

Avant-propos

1. Une transformation digitale	13
2. Pourquoi ce livre ?	14
3. À qui est destiné ce livre ?	14
3.1 Cas n°1 - C'était mieux avant !	14
3.2 Cas n°2 - Comment puis-je améliorer mon agilité dans mes tests ?	15
3.3 Cas n°3 - Mon équipe veut passer au DevOps, quels sont mes besoins de tests ?	15
3.4 Cas n°4 - Je suis dans une équipe agile en « Fast IT » et tout le monde code et teste	16
4. Comment lire ce livre ?	16
5. Qui suis-je ?	18

Chapitre 1

Quelques rappels

1. Introduction	21
2. Les 4 valeurs de l'agile - Rappel du Manifeste Agile	22
3. Les 7 principes du test - Rappel ISTQB	23
3.1 Les tests montrent la présence de défauts	23
3.2 Les tests exhaustifs sont impossibles	24
3.3 Tester tôt	24
3.4 Regroupement des défauts	26
3.5 Paradoxe du pesticide	26
3.6 Les tests dépendent du contexte	28
3.7 Illusion de l'absence de défaut	28
4. Techniques de test de base	29
4.1 Partitionnement de classes d'équivalences et valeurs limites	29
4.2 Heuristique CORRECT	32
4.3 Table de décision	34
4.3.1 Approche analytique	34
4.3.2 Heuristique - Test n-wise	36
4.4 Transitions d'états	37

2 _____ Le test en mode Agile

5. Le Cycle de vie logiciel	39
5.1 Projet à l'échelle d'une seule équipe SCRUM	39
5.1.1 Scrum en quelques lignes.	40
5.1.2 Sprint Refinement Meeting (grooming).	42
5.1.3 Sprint planning	44
5.1.4 Daily Scrum Meeting.	47
5.1.5 Sprint Review.	50
5.1.6 Rétrospective	51
5.1.7 Réflexions sur les rétrospectives	63
5.2 Projet à l'échelle de plusieurs équipes	65
5.2.1 Cas de frameworks agiles existants	66
5.2.2 Point de vue du modèle SAFe	66
5.2.3 Point de vue du modèle Spotify	72
5.2.4 Point de vue Less	76
5.2.5 Point de vue DAD - « Disciplined Agile Delivery ».	79
5.2.6 Point de vue Scrum@Scale - S@S.	82
5.3 Approche Composant vs Fonctionnalité.	84

Chapitre 2 L'état d'esprit

1. Introduction - Exemple de l'industrie du textile	91
1.1 Contrôle des ballots	93
1.2 Contrôle pendant le tissage	93
1.3 Amélioration du procédé de tissage	94
1.4 Le test agile	95
2. État d'esprit Lean	95
2.1 Le Lean, les origines de l'agile	95
2.2 Quelques pratiques Lean utiles au test	96
2.2.1 Kanban	97
2.2.2 Juste à Temps - « Just in time ».	97
2.2.3 5S.	98
2.2.4 Intégrer la qualité dès la conception	99
2.2.5 Le First Pass Yield : faire bon du 1er coup !	100
2.2.6 Analyse de cause racine	100
2.2.7 Petits stocks, régularité des flux	109

2.3	Manifeste du test agile	109
2.3.1	Tester pendant le sprint	110
2.3.2	Prévenir les bugs.	111
2.3.3	Comprendre ce que l'on doit tester	113
2.3.4	Concevoir un meilleur système	114
2.3.5	L'équipe est responsable de la qualité du produit	114

Chapitre 3 **Versant métier**

1.	Introduction.	125
2.	Définition du Prêt (DoR)	125
3.	Clarté d'une US	129
4.	Test des US	130
4.1	Ingénierie des critères d'acceptation des US	130
4.1.1	Définition du Terminé (DoD)	131
4.1.2	BDD, Gherkin et ATDD	132
4.2	Préparation de la revue d'US ou d'Epic	141
4.3	Revue de définition des US	142
4.3.1	Affinement des US	142
4.3.2	Approche « Los Très Amigos ».	143
4.3.3	Visite guidée.	144
4.4	Revue de réalisation de l'US.	145
5.	Le Produit	146
6.	La valeur métier	149
6.1	Modèle MoSCoW	149
6.2	Diagramme de Kano.	149
6.3	Priorisations complexes (Weighted Shortest Job First - WSJF)	152
7.	Production des spécifications	153
7.1	Cas des exigences et spécifications	153
7.2	Cas des US.	154
7.2.1	Le rôle de chacun	154
7.2.2	Le récit utilisateur	155

