

Avant-propos

1. Le jour où tout a commencé 9
2. Pourquoi ce livre ? 10

Chapitre 1

Introduction aux tests

1. Définition des tests 13
2. Objectif des tests 14
3. Niveaux et types de tests 15
 - 3.1 Définition 15
 - 3.2 Types de tests non fonctionnels 17
 - 3.3 Phase de tests de composants 18
 - 3.3.1 Objectif 18
 - 3.3.2 Activités 18
 - 3.4 Phase de tests d'intégration 20
 - 3.4.1 Objectif 20
 - 3.4.2 Activités 20
 - 3.5 Phase de validation : tests système 24
 - 3.5.1 Objectif 24
 - 3.5.2 Activités 24
 - 3.6 Phase de tests d'acceptation 26
 - 3.6.1 Objectif 26
 - 3.6.2 Activités 27
4. Les tests dans le cycle de développement en V 28
 - 4.1 Définition 28
 - 4.2 Les étapes du cycle en V 29
 - 4.2.1 Définition des exigences 29
 - 4.2.2 Conception fonctionnelle 30
 - 4.2.3 Conception technique 30
 - 4.2.4 Conception technique détaillée 30
 - 4.2.5 Réalisation (codage) 31

2 — Les tests logiciels

Processus et outils pour des logiciels de qualité

4.3	Processus de test versus processus de développement	31
4.3.1	Préparation	32
4.3.2	Planification	33
4.3.3	Conception	34
4.3.4	Développement	34
4.3.5	Test	35
4.3.6	Déploiement	36
5.	Les tests de non-régression	37
5.1	Objectif	37
5.2	Principes et bonnes pratiques des tests de non-régression	38
5.3	Organisation de la campagne de tests	39

Chapitre 2

Organisation et implémentation des tests

1.	Stratégie de test	41
1.1	Objectif	41
1.2	Approche générale	43
2.	Périmètre de test	44
3.	Gestion des risques	46
3.1	Risques projet	46
3.2	Risques produit	48
3.3	Évaluation d'un risque produit	49
4.	Estimation des tests	51
4.1	Objectifs	51
4.2	Approche par ratio	52
4.2.1	Ratio développement/test	53
4.2.2	Ratio préparation des tests/exécution des tests	53
4.2.3	Ratio taux d'échec	55
4.2.4	Ratio taux de rejeu	58

4.3	Approche par unité d'œuvre.	58
4.3.1	Approche par cas de tests.	59
4.3.2	Approche dans le cadre d'une évolution majeure.	60
5.	Environnements de test	61
5.1	Types d'environnements de test	61
5.2	Configuration et gestion des environnements de test	62
6.	Données de test	63
6.1	Introduction	63
6.2	Principes des données de test.	63
6.2.1	Représentativité	63
6.2.2	Volumétrie	64
6.2.3	Variabilité	65
6.2.4	Confidentialité	66
6.3	Types de données de test	68
6.3.1	Données d'entrée	68
6.3.2	Sources de données de tests	71
6.3.3	Automatisation et intégration des données de test	71
7.	Outils de tests.	71
7.1	Objectifs	71
7.2	Outils de tests : gestion des exigences.	73
7.3	Outils de tests : gestion des tests.	74
7.4	Outils de tests : gestion des anomalies	74
7.5	Outils de tests : management	74
8.	Critères d'acceptation métier.	75
9.	Rôles et responsabilités	76
9.1	Objectifs	76
9.2	Organisation d'une équipe de test	77
9.2.1	Chef de projet test	78
9.2.2	Équipe de test.	79
9.2.3	Le testeur automatique	80

4 Les tests logiciels

Processus et outils pour des logiciels de qualité

10. Suivi des tests	81
10.1 Objectif	81
10.2 Indicateurs de suivi de test.	81
11. Implémentation des tests.	83
11.1 Gestion des exigences.	83
11.2 Conception et implémentation des tests	85
11.2.1 Conception des tests	85
11.2.2 Implémentation des tests.	86

