de référence	248
8.2.1 Périmètre de la référence	249
8.2.2 À quel moment définir des User Stories de référence ?	250

8.3 Périmètre des User Stories	250
8.4 Manque d'informations pour estimer	251
8.5 Découpage en tâches	252
8.5.1 Taille des tâches	253
8.5.2 Définir les tâches pour terminer une User Story	253
8.5.3 Estimation des tâches	254
9. Planifier	255
9.1 Planification agile ou prédictive	255
9.2 Vitesse	262
9.2.1 Corriger la vitesse en fin de sprint	262
9.2.2 Unifier la vitesse de plusieurs équipes	263
9.2.3 Planifier à long terme avec la vitesse	264
9.2.4 Ne pas convertir les points en unités de temps	265
9.3 Donner une date de fin	266
10. Métriques agiles	267
10.1 Burndown Chart	267
10.2 Humeur de l'équipe	269
10.3 Vitesse	269
11. Vers l'autogestion	270

Outillage pour l'agilité

1. Introduction	273
2. Tableur	274
3. Trello	275
4. Tableau de post-its	279
5. Outils d'ALM	281
5.1 Jira Agile	282
5.1.1 Pilotage des projets	283
5.1.2 Simplification des tâches quotidiennes	284
5.2 Team Foundation Server	284
5.2.1 Intégration de TFS dans l'écosystème logiciel	285
5.2.2 Recueil du feedback avec TFS	286
6. User Story Map	286

Retour d'expérience

1. Introduction

Conduite de projets agiles

Management alternatif dans une équipe de développement agile

289

2. Retour des dirigeants

289

3. Retour des Product Owners

290

4. Retour des architectes

291

5. Retour du management

291

6. Retour de l'équipe de développement

292

Index

293