

Table des matières

Remerciements	1
Avant-propos	3
Chapitre 1. Gestion des projets de test	5
1.1. Principes généraux	5
1.1.1. Qualité des exigences	6
1.1.2. Complétude des livraisons.	7
1.1.3. Disponibilité des environnements de test	8
1.1.4. Disponibilité des données de test.	8
1.1.5. Respect des livraisons et des plannings	9
1.1.6. Coordination et mise en place des environnements	10
1.1.7. Validation des prérequis (TRR).	11
1.1.8. Fourniture des jeux de données (JDD).	11
1.1.9. Décisions de <i>Go-NoGo</i> (TRB)	12
1.1.10. Livraisons et déploiements continus	12
1.1.10.1. Livraison continue	13
1.1.10.2. Tests continus	13
1.1.10.3. Déploiement continu	14
1.2. Suivi des projets de test.	14
1.3. Risques et systèmes de systèmes	14
1.4. Particularités liées aux systèmes de systèmes.	15
1.5. Particularités liées aux méthodologies	16
1.5.1. Définition des composants.	17
1.5.2. Activités de test et d'assurance qualité	17
1.6. Particularités liées aux équipes	17

Chapitre 2. Processus de test	19
2.1. Organisation	21
2.2. Planification	22
2.2.1. WBS projet et planification	23
2.3. Contrôle des activités de test.	25
2.4. Analyse	26
2.5. Conception	27
2.6. Implémentation	28
2.7. Exécution des tests	30
2.8. Évaluation.	31
2.9. <i>Reporting</i>	32
2.10. Clôture	33
2.11. Gestion de l'infrastructure	34
2.12. Revues	35
2.13. Adaptation des processus	36
2.14. Matrice RACI.	37
2.15. Automatisation des processus ou des tests	38
2.15.1. Automatiser ou industrialiser ?	38
2.15.2. Automatiser quoi ?	38
2.15.3. Choisir quoi automatiser	40
Chapitre 3. Amélioration continue des processus	41
3.1. Modéliser les améliorations	42
3.1.1. PDCA et IDEAL	42
3.1.2. CTP	43
3.1.3. SMART.	45
3.2. Pourquoi et comment s'améliorer ?.	45
3.3. Méthodes d'amélioration.	45
3.3.1. Référentiels externes et/ou internes	46
3.3.1.1. S'améliorer avec un référentiel externe.	46
3.3.1.2. S'améliorer avec un référentiel interne (CTP).	47
3.3.1.3. S'améliorer à partir des défauts (ODC).	47
3.4. Qualité des processus	50
3.4.1. <i>Fault seeding</i>	50
3.4.2. Statistiques	50
3.4.3. <i>A posteriori</i>	50
3.4.4. Éviter l'introduction de défauts	51
3.5. Efficacité des activités d'amélioration	51
3.6. Recommandations	53

Chapitre 4. Équipes de test, QA, QC ou IV&V	55
4.1. Nécessité d'une équipe de test.	56
4.2. Caractéristiques d'une bonne équipe de test.	57
4.3. Profil idéal d'une équipe de test.	59
4.4. Évaluation de l'équipe	60
4.4.1. Tableau d'évaluation des compétences	60
4.4.2. Composition	63
4.4.3. Sélectionner, embaucher et retenir	63
4.5. <i>Test manager</i>	64
4.5.1. Mener ou diriger ?	64
4.5.2. Évaluer et mesurer	65
4.5.3. Questions récurrentes pour les <i>test managers</i>	66
4.6. Analyste de test	68
4.7. Analyste technique de test	69
4.8. Automaticien de test	69
4.9. Technicien de test	70
4.10. Choisir ses testeurs.	70
4.11. Formation, certification ou expérience ?	71
4.12. Embaucher ou sous-traiter ?	72
4.12.1. Sous-traitance efficace	73
4.13. Organisation des équipes de test multiniveau	73
4.13.1. Conformité, stratégie et organisation.	74
4.13.2. Équipes de tests unitaires (TU/TC)	74
4.13.3. Équipe de tests d'intégration (TI)	75
4.13.4. Équipe de tests système (SYST)	75
4.13.5. Équipe de tests d'acceptation (UAT)	76
4.13.6. Équipes de tests techniques (TT)	76
4.14. Défis de l'internalisation et de l'externalisation.	77
4.14.1. Internalisation et colocalisation	77
4.14.2. Externalisation proche	78
4.14.3. Externalisation géographiquement distante	79
 Chapitre 5. Évaluation de la charge de test	 81
5.1. Difficulté d'évaluation	81
5.2. Techniques d'évaluation	82
5.2.1. Techniques basées sur l'expérience	82
5.2.2. Techniques basées sur les points de fonction ou TPA	83
5.2.3. Augmentation constante des exigences	86
5.2.4. Techniques basées sur des informations historiques	87