Chapitre 3

Exécution des tests

1. Objectif	89
2. Campagne de tests	90
3. Gestion des anomalies	91
3.1 Objectifs	91
3.2 Processus de traitement des anomalies	92
3.2.1 Détection de l'anomalie	93
3.2.2 Assignment de l'anomalie.	97
3.2.3 Traitement de l'anomalie.	97
3.2.4 Validation du correctif de l'anomalie	98
3.2.5 Rejet de l'anomalie	99
3.2.6 Fermeture de l'anomalie	100
3.2.7 Pilotage du traitement des anomalies	100
3.3 Anomalies de production	102
4. Rapport de tests	103
5. Bilan de test.	104
5.1 Définition	104
5.2 Méthode de starfish	105
5.3 Mise en œuvre d'une starfish retrospective	106

Chapitre 4

Recette utilisateur

1. Objectif	109
2. Planification de la recette utilisateur.	110
3. Préparation de la recette utilisateur.	110
4. Exécution de la recette utilisateur	111
5. Gestion des anomalies métiers.	112
5.1 Qualification d'une anomalie métier.	112
5.2 Traitement et correction d'une anomalie métier	114
6. Suivi et reporting	115
7. Clôture de la phase de recette métier	115
8. Amélioration continue	116

Chapitre 5

Les tests dans un modèle agile

1. Principes de l'agilité.	117
1.1 Pourquoi l'agilité ?	117
1.1.1 Les valeurs du manifeste agile	118
1.1.2 Les principes du Manifeste agile	121
1.2 Scrum	130
1.2.1 Caractéristiques de Scrum	130
1.2.2 Les principaux rôles	132
1.2.3 Cérémonies agiles	134
2. Stratégie de test en mode agile	136
2.1 La gestion des exigences en mode agile.	136
2.1.1 User story	137
2.1.2 Epic	139
2.1.3 Thème	139

6 Les tests logiciels

Processus et outils pour des logiciels de qualité

2.2	La gestion des tests en mode agile	140
2.2.1	Introduction	140
2.2.2	Planification des tests en mode agile	141
2.2.3	Les types de test en mode agile	142
2.2.4	Préparation des tests en mode agile	143
2.2.5	Exécution des tests en mode agile	145
2.2.6	Les tests métier dans un projet en mode agile	146
2.3	Les critères de qualité d'une user story	147
2.3.1	Les critères de qualité du DoR (Definition of Ready) . .	147
2.3.2	Les critères de qualité du DoD (Definition of Done) . .	148
2.4	Principe du TDD et du BDD	149
2.4.1	TDD (Test-Driven Development)	149
2.4.2	ATDD (Acceptance Test-Driven Development)	151
2.4.3	BDD (Behavior-Driven Development)	152
3.	Introduction à SAFe	153
3.1	Les principes de SAFe	154
3.2	L'organisation de SAFe	157
3.2.1	Les niveaux	157
3.2.2	Les événements dans SAFe	158
3.2.3	Les rôles clés dans SAFe	159
4.	Introduction à l'approche DevOps	160
4.1	Définition	160
4.2	Les pratiques et outils DevOps	161

Chapitre 6

Introduction à l'automatisation des tests

1.	Objectifs	163
2.	La démarche de l'automatisation	165
2.1	Principes	165
2.2	L'approche de l'automatisation des tests	167

3. Les critères d'éligibilité à l'automatisation des tests	168
4. Le ROI (retour sur investissement)	169
4.1 Calcul du ROI de l'automatisation des tests	170
4.2 Quand l'automatisation des tests devient-elle rentable ?	172
4.3 Préconisations pour augmenter le ROI	172
5. Outils d'automatisation des tests	173

Chapitre 7

Outil HP ALM pour la gestion des tests

1. Présentation	175
2. Gestion des accès HP ALM.	176
3. Gestion des tests avec les modules de HP ALM	178
3.1 Gérer les versions applicatives et les campagnes de tests : module Management.	178
3.2 Gérer l'arbre des exigences du projet : module Requirements	182
3.3 Gérer l'arbre des cas de tests et des campagnes de tests : module Testing	186
3.3.1 Gérer l'arbre des cas de tests : module Test Plan	187
3.3.2 Gérer l'arbre des campagnes de tests : module Test Lab	190
3.3.3 Lancer l'exécution d'un test depuis le module Test Lab	194
4. Gestion des anomalies dans HP ALM	196
4.1 Création d'une anomalie : module Defects.	196
4.2 Cycle de vie d'une anomalie dans HP ALM	203
4.3 Personnalisation des champs dans HP ALM.	204
4.4 Recherche et filtres	205
4.5 Alertes et notifications	205
4.6 Intégration avec d'autres outils	206