4 _____ Le test en mode Agile

7.3	Cas des Epics	157
7.3.1	Les Features - les Fonctionnalités	158
7.3.2	Les Capabilities.	160
7.3.3	Les Epics du Portfolio.	161

Chapitre 4 **Versant industriel du test**

1.	Introduction	163
2.	Stratégie de test	163
2.1	Vision académique de la stratégie de test	163
2.2	Catégories d'approches.	166
2.3	Tester tôt.	169
2.4	Approche par le risque	171
2.4.1	Caractérisation du niveau de risque	171
2.4.2	Principe du Risk-Based Testing - RBT	174
2.4.3	Granularité	176
2.4.4	Niveau « données de tests »	176
2.4.5	Niveau « scénarios de tests »	176
2.4.6	Niveau « exigence de tests »	177
2.4.7	Niveau « campagnes de tests »	177
2.4.8	Exemple de gestion des campagnes de tests de régression	179
2.4.9	Périmètre vs Qualité et objectifs de test.	180
2.5	Heuristique pour une stratégie de test	182
2.6	Context-Driven Testing (CDT) - Test Piloté par le Contexte.	186
2.6.1	Valeur d'une pratique dépendante du contexte.	187
2.6.2	Les Best Practices	188
2.6.3	Les gens qui travaillent ensemble	188
2.6.4	Prédictibilité des projets.	189
2.6.5	Résolution du problème.	189
2.6.6	Le challenge intellectuel.	189
2.6.7	Esprit critique.	189
3.	Tests systèmes et bout-en-bout	189
3.1	Organisation des tests scénarisés	190
3.2	Ingénierie des Tests systèmes/bout-en-bout.	191

3.3	Campagne de test	192
3.3.1	Planification	193
3.3.2	Monitoring et Contrôle	194
3.3.3	Clôture	195
3.4	Tests de régression	195
3.4.1	Smoke tests	195
3.4.2	Campagnes prédéfinies par le risque	196
3.4.3	Campagnes prédéfinies par thème	198
3.4.4	Check-list	199
4.	Tests exploratoires	200
4.1	Préparation des tests exploratoires	200
4.1.1	La charte	202
4.1.2	La zone de recherche	202
4.1.3	L'état d'esprit - Technique de la Persona	203
4.1.4	Les moyens d'une session de TE	206
4.1.5	Session de TE avec Intervenants Externes	206
4.2	Session de Test exploratoire	208
4.3	Fin de Session de Test exploratoire	210
5.	Ingénierie des Tests de recette	210
5.1	Préparation de la recette	210
5.2	Cas des certifications et des tests réglementaires	211
6.	Stratégies de gestion des anomalies	214
6.1	Loi de Pareto	214
6.2	Taxonomie des erreurs	215
6.2.1	Quelques chiffres de l'industrie	215
6.2.2	Bugs ou parasites ?	216
6.2.3	Quels types de bugs ?	217
6.3	La dette technique	219
6.3.1	Mesurer la dette	220
6.3.2	Attitudes face à la dette	228
7.	Rôles	230
8.	Reporting	231
8.1	Généralités	231
8.2	Contenu d'un rapport	231

6 _____ Le test en mode Agile

9. Stratégies Données et Environnements de tests	232
9.1 Clonage des données de production.	233
9.2 Gestion des données de test	233
9.2.1 Génération des données.	233
9.2.2 Optimisation des tests par les valeurs	234
9.2.3 Génération aléatoire des valeurs nominales	235
9.2.4 Approche combinatoire aléatoire	236
9.2.5 Generative Adversarial Network - GAN.	236
9.2.6 Minimisation des données.	237
9.2.7 Anonymisation des données	238
9.2.8 Cas des données sujettes à la péremption	239
9.2.9 Dictionnaire de données	239
9.2.10 Isolation des données.	240
9.2.11 Synthèse	241
9.3 Reprise de données	243

