

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