8 _____ Les tests logiciels

Processus et outils pour des logiciels de qualité

Chapitre 8

Gestion des tests avec Jira

1. Présentation	207
2. Gestion des tests	208
2.1 Création des cas de tests	208
2.2 Organisation des tests avec un ticket Test Set	214
2.3 Exécution des tests avec un ticket de type Test Execution	216
2.4 Planification des tests avec un ticket de type test Plan dans Xray	220
3. Gestion des anomalies	223
3.1 Propriétés d'une anomalie	223
3.2 Cycle de vie d'une anomalie	226
 Conclusion	 231
Index	233

Préfaces

Avant-propos

1. Une transformation digitale	15
2. Pourquoi ce livre ?	16
3. À qui est destiné ce livre ?	16
3.1 Cas n°1 - C'était mieux avant !	16
3.2 Cas n°2 - Comment puis-je améliorer mon agilité dans mes tests ? . .	17
3.3 Cas n°3 - Mon équipe veut passer au DevOps, quels sont mes besoins de tests ?	18
3.4 Cas n°4 - Je suis dans une équipe agile en « Fast IT » et tout le monde code et teste	18
4. Comment lire ce livre ?	18
5. Qui suis-je ?	21

Chapitre 1

Quelques rappels

1. Introduction	23
2. Les quatre valeurs de l'agile - Rappel du manifeste agile	24
3. Les sept principes du test - Rappel ISTQB	25
3.1 Principe du test #1 - Les tests montrent la présence de défauts, et non l'absence de défauts	25
3.2 Principe du test #2 - Le test exhaustif est impossible	26
3.3 Principe du test #3 - Tester tôt économise du temps et de l'argent . .	26
3.4 Principe du test #4 - Regroupement des défauts	28
3.5 Principe du test #5 - Usure des tests (ex. Paradoxe du pesticide) . . .	28
3.6 Principe du test #6 - Le test dépend du contexte	30
3.7 Principe du test #7 - Illusion de l'absence de défaut	31
4. Techniques de base du test	33
4.1 Partitionnement de classes d'équivalences et valeurs limites	33
4.2 Table de décision	36
4.3 Transitions d'états	40

2 _____ Le test en mode Agile

4.4	Heuristiques de test	41
4.4.1	CORRECT	42
4.4.2	Heuristique - Test n-wise	44
5.	Le cycle de vie logiciel	46
5.1	Projet à l'échelle d'une seule équipe Scrum	46
5.1.1	Scrum en quelques lignes.	49
5.1.2	Sprint Refinement Meeting (ou grooming)	52
5.1.3	Sprint Planning	55
5.1.4	Daily Scrum Meeting.	60
5.1.5	Sprint Review.	65
5.1.6	Rétrospective	68
5.2	Projet à l'échelle de plusieurs équipes	85
5.2.1	Cas de frameworks agiles existants	87
5.2.2	Point de vue du modèle SAFe	88
5.2.3	Point de vue du modèle Spotify	96
5.2.4	Point de vue LeSS	102
5.2.5	Point de vue « Disciplined Agile »	105
5.2.6	Point de vue Scrum@Scale - S@S.	108
5.2.7	Point de vue de Nexus	112
5.3	Développer sa propre vision de l'agilité à l'échelle	113
5.3.1	Approche par composants vs fonctionnalités	113
5.3.2	Structuration des équipes par rapport à la loi de Conway	118
5.3.3	Équipes « Domain-Driven Design »	119
5.3.4	Organiser l'agilité et le test à l'échelle.	121

Chapitre 2 L'état d'esprit

1.	Introduction - Exemple de l'industrie du textile	123
1.1	Contrôle des ballots	125
1.2	Contrôle pendant le tissage	125
1.3	Amélioration du procédé de tissage	126
1.4	Le test agile	127