Chapitre 5 Tests non fonctionnels

1. Introduction	247
2. Tests de sécurité.	248
2.1 Généralités sur les tests de sécurité	249
2.2 Orientations en termes de sécurité	251
2.3 Principes.	253
2.3.1 Séparation.	253
2.3.2 Contrôle	254
2.3.3 Étapes pour définir un test de sécurité	254
2.3.4 Technique de tests pour les OpSec	257
2.3.5 Types d'anomalies	262
2.3.6 Activités à contrôler.	266
2.4 Organisation des tests de sécurité	268
2.4.1 Modèle par jalons.	269
2.4.2 Correspondance entre sécurité et activités compatibles agile.	269
2.4.3 Vision SEAP	270
2.4.4 Approche par le risque	270
2.4.5 Modèle Microsoft SDL.	272

- 2.4.6 Modèle PASTA 273
- 2.4.7 Vision Coveros 274
- 2.4.8 Security Cards 275
- 2.5 Quelques techniques de tests 277
 - 2.5.1 Analyse de sécurité pour les exigences 277
 - 2.5.2 Analyse de failles de sécurité sur le design et le code 284
 - 2.5.3 Tests de pénétration 298
 - 2.5.4 Détection d'intrusion 305
 - 2.5.5 Intégration de la sécurité dans une approche DevOps 309
 - 2.5.6 Analyse de patch & applications sur la production 310
- 3. Tests d'utilisabilité 311
 - 3.1 Le test du Magicien d'Oz 312
 - 3.2 L'Experience Map 312
 - 3.3 Retours des utilisateurs en production 315
 - 3.4 Accessibilité 316
 - 3.4.1 Intérêt de l'accessibilité 316
 - 3.4.2 Comprendre les recommandations 320
 - 3.4.3 Certification et pénalités 321
- 4. Tests de charges 322
- 5. Tests de migration ou reprise de données 326
 - 5.1 Généralités sur les Migrations 326
 - 5.2 Migrations impliquant la reprise des données 329
 - 5.2.1 Classification des données 331
 - 5.2.2 Migration de type Big Bang 331
 - 5.2.3 Migration au fil de l'eau 332
 - 5.2.4 Outillage de la migration 334
 - 5.3 Phase finale de la migration - le décommissionnement 335
- 6. Tests de tolérance aux pannes 336
 - 6.1 Facteurs de défaillance 336
 - 6.1.1 L'Information 337
 - 6.1.2 La disponibilité 340
 - 6.1.3 La compensation 343
 - 6.1.4 La stabilisation 343
 - 6.2 Conception 344
 - 6.2.1 N+1 344
 - 6.2.2 Fail fast 344

8 _____ Le test en mode Agile

6.2.3 Paxos	344
6.2.4 Raft	346
6.3 Reprise de fonctionnement	347
6.3.1 Définition des tests de reprise de fonctionnement	348
6.3.2 Préparation des tests de reprise de fonctionnement	348
6.3.3 Exécution des tests de reprise de fonctionnement.	349
7. Tests d'exploitabilité	350
8. Tests de traçabilité.	351
8.1 Périmètre	351
8.2 La vérification	354
9. Tests de sûreté de fonctionnement	354

Chapitre 6 Versant technique du test

1. Introduction	359
2. Test de conception et de code	359
2.1 Préparation des tests pour la conception et le code	359
2.1.1 Responsabilité collective	360
2.1.2 Préparation de la revue de conception et de code	361
2.1.3 Tests Unitaires	377
2.1.4 Pair programming	404
2.2 Revue de conception et de code	406
2.2.1 Inspection	406
2.2.2 Méthode B	408
2.3 Software Craftsmanship	410
3. Ingénierie des tests d'intégration	414
3.1 Approche Big Bang	416
3.2 Approche Top-down.	417
3.3 Approche Bottom-up	419
3.4 Approche Sandwich	421
3.5 Autres approches.	422
3.5.1 Intégration par Collaboration	422
3.5.2 Intégration au Backbone	422
3.5.3 Intégration à fréquence élevée.	422