5.2.5. WBS ou TBS	88
5.2.6. Agilité, estimation et vélocité	88
5.2.7. Retroplanning	89
5.2.8. Ratio développeurs/testeurs	90
5.2.9. Éléments influençant l'estimation	90
5.3. Présentation de la charge de test	93
5.3.1. Vérifier et valider l'évaluation	93
5.3.2. Quelques valeurs	94
5.4. Comprendre la charge de test	95
5.4.1. Couverture des composants	95
5.4.1.1. Couverture statique	95
5.4.1.2. Couverture dynamique	95
5.4.2. Couverture des fonctionnalités	96
5.4.3. Couverture technique	96
5.4.3.1. Sécurité	96
5.4.3.2. Performances	96
5.4.4. Préparation des campagnes	97
5.4.5. Exécution des campagnes	97
5.4.6. Gestion des anomalies	98
5.5. Défendre la charge de test	98
5.6. <i>Multi-tasking</i> et <i>crunch</i>	99
5.7. Adapter et suivre la charge de test	100

Chapitre 6. Métriques, KPI et mesures 101

6.1. Choix des métriques	102
6.2. Précision des métriques	103
6.2.1. Cas particulier du coût des défauts	104
6.2.2. Cas particulier des anomalies	104
6.2.3. Précision ou ordre de grandeur ?	104
6.2.4. Fréquence de mesure	105
6.2.5. Utilisation des métriques	105
6.2.6. Amélioration continue des métriques	107
6.3. Métriques produit	107
6.3.1. <i>First Time Right</i> (FTR)	108
6.3.2. Taux de couverture	109
6.3.3. <i>Code churn</i> (changement du code)	110
6.4. Métriques de processus	111
6.4.1. Métriques d'efficacité	111
6.4.1.1. <i>Defect Detection Efficiency</i> (DDE)	111
6.4.1.2. <i>Defect Removal Efficiency</i> (DRE)	111

6.4.1.3. Couverture des risques (CRI)	112
6.4.1.4. Tests de régression automatisés (TRA)	113
6.4.1.5. Efficacité de conception TDD	113
6.4.2. Métriques d'efficacité	113
6.4.2.1. Défauts par FP	114
6.4.2.2. Autres métriques	114
6.5. Définition des métriques	115
6.5.1. Métriques du modèle qualité	116
6.5.1.1. Efficacité.	116
6.5.1.2. Complétude	117
6.5.1.3. Introduction de défauts	117
6.6. Validation des métriques et des mesures.	117
6.6.1. <i>Baseline</i>	118
6.6.2. Données historiques	118
6.6.3. Améliorations périodiques.	119
6.7. <i>Reporting</i> des mesures	120
6.7.1. <i>Reporting</i> interne aux tests	120
6.7.2. <i>Reporting</i> vers l'équipe de développement	121
6.7.3. <i>Reporting</i> vers le management	121
6.7.4. <i>Reporting</i> vers les clients ou <i>product owners</i>	122
6.7.5. <i>Reporting</i> vers la direction	124
Chapitre 7. Gestion des exigences.	125
7.1. Documents d'exigences	125
7.2. Qualités des exigences	126
7.3. Bonnes pratiques en gestion des exigences	128
7.3.1. Élicitation	128
7.3.2. Analyse	129
7.3.3. Spécification	129
7.3.4. Validation	130
7.3.5. Administration des exigences	130
7.3.6. Exigences et gestion des connaissances métier.	131
7.3.7. Exigences et gestion de projet	131
7.4. Niveaux d'exigences	132
7.5. Complétude des exigences	132
7.5.1. Gérer les TBD et les TBC	133
7.5.2. Éviter les incomplétudes.	133
7.6. Exigences et agilité	134
7.7. Problématiques associées aux exigences.	134