2.	État d'esprit lean	127
2.1	Le lean, les origines de l'agile	127
2.2	Les pratiques lean	130
2.2.1	Juste-à-temps - « Just in time »	130
2.2.2	Jidoka - autonomatisation.	132
2.2.3	Heijunka – flux réguliers	133
2.2.4	Standardisation	135
2.2.5	5S – rangez !	138
2.2.6	Kaizen et l'amélioration continue	139
3.	Manifeste du test agile selon Samantha Laing et Karen Greaves.	152
3.1	« Tester pendant le sprint »	153
3.2	« Prévenir les bugs »	154
3.3	« Comprendre ce que l'on doit tester »	156
3.4	« Concevoir un meilleur système »	157
3.5	« L'équipe est responsable de la qualité du produit »	159
4.	Autres visions du test agile	168

Chapitre 3 Versant métier

1.	Introduction.	171
2.	Quelques techniques d'idéation	171
2.1	Design Thinking	172
2.2	Lean Start-up.	175
2.3	Event Storming.	178
3.	Idéation et spécification des besoins	180
3.1	Backlog du portfolio	180
3.2	Temporalisation des demandes	181
3.3	La valeur métier	183
3.3.1	Modèle MoSCoW	184
3.3.2	Diagramme de Kano	185
3.3.3	Priorisations complexes (Weighted Shortest Job First - WSJF)	187
3.4	Qualité des demandes.	188
3.4.1	Définition du Prêt (Definition of Ready - DoR)	188
3.4.2	Définition du Terminé (DoD)	197

4 _____ Le test en mode Agile

3.4.3	DoR et DoD à l'échelle	199
3.4.4	Mise à jour des DoR et DoD	199
3.5	Description d'une US en mode BDD	199
3.6	Ingénierie des critères d'acceptation des US	209
3.7	Formalisation d'une epic chez SAFe	211
3.8	Découpage des éléments du backlog	213
3.9	Des spécifications traditionnelles en agile ?	214
3.10	Revue de définition des US.	217
3.10.1	Affinement des US.	217
3.10.2	Approche « Los Très Amigos » - les trois amis	217
3.10.3	Revue d'un élément plus gros qu'une US.	220
4.	Test des éléments.	220
4.1	Démarche ATDD pour développer une US.	220
4.2	Revue de réalisation de l'US	222
4.3	Tests pour les macrospecifications	222
4.4	Visite guidée	223

Chapitre 4 Versant industriel du test

1.	Introduction	225
2.	Stratégie de test	225
2.1	Vision académique de la stratégie de test	225
2.2	Testabilité d'une stratégie de test	228
2.3	Catégories de stratégies	229
2.4	Tester tôt.	232
2.5	Approche par le risque	234
2.5.1	Caractérisation du niveau de risque	235
2.5.2	Principe du Risk-Based Testing - RBT	238
2.5.3	Granularité	239
2.5.4	Niveau « données de tests »	240
2.5.5	Niveau « scénarios de tests »	240
2.5.6	Niveau « exigence de tests »	240
2.5.7	Niveau « campagnes de tests »	240
2.5.8	Périmètre vs Qualité et objectifs de test.	242
2.6	Heuristique pour une stratégie de test	246

2.7	Context-Driven Testing (CDT) - Test Piloté par le Contexte.	250
2.7.1	Valeur d'une pratique dépendante du contexte.	252
2.7.2	Les Best Practices	253
2.7.3	Des personnes qui travaillent ensemble.	253
2.7.4	Prédictibilité des projets	254
2.7.5	Résolution du problème	254
2.7.6	Le challenge intellectuel	254
2.7.7	Esprit critique.	254
3.	Tests systèmes / bout-en-bout	255
3.1	Stratégies de tests bout-en-bout	256
3.1.1	Approche « WaterScrumFall »	256
3.1.2	Approche itérative & incrémentale	258
3.1.3	Approche par le Program Board de SAFe	260
3.1.4	Approche « test par l'usage »	262
3.2	Organisation des tests	267
3.2.1	Vision jacobiniste du test – une fausse bonne idée ?.	267
3.2.2	Ingénierie des tests systèmes/bout-en-bout.	270
3.2.3	Structure dans un outil de gestion des tests	272
3.2.4	Planification	277
3.2.5	Monitoring et Contrôle.	280
3.2.6	Clôture des tests	280
3.3	Tests de régression	281
3.3.1	Campagne définie par l'impact.	281
3.3.2	Smoke tests – les tests fumigatoires.	282
3.3.3	Campagnes de régression prédéfinies par le risque	283
3.3.4	Campagnes prédéfinies par thème	285
3.3.5	Check-list	285
4.	Tests exploratoires.	287
4.1	Préparation des tests exploratoires	288
4.1.1	La charte.	288
4.1.2	La zone de recherche	290
4.1.3	L'état d'esprit - Technique de la persona	291
4.1.4	Les moyens d'une session de TE	294
4.2	Session de Test exploratoire.	296
4.3	Fin de Session de Test exploratoire	298