3.5.4	Intégration de Services Distribués	422
3.5.5	Intégration Continue	424
4.	Stratégies d'automatisation	425
4.1	Difficultés d'automatisation des tests.	426
4.2	Plateformes techniques de test	426
4.2.1	Application web.	427
4.2.2	Automatisation des tests pour applications web	428
4.2.3	Automatisation des tests pour navigateurs graphiques pour ordinateur personnel	429
4.2.4	Automatisation des tests pour navigateurs textuels.	430
4.2.5	Application mobile	431
4.2.6	Des plateformes au secours du problème de portabilité	436
4.2.7	Application native pour ordinateur personnel.	439
4.3	Le cornet de glace des tests automatisés	440
4.4	Élaboration des scripts de tests	441
4.4.1	Automatisation des tests de composants.	442
4.4.2	Automatisation des tests d'intégration	443
4.4.3	Automatisation des tests fonctionnels.	443
4.4.4	Compétences de programmation	444
4.4.5	Approche par mots-clés.	446
4.4.6	Model-Based Testing - MBT	448
4.4.7	Priorisation pour l'automatisation	452
4.4.8	Qui automatise ?	454
4.5	Répétabilité des scripts	456
4.6	Robustesse des scripts	457
4.6.1	Concept de Jidoka	457
4.6.2	Insensibilité aux bruits - Objectif du script	458
4.6.3	Les Faux-Positifs	458
4.7	Maintenabilité des scripts	459
4.7.1	Design pattern Page Object	461
4.7.2	Objets métier	462
4.7.3	Design pattern ScreenPlay	463
4.7.4	Framework des tests automatiques	464
4.8	Exécution des scripts	465
4.8.1	Exécution en aveugle.	466
4.8.2	Reporting d'exécution.	466

10 _____ Le test en mode Agile

5. Environnement technique du test	467
5.1 Alpha Testing et Beta Testing	467
5.2 Projet DevOps	469
5.2.1 Culture et généralités	469
5.2.2 Processus d'Intégration Continue (PIC)	474
5.2.3 Observabilité	490
5.2.4 Patterns & antipatterns organisationnels	490
5.2.5 Feature Flipping	495
5.2.6 Notion de Conteneur	497
5.2.7 Stratégie Shift Right	506
5.2.8 Outils	511

Chapitre 7

Facteurs de succès

1. Introduction	513
2. Le Contexte organisationnel	513
2.1 Le Management 3.0.	515
2.2 Holocratie	518
2.3 Et quoi d'autre ?	518
3. Vue globalo-locale	521
3.1 Point de vue de la régression.	521
3.2 Lien entre les anomalies	522
3.3 Point de vue de l'humeur	522
3.4 En résumé	523
4. Culture générale	524
4.1 Gestion de la connaissance comme enjeux du test	524
4.2 Vouloir apprendre	526
4.3 Repérage de la connaissance	528
4.4 Actualisation de la connaissance	529
4.5 Préservation de la connaissance	529
4.5.1 La Solution Intent	530
4.5.2 La CMDB d'ITIL	531
4.5.3 Généralités sur les outils	532

- 4.6 Valorisation de la connaissance 536
 - 4.6.1 Mise en avant d'un document 536
 - 4.6.2 Mesure de la compétence acquise 536
- 4.7 Partage de la connaissance 539
 - 4.7.1 Généralités sur la documentation 540
 - 4.7.2 Modélisation Agile 542
 - 4.7.3 État d'esprit du modélisateur Agile 551
 - 4.7.4 La notion de « Ba - » 551
- 5. Ce qui fait l'âme du Testeur 554
 - 5.1 Ceux qui testent 554
 - 5.1.1 Le Pragmatique 556
 - 5.1.2 Le Joueur 556
 - 5.1.3 Le Méthodique 557
 - 5.1.4 L'Empirique 557
 - 5.1.5 Le relationnel 557
 - 5.1.6 Changement des mentalités 561
 - 5.2 Passion et motivation intrinsèque 566
 - 5.3 La confiance 567
 - 5.3.1 La confiance en une version 567
 - 5.3.2 La confiance comme attribut du Leadership 568
 - 5.4 Biais cognitifs et test 569
 - 5.4.1 L'intuition du Testeur 571
 - 5.4.2 Biais de confirmation 572
 - 5.4.3 Dissonance cognitive chez le Testeur 573
 - 5.4.4 Effet de cadrage 574
 - 5.4.5 Corrélation illusoire 575
 - 5.4.6 Raisonnement fallacieux 575
 - 5.4.7 Illusions cognitives 577
 - 5.4.8 Identification de motifs à partir du vide 579
 - 5.5 Pensée critique 585
 - 5.6 Créativité 589
 - 5.7 Le Leadership 592
 - 5.8 Les valeurs du Testeur Agile 594