Chapitre 8. Gestion des anomalies.	137
8.1. Gestion des anomalies en MOA et en MOE.	137
8.1.1. Qu'est-ce qu'une anomalie ?	137
8.1.2. Anomalies en MOA	138
8.1.3. Anomalies en MOE.	138
8.2. <i>Workflow</i> de gestion des anomalies.	139
8.2.1. Exemple	139
8.2.2. Simplifier.	141
8.3. Réunion de triage	141
8.3.1. Priorité et sévérité des anomalies.	142
8.3.2. Détection d'anomalies	143
8.3.3. Correction et urgence.	143
8.3.3.1. Criticité des corrections	144
8.3.4. Respect des processus	144
8.4. Spécificités des TDD, ATDD et BDD	145
8.4.1. <i>Test Driven Development</i> (TDD).	145
8.4.2. ATDD et BDD	146
8.5. <i>Reporting</i> d'anomalies	147
8.5.1. Gestion du <i>backlog</i> des anomalies	148
8.6. Autres <i>reportings</i> utiles.	150
8.7. Ne pas oublier les petits défauts.	151
Chapitre 9. Gestion de configuration	153
9.1. Pourquoi gérer la configuration ?	153
9.2. Impact de la gestion de configuration	154
9.3. Composants.	155
9.4. Processus	156
9.5. Organisation et standards.	156
9.6. <i>Baseline</i> ou paliers, branches et fusions	157
9.6.1. Paliers	158
9.6.2. Branches	158
9.6.3. Fusion.	159
9.7. <i>Change Control Board</i> (CCB).	159
9.8. Fréquences des livraisons	160
9.9. Modularité	160
9.10. Gestion des versions.	162
9.11. Gestion des livraisons	162
9.11.1. Préparation de livraison	164
9.11.2. Validation de livraison	165
9.12. Gestion de configuration et déploiements	166

Chapitre 10. Automatisation et outils de test	169
10.1. Objectifs de l'automatisation des tests	169
10.1.1. Trouver plus de défauts.	170
10.1.2. Automatiser les tests dynamiques.	171
10.1.3. Trouver toutes les régressions	172
10.1.4. Exécuter les campagnes plus rapidement	173
10.2. Challenges de l'outillage des tests.	174
10.2.1. Positionnement de l'automatisation	174
10.2.2. Analyse des processus de test	174
10.2.3. Intégration des outils	175
10.2.4. Qualification des outils	176
10.2.5. Synchronisation des cas de test	177
10.2.6. Gestion des données de test	177
10.2.7. Traitement du <i>reporting</i> (confiance dans les tests automatisés).	178
10.3. Outiller quoi ?	178
10.4. Outillage des tests	179
10.4.1. Sélection des outils	180
10.4.2. Calcul du retour sur investissement (ROI)	182
10.4.3. Éviter l'abandon des outils et de l'automatisation	183
10.5. Stratégies de tests automatisés	183
10.6. Challenge de l'automatisation des tests de SoS	184
10.6.1. Maîtriser l'automatisation	185
10.6.2. Préparer l'automatisation.	186
10.6.3. Injection de défauts	187
10.7. Typologie d'outils et leurs défis spécifiques.	187
10.7.1. Outils de tests statiques <i>versus</i> tests dynamiques.	188
10.7.2. <i>Data Driven Testing</i> (DDT)	189
10.7.3. <i>Keyword Driven Testing</i> (KDT)	190
10.7.4. <i>Model Based Testing</i> (MBT)	190
10.8. Tests de (non-)régression automatisés	191
10.8.1. TNR lors des <i>builds</i>	192
10.8.2. TNR lors des changements d'environnement.	192
10.8.3. TNR de prévalidation, <i>health</i> ou <i>sanity checks</i> , <i>smoke tests</i>	193
10.8.4. Automatiser quoi ?	194
10.8.4.1. Automatisation des tests statiques	195
10.8.4.2. Automatisation des tests unitaires	195
10.8.4.3. Automatisation des tests fonctionnels	195
10.8.4.4. Automatisation des tests techniques	196
10.8.5. <i>Frameworks</i> de test	196
10.8.6. Cas de test E2E	198
10.8.7. Maintenance des cas de test automatisés ou non	199

10.9. Reporting 200
 10.9.1. Reporting automatisé pour le *test manager* 200

Chapitre 11. Standards et réglementations 203

11.1. Définition de standards 205
 11.2. Utilité et intérêt. 205
 11.3. Implémentation. 206
 11.4. Démonstration de conformité (IADT) 206
 11.5. Pseudo-standards et bonnes pratiques 207
 11.6. Adaptation des standards aux besoins 208
 11.7. Standards et procédures. 208
 11.8. Cohérence interne et externe des standards 209