6 ————— Le test en mode Agile

5.	Ingénierie des tests de recette	299
5.1	Création des critères d'une recette	299
5.2	Application de la recette dans le projet	300
5.3	Cas des certifications et des tests réglementaires	300
5.4	Préparation de la recette	302
5.5	Passage de la recette	302
6.	Stratégies de gestion des anomalies	304
6.1	Taxonomie des erreurs	306
6.1.1	Origine des anomalies	306
6.1.2	Bugs ou parasites ?	307
6.1.3	Quels types de bugs ?	308
6.2	Loi de Pareto	309
6.3	Le « Bug cap » de Microsoft	310
6.4	Approche « zéro défaut »	311
6.5	Anticiper la création de bugs par des pratiques « agiles »	312
6.6	Anticiper la création de bugs par la dette technique	312
6.6.1	Prise de conscience	314
6.6.2	Mesurer la dette	316
6.6.3	Attitudes face à la dette	322
7.	Rôles	324
7.1	Rôle du Product Manager dans le test	324
7.2	Rôle du Product Owner dans le test	326
7.3	Rôle du Scrum Master dans le test	327
7.4	Rôle des développeurs dans le test	328
7.5	Rôle du test manager	328
7.5.1	Posture du TM	328
7.5.2	Organisation & planification	332
7.5.3	La voix du client	333
7.6	Rôle du testeur	334
7.7	Autres rôles	338
7.8	Rôle de l'IA	339
8.	Reporting	341
8.1	Généralités	341
8.2	Contenu d'un rapport	342

9. Living Documentation	343
9.1 Généralités.....	343
9.2 Des tests vivants.....	347
9.3 Une documentation vivante mais stable.....	348
10. Stratégies sur les données et environnements de tests	351
10.1 Clonage des données de production	351
10.2 Gestion des données de test.....	352
10.2.1 Génération des données	352
10.2.2 Optimisation des tests par les valeurs	352
10.2.3 Génération des valeurs par perturbation	353
10.2.4 Approche combinatoire aléatoire	355
10.2.5 Generative Adversarial Network - GAN	355
10.2.6 Minimisation des données	356
10.2.7 Anonymisation des données.....	358
10.2.8 Cas des données sujettes à la péremption	358
10.2.9 Dictionnaire de données	359
10.2.10 Isolation des données	360
10.2.11 Synthèse	361
10.3 Reprise de données	364
11. Testing et DevOps.....	367
11.1 Culture et généralités.....	367
11.2 À quoi ressemble le test avec DevOps ?	370
11.3 Le Continuous Testing.....	373

Chapitre 5

Tests non fonctionnels

1. Introduction.....	375
2. Tests de sécurité	377
2.1 Généralités sur les tests de sécurité	378
2.2 Orientations en termes de sécurité	380
2.3 Principes	382
2.3.1 Séparation	382
2.3.2 Contrôle	383

8 ————— Le test en mode Agile

2.4	Étapes pour définir un test de sécurité	383
2.4.1	Technique de tests pour les OpSec	386
2.4.2	Types d'anomalies	391
2.4.3	Activités à contrôler	395
2.5	Organisation des tests de sécurité	398
2.5.1	Modèle par jalons	398
2.5.2	Correspondance entre sécurité et activités compatibles agile	398
2.5.3	Vision SEAP	399
2.5.4	Approche par le risque	399
2.5.5	Modèle Microsoft SDL	401
2.5.6	Modèle PASTA	402
2.5.7	Vision Coveros	403
2.5.8	Security Cards	404
2.6	Quelques techniques de tests	406
2.6.1	Analyse de sécurité pour les exigences	406
2.6.2	Analyse de failles de sécurité sur le design et le code	412
2.6.3	Tests de pénétration	425
2.6.4	Détection d'intrusion	433
2.6.5	DevSecOps - Intégration de la sécurité dans une approche DevOps	437
2.6.6	Analyse de patch et nouvelles versions	438
2.7	Rôles autour de la sécurité	440
3.	Tests d'utilisabilité	442
3.1	Questionnaire de satisfaction	446
3.2	UX Curves – les courbes d'expériences d'utilisateurs	447
3.3	Organisation des tests d'utilisabilité	450
3.3.1	Sessions de tests d'utilisabilité avec des utilisateurs	451
3.3.2	Retours des utilisateurs en production	455
3.3.3	Co-design	455
3.4	Accessibilité	458
3.4.1	Intérêt de l'accessibilité	458
3.4.2	Comprendre les recommandations	463
3.4.3	Certification et pénalités	464
4.	Tests de charges	465