12 _____ Le test en mode Agile

Annexes

Bibliographie.....	597
Index.....	629

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

Préface

Avant-propos

1. Pourquoi ce livre ?	11
2. À propos du titre de ce livre	13
3. Pour qui est ce livre ?	16
4. Qui suis-je ?	17
5. Comment lire ce livre	18
5.1 La revue Jackie Brown du test agile à l'échelle	18
5.2 Modes de lecture	18
5.3 Avertissement	19

Chapitre 1 Pantesting

1. Vue d'ensemble	21
2. Testabilité	22
3. Panarchie	29
3.1 Le concept	29
3.2 Panarchie et panarchie	31
3.3 Panarchie et résilience	33
4. Théorie des contraintes - TdC	34
5. Double boucle d'apprentissage	37

2 _____ Conduite de tests agiles

pour SAFe et LeSS

6.	Pantesting en entreprise	42
6.1	Atelier « Panarchy »	42
6.2	Panarchie et tests	44

Chapitre 2 Pantesting appliqué au métier

1.	Introduction	47
2.	Testabilité des fonctionnalités	47
2.1	Capture du besoin et génération des idées	47
2.1.1	Lean Canvas et compagnie	50
2.1.2	25/10 Crowdsourcing pour dégager des idées d'un groupe	53
2.1.3	1-2-4-Tous	53
2.1.4	EventStorming	53
2.1.5	User Story Mapping	55
2.1.6	Autres moyens	56
2.2	Affinage des PBI	57
2.3	User Stories - US	62
2.3.1	Rédaction des US	62
2.3.2	Vision de SAFe	62
2.3.3	Vision de LeSS	64
2.3.4	US et spécifications exécutables	65
2.3.5	Estimation des US	68
2.4	Macro-US	72
2.4.1	Préparation des macro-US	72
2.4.2	Estimation des macros US	74
2.5	Enablers	76
3.	Testabilité des exigences non fonctionnelles	78
4.	Documentation liée à la solution	81
4.1	Modélisation	82
4.1.1	Solution Intent de SAFe	85
4.1.2	CMDB d'ITIL	87
4.1.3	Living documentation - la documentation c'est le code	88
4.2	Pantesting lié à la documentation	90
5.	DoD à l'échelle	92

6. DoR à l'échelle 94
7. Résumé..... 95

Chapitre 3

Pantesting appliqué à la testabilité technique

1. Architecture agile..... 99
1.1 Principes d'architecture 102
1.2 Autres visions sur les principes d'architecture agiles 109
1.3 Adaptation TOGAF à SAFe 111
1.4 Vision architecturale..... 112
1.5 Conception et partage 113
2. Domain-Driven Design - DDD..... 121
2.1 Techniques de compréhension et de communication du domaine .. 124
2.2 Techniques de préservation de l'intégrité du modèle..... 126
2.3 Conception dirigée par le modèle 133
2.4 Processus de modélisation 143
2.5 Exemple de combinaison SAFe-TOGAF-DDD..... 145
3. Model-Based System Engineering - MBSE..... 147
3.1 Présentation du MBSE 147
3.2 MBSE et cycle de vie..... 150
3.3 Du MBSE vers le PBSE 153
3.4 MBSE et les tests..... 154
3.4.1 Model-Based Testing - MBT..... 154
3.4.2 MBSA 159
3.4.3 Pantesting et MBSE..... 161
4. DevOps..... 164
4.1 Visions sur DevOps 164
4.1.1 Vision de Kim..... 164
4.1.2 Vision de SAFe..... 165
4.1.3 Vision de Spotify 166
4.1.4 Vision de Barclays Bank 166
4.1.5 Vision de Google - SRE 167
4.2 Le pipeline 170