Chapitre 12. Étude de cas 211

12.1. Étude de cas : amélioration d’un système complexe existant 211
 12.1.1. Contexte et organisation 212
 12.1.1.1. Contexte 212
 12.1.1.2. Organisation des *sprints* et impacts 213
 12.1.2. Risques, caractéristiques et domaines métier 215
 12.1.2.1. Risques et défis 215
 12.1.2.2. Caractéristiques à couvrir 216
 12.1.2.3. Domaines. 217
 12.1.3. Approche et environnement 217
 12.1.3.1. Approche 218
 12.1.3.2. Test des API 220
 12.1.3.3. Implémentation 220
 12.1.3.4. Environnements. 222
 12.1.3.5. Déploiement 223
 12.1.3.6. Focalisation des corrections. 223
 12.1.3.7. Parallélisation des versions 224
 12.1.3.8. Gestion des risques 225
 12.1.3.9. *Shift left*. 227
 12.1.3.10. Conditions de test et tests exploratoires 228
 12.1.3.11. Scénarios de test UAT 228
 12.1.4. Ressources, outils et personnel 228
 12.1.4.1. Ressources d’environnements 228
 12.1.4.2. Automatisation et outils 229
 12.1.4.3. Personnel. 229
 12.1.4.4. Amélioration continue 230

12.1.5. Livrables, <i>reporting</i> et documentation	230
12.1.6. Planning et déroulement	231
12.1.6.1. Planning	232
12.1.6.2. Déroulement	233
12.1.6.3. Échanges et interactions	235
12.1.7. Aspects logistiques et campagnes	235
12.1.8. Techniques de test	236
12.1.9. Conclusions et retour d'expérience	237
12.1.9.1. Commentaires et documentation	237
12.1.9.2. Architecture	238
12.1.9.3. Planning et suivi	238
12.1.9.4. Formation et partage des connaissances	239
12.1.9.5. <i>Backlog</i> de défauts	239
12.1.9.6. Critères d'acceptation	239
12.1.9.7. Tests techniques	239
12.1.9.8. Fiabilité des tests et des revues	240
12.1.9.9. <i>Reporting</i> et synthèse	240
12.1.9.10. Estimation de la charge	240
12.1.9.11. Environnements	241
Chapitre 13. Challenges des tests	243
13.1. Dette technique	243
13.1.1. Origine de la dette technique	244
13.1.2. Éléments de la dette technique	246
13.1.3. Mesurer la dette technique	246
13.1.4. Réduire la dette technique	247
13.2. Défis spécifiques des systèmes de systèmes	248
13.3. Gestion correcte des projets	249
13.4. DevOps	250
13.4.1. Idéaux de DevOps	251
13.4.2. Défis particuliers du DevOps	252
13.5. Internet des objets (IoT)	252
13.6. Big Data	254
13.7. Services et microservices	255
13.8. Containers, Docker, Kubernetes, etc.	255
13.9. Intelligence artificielle (AI) et <i>Machine Learning</i> (ML)	256
13.10. Multiplateforme, mobilité et disponibilité	258
13.11. Complexité	259
13.12. Dépendances insoupçonnées	260
13.13. Automatisation	260

13.13.1. Attentes irréalistes	261
13.13.2. ROI difficile à atteindre	262
13.13.3. Difficulté d'implémentation	263
13.13.3.1. Multiplicité des environnements	263
13.13.3.2. Interactions des environnements	264
13.13.3.3. Reproductibilité des tests.	264
13.13.4. Penser à la maintenance.	265
13.13.5. Quelle confiance accorder aux outils ?	265
13.14. Sécurité.	266
13.15. Aveuglement ou dissonance cognitive	267
13.16. Quatre vérités	268
13.16.1. Importance des individus	268
13.16.2. Qualité <i>versus</i> quantité	269
13.16.3. Formation, expérience et expertise	269
13.16.4. Intérêt des certifications.	270
13.17. Besoin d'anticipation	271
13.18. Toujours se réinventer	272
13.19. <i>Last but not least</i>	272
Glossaire	275
Bibliographie	283
Index	291
Sommaire de <i>Les tests de logiciels 1</i>	295
Sommaire de <i>Les tests de logiciels 2</i>	297