

Inhalt

Geleitwort	XI
Vorwort	XIX
Die Autoren	XXIII
1 Agil – ein kultureller Wandel	1
1.1 Der Weg zur agilen Entwicklung	1
1.2 Gründe für die agile Entwicklung	4
1.3 Die Bedeutung des Agilen Manifests für das Testen von Software	8
1.4 Agiles Arbeiten erfordert einen kulturellen Wandel bei den Benutzern	10
1.5 Konsequenzen der agilen Entwicklung für die Softwarequalitätssicherung	12
1.5.1 Räumliche Auswirkungen	12
1.5.2 Zeitliche Folgen	13
2 Agile Vorgehensmodelle und deren Sicht auf Qualitätssicherung	17
2.1 Herausforderungen in der Qualitätssicherung	18
2.1.1 Qualität und Termin	18
2.1.2 Qualität und Budget	19
2.1.3 Der Stellenwert des Softwaretests	20
2.1.4 Fehler aus Vorprojekten (Technical Debt)	21
2.1.5 Testautomatisierung	22
2.1.6 Hierarchische Denkweise	23
2.2 Der Stellenwert des Teams	23
2.3 Qualitätssicherung in agilen Projekten	25
2.3.1 Scrum	26
2.3.1.1 Qualitätssicherung in den Sprints	26
2.3.1.2 Sprint Review Meeting	27
2.3.1.3 Sprint Retrospektive	27
2.3.2 Kanban	29
2.3.2.1 Kaizen – Continuous Improvement	29
2.4 Continuous Integration	30
2.5 Lean Software Development	31

3	Die Organisation des Softwaretests in agilen Projekten	33
3.1	Die Platzierung von Tests in agilen Projekten	34
3.1.1	Die Testaktivitäten gemäß ISTQB	34
3.1.1.1	Testplanung, Testüberwachung und -steuerung	34
3.1.1.2	Testanalyse und Testentwurf	38
3.1.1.3	Testrealisierung und Testdurchführung	39
3.1.1.4	Abschluss der Testaktivitäten	40
3.1.2	Welcher Test wofür – die vier Testquadranten agilen Testens	42
3.1.2.1	Erster Quadrant: technisch orientiert und teamunterstützend	43
3.1.2.2	Zweiter Quadrant: fachlich orientiert und teamunterstützend	46
3.1.2.3	Dritter Quadrant: fachlich orientiert und produkthinterfragend	49
3.1.2.4	Vierter Quadrant: technisch orientiert und produkthinterfragend	50
3.1.2.5	Der Kontext	52
3.1.3	Tipps für den Softwaretest aus agiler Perspektive	53
3.1.4	Agil im Großen mit SAFe® oder LeSS	55
3.1.4.1	Testen mit SAFe®	56
3.1.4.2	Testen mit LeSS	60
3.2	Praxisbeispiele	63
3.2.1	Die Rolle des Testers und