4 --- Conduite de tests agiles

pour SAFe et LeSS

4.3	Continuous Testing - tester continuellement.	189
4.3.1	État d'esprit	193
4.3.2	Le CT c'est où on peut, dès qu'on peut !	194
4.3.3	Le CT, ce n'est pas que l'automatisation	196
4.3.4	Automatiser en continu c'est apprendre en continu	197
4.3.5	Automatisation et Craftsmanship dans l'équipe	198
4.3.6	Automatisation et autonomie.	199
4.3.7	Le temps d'exécution	202
4.3.8	L'environnement d'exécution	205
4.3.9	La portée du CT	209
4.3.10	Un banc de test idéal (ou presque)	211
4.3.11	Module de planification.	212
4.3.12	Manager de tests	214
4.3.13	Dictionnaire de données	215
4.3.14	Tableau de bord	216
4.3.15	Interface de pilotage logicielle	218
4.3.16	Interface de pilotage matérielle	218
4.3.17	Interface de pilotage d'outils externes	222
4.3.18	Interface de pilotage du banc	223
4.3.19	Solutions sur étagère	224
4.4	Infrastructure	229
4.5	Transformation DevOps	232
4.6	Mesure de la maturité DevOps	237
4.6.1	Évaluation DevOps selon sa vision	237
4.6.2	Évaluation DevOps selon SAFe.	238
4.6.3	Évaluation DevOps selon Humble	241
4.6.4	Métamodèle de maturité.	243
4.6.5	Évolution du pipeline.	244
4.6.6	Évolution des équipes	246
4.6.7	Définition de la notion de maturité	250
4.6.8	Évaluation de la maturité	252
4.7	Pantesting et DevOps	254
5.	Du code propre avec le Software Craftsmanship	257
6.	Architecture Decision Record - une pratique autour de la prise de décision architecturale	265
7.	Résumé	266

Chapitre 4
Pantesting appliqué au cycle de développement

- 1. Introduction 269
- 2. Rôle de la vision 270
- 3. Cycles de développements de solutions Lean 272
 - 3.1 Design Thinking 274
 - 3.2 Lean Startup 278
 - 3.3 Lean UX 280
 - 3.3.1 Identifier les hypothèses 283
 - 3.3.2 Créer un MVP 283
 - 3.3.3 Expérimenter 289
 - 3.3.4 Feedback et analyse 290
 - 3.3.5 Lean UX à la sauce SAFe 291
 - 3.4 Cycles de développements de la solution et Pantesting 292
- 4. Value Stream (courant de valeur) 294
- 5. Planification à l'échelle 299
 - 5.1 Contexte de la solution 299
 - 5.2 Préparation du Backlog 302
 - 5.3 PI Planning de SAFe 304
 - 5.3.1 Préparation du meeting 305
 - 5.3.2 Déroulé du meeting 307
 - 5.3.3 Sessions de travail sur les fonctionnalités 308
 - 5.3.4 Identification des risques sur le PI 309
 - 5.3.5 Vote de confiance 310
 - 5.3.6 Rétrospective 311
 - 5.3.7 Program Board 312
 - 5.4 Alternative au PI Planning avec LeSS 314
 - 5.5 Planification et Pantesting 316
 - 5.6 Roadmap de la solution 320
- 6. Mêlée quotidienne 321
- 7. Acceptation des PBI et de la solution 322
 - 7.1 Tests pendant le sprint 323
 - 7.1.1 Commencer par les tests - la qualité intégrée à la conception 323
 - 7.1.2 Cas des ENF 326
 - 7.1.3 Tests induits par les dépendances à la réalisation 328

6 --- Conduite de tests agiles

pour SAFe et LeSS

7.1.4	Qui teste ?	331
7.2	Revue d'itération	333
8.	Synchronisation des développements	337
8.1	Flux tirés	337
8.2	Modèle VATI de la théorie des contraintes	340
8.3	Outils de synchronisation de la théorie des contraintes	345
8.3.1	Le tambour	346
8.3.2	Le tampon	349
8.3.3	La corde	350
9.	Rétrospective à tous les niveaux	351
9.1	Meeting I&A de SAFe	352
9.2	Point de vue LeSS	353
9.3	Pratiques et principes sous-jacents	355
10.	Résumé	357

Chapitre 5 Pantesting appliqué au support

11.	Introduction	359
12.	Rôles	359
12.1	Les équipes de composants et de fonctionnalités	359
12.2	Les communautés	369
12.3	Le management agile	382
12.3.1	Vision historique du management	383
12.3.2	Raisons d'une transformation inéluctable	387
12.3.3	Propositions existantes	390
12.3.4	Les écocycles du management agile et Pantesting	414
12.3.5	Spécificités du manager de test agile	431
12.4	Le client	436
12.5	Le fournisseur	438
12.5.1	Sélection des fournisseurs	439
12.5.2	Contractualisation agile	443
12.5.3	Suivi du fournisseur	452
12.6	Les équipes distantes	453
12.6.1	Types de distance	453
12.6.2	ChatOps	454