5. Tests de migration ou reprise de données	469
5.1 Généralités sur les migrations	469
5.2 Migrations techniques	472
5.3 Migrations impliquant la reprise des données	474
5.3.1 Classification des données	476
5.3.2 Migration de type big bang.	477
5.3.3 Migration au fil de l'eau	477
5.3.4 Outillage de la migration.	478
5.4 Phase finale de la migration - le décommissionnement	480
6. Tests de tolérance aux pannes	480
6.1 Facteurs de défaillance	481
6.1.1 L'information.	482
6.1.2 La disponibilité.	484
6.1.3 La compensation	487
6.1.4 La stabilisation.	488
6.2 Conception	489
6.2.1 N+1	489
6.2.2 Fail fast.	489
6.2.3 Paxos.	489
6.2.4 RAFT	490
6.3 Reprise de fonctionnement	492
6.3.1 Définition des tests de reprise de fonctionnement	493
6.3.2 Préparation des tests de reprise de fonctionnement	493
6.3.3 Exécution des tests de reprise de fonctionnement	494
7. Tests d'exploitabilité.	495
8. Tests de traçabilité	497
8.1 Périmètre.	497
8.2 La vérification	500
9. Tests de sûreté de fonctionnement	501
10. Ecoconception	504
10.1 Pourquoi devrions-nous tester l'écoconception ?	504
10.2 Comment améliorer votre impact ?	510
10.2.1 Le Référentiel Général d'Ecoconception de Services Numériques - RGEN	511
10.2.2 Le Guide de Référence de conception responsable de services numériques - GR 491	513

10 _____ Le test en mode Agile

10.2.3 Le guide d'écoconception de services numériques	513
10.2.4 115 bonnes pratiques d'écoconception web	514
10.2.5 Manifeste pour une agilité responsable	514
11. Proposition d'organisation pour le test des ENF	515

Chapitre 6

Versant technique du test

1. Introduction	519
2. Le Software Craftsmanship (SC ou « Craft »)	520
3. Test de conception et de code	524
3.1 Prévenir les défauts dans le code	524
3.1.1 Responsabilité collective du code	524
3.1.2 Une bonne conception, plutôt qu'une bonne revue	531
3.1.3 Du bon code, plutôt qu'une bonne revue	536
3.1.4 Tests Unitaires (TU)	551
3.2 Revue de conception et de code	581
3.2.1 Inspection	581
3.2.2 Méthode B	583
4. Ingénierie des Tests d'Intégration (TI)	585
4.1 Tests d'interopérabilité	586
4.1.1 Approche big bang	586
4.1.2 Approche top-down	589
4.1.3 Approche bottom-up	589
4.1.4 Approche sandwich	590
4.1.5 Autres approches	591
4.2 Tests de portabilité	594
4.3 Tests de coexistence	594
5. Stratégie d'automatisation	595
5.1 Vision haut niveau de l'automatisation	597
5.1.1 Identification de la mesure	599
5.1.2 Modèle de mesure	599
5.1.3 Automatisation - la séparation homme / machine	603
5.1.4 Renforcement	606
5.2 Massification	611