- 12.6.3 Synchronisation 460
- 12.6.4 Organisation des équipes 462
- 12.6.5 Indépendance et séparation
 - entre les développeurs et les testeurs 464
- 12.6.6 "Translocation" des équipes et des compétences 466
- 12.6.7 Rôle de l'empathie dans le Pantesting 471
- 12.7 Les Ressources humaines 472
 - 12.7.1 Généralités sur les RH 472
 - 12.7.2 RH et agilité 474
 - 12.7.3 RH et Pantesting 484
- 13. Budget au niveau de l'entreprise 486
 - 13.1 Budget traditionnel 487
 - 13.2 Budget agile 489
 - 13.3 Période de budgétisation 490
 - 13.4 Flux des idées 491
 - 13.5 Équipe de budgétisation 492
 - 13.6 Priorisation des idées 493
 - 13.7 Suivi du budget 494
 - 13.8 Budget et Pantesting 496
- 14. Outils et automatisation 498
- 15. Lean QMS 502
 - 15.1 Description du Lean QMS 504
 - 15.2 Transformation d'un système qualité existant 517
 - 15.3 OKR pour des métriques Lean 521
 - 15.4 Lean QMS et Pantesting 524
- 16. Gestion de la connaissance 526
 - 16.1 Stratégies de gestion de la connaissance 526
 - 16.2 Focus sur la transmission de la connaissance 529
 - 16.3 Connaissance tacite et Ba 530
 - 16.4 Format de la connaissance tangible 534
 - 16.5 Pantesting & gestion de la connaissance 537
- 17. Culture d'entreprise 541
 - 17.1 Généralités sur la culture d'entreprise 541
 - 17.2 Pièges des organisations 543
 - 17.3 Propositions pour lutter contre ces pièges 546
- 18. Résumé 549

8 _____ Conduite de tests agiles

pour SAFe et LeSS

Chapitre 6

Mise en place du Pantesting

1. Introduction	551
2. Démarrer	551
2.1 Vision pour le changement	551
2.2 Roadmap	552
2.3 Pièges de la roadmap	554
3. Maintenir et améliorer	559
3.1 Diversifier les écocycles et les prioriser	559
3.2 Ajouter de la testabilité	560
3.3 Gérer le Pantesting	561
3.3.1 ASCCR de Vance	561
3.3.2 Double boucle d'apprentissage manageriale d'Augustine	561
3.3.3 Niveaux de connexion inter-écocycle	562
3.4 Mettre en musique le Pantesting	562
3.4.1 Cartographie	562
3.4.2 Rôles dans le Pantesting	564
3.4.3 Métriques	565

Chapitre 7

Annexes

1. Quelques abréviations	567
2. Valeurs et principes du manifeste agile et principes du test	570
3. Pratiques agiles	571
4. Basiqes du test agile	572
4.1 Principes du test	572
4.2 Vision du test agile	572
4.3 Quatre quadrants du test agile	573
4.4 Valeurs du test agile	574
4.5 Manifeste du test agile	574
4.6 État d'esprit du Testeur agile	575
5. Techniques de Test	575

6. Pratiques Lean	576
6.1 Challenge « 30 jours sur le Lean »	576
6.2 Lean Software Development - les principes Lean pour le logiciel ...	578
6.3 Kaizen	580
6.3.1 Les vingt règles Kaizen de Kobayashi	581
6.3.2 Les dix règles Kaizen de Dobbs	582
6.3.3 Les douze défis Kaizen pour le test classique	582
6.4 Kanban	583
7. SAFe vu d'en haut	585
7.1 Pourquoi SAFe est-il le plus utilisé ?	585
7.2 Les valeurs de SAFe	585
7.3 Les dix principes de SAFe	586
7.4 Description sommaire et spécificités	588
7.4.1 Couches	588
7.4.2 Meetings clés de SAFe	589
7.4.3 Le PI Planning	589
7.4.4 ART Sync : PO Sync et Scrum of Scrum	590
7.4.5 Éléments d'idéation	591
7.4.6 Rôles	593
7.4.7 Estimations WSJF	593
7.4.8 Story Points - la vision SAFe	594
8. LeSS	595
8.1 Smaller LeSS	597
8.2 Huge LeSS	598
9. Comparatif de modèles d'agile à l'échelle	599
10. Crédits	601
11. Références bibliographiques	604
Index	653