5.3	Automatisation de scripts de tests fonctionnels	612
5.3.1	Pilotage d'interfaces vocales	613
5.3.2	Pilotage d'interfaces tactiles	614
5.3.3	Pilotage d'interfaces liées à l'OS - client lourd	615
5.3.4	Pilotage d'interfaces web	615
5.3.5	Pilotage d'interfaces « machine 2 machine »	616
5.4	Frameworks et plateformes de test	617
5.4.1	Agilitest	619
5.4.2	Appium	620
5.4.3	Cerberus	620
5.4.4	Cucumber	621
5.4.5	Cypress	622
5.4.6	Playwright	623
5.4.7	Robot Framework	624
5.4.8	Selenium	626
5.4.9	Katalon / Ranorex / Tosca / TestComplete / UFT(QTP)	626
5.4.10	UI Path	628
5.4.11	Autres catégories d'automatisation	628
5.5	Spécificités techniques des types d'applications	632
5.5.1	Automatisation des tests pour applications web	632
5.5.2	Application mobile	634
5.6	Conception des scripts	640
5.7	Framework des tests automatiques	641
5.7.1	Compétences de programmation	643
5.7.2	Patrons de conception	648
5.8	Exécution des scripts	651
5.8.1	Exécution en aveugle	651
5.8.2	Reporting d'exécution	652
5.9	Qui automatise ?	653
5.9.1	Équipe dédiée à l'automatisation	653
5.9.2	Des automaticiens dans l'équipe de développement	654
5.9.3	Les développeurs automatisent les tests	654
5.9.4	Et dans la vraie vie ?	654
6.	Environnement technique du test	656
6.1	Environnement d'intégration	656
6.1.1	Branching Model	656
6.1.2	Feature Flag (ou Toggle)	658

12 _____ Le test en mode Agile

6.1.3	Exécution périodique des tests	659
6.2	Environnement de déploiement	660
6.2.1	Installation du produit	660
6.2.2	Mise à disposition d'un environnement de déploiement	662
6.3	Opérations	671
6.3.1	Qualité de service	671
6.3.2	Continuité de service	673
6.3.3	Gestion des catastrophes	674
6.3.4	Zero Downtime Deployment - ZDD	674
6.3.5	Observabilité	676

Chapitre 7 Facteurs de succès

1.	Tester c'est apprendre	681
1.1	Gestion de la connaissance comme enjeux du test	682
1.2	Vouloir apprendre	685
1.3	Repérer la connaissance	686
1.4	Actualiser la connaissance	687
1.5	Préserver la connaissance	687
1.5.1	La Solution Intent	688
1.5.2	Modélisation agile	689
1.5.3	La CMDB d'ITIL	698
1.6	Partager la connaissance	699
1.6.1	Formats de transmission	701
1.6.2	Mentoring	713
2.	Le contexte organisationnel dans l'apprentissage	716
2.1	Bureaucratie agile	718
2.2	Réseau d'équipes agiles	719
2.3	Le management 3.0	720
2.4	Holocratie	722
2.5	Vue transverse sur ces structures	722
2.5.1	Points communs des organisations autonomes	724
2.5.2	La notion de « Ba »	725
2.5.3	Travailleur de la connaissance	728

3.	Ce qui fait l'âme du testeur	729
3.1	Ceux qui testent	729
3.1.1	Le pragmatique	731
3.1.2	Le joueur	731
3.1.3	Le méthodique	732
3.1.4	L'empirique	732
3.1.5	Le relationnel	732
3.1.6	Changement des mentalités	736
3.2	La confiance	742
3.2.1	La confiance en une version	742
3.2.2	La confiance comme attribut du leadership	743
3.3	La sclérose du test	744
3.3.1	La sclérose des anomalies	744
3.3.2	La sclérose administrative	749
3.3.3	La sclérose nombriliste : améliorer le test avant le produit	749
3.4	Pensée critique (PC)	750
3.4.1	Introduction à la pensée critique	751
3.4.2	Les grands penseurs de la pensée critique	752
3.4.3	La PC dans le monde actuel	758
3.4.4	La PC pour le test	759
3.4.5	Quelques outils pour la PC	760
3.4.6	Biais cognitifs	766
3.4.7	La bonne distance - Limites de la pensée critique	778
3.5	Créativité	779
3.6	Le leadership pour changer	781
3.7	Stratégies pour le changement	783
3.7.1	Une plateforme en feu	783
3.7.2	Le modèle ADKAR	784
3.7.3	Modèle de Lewin	785
3.8	Les valeurs du testeur agile	786

14 _____ Le test en mode Agile

Annexes

Bibliographie.....	789
Annexe	853
Index.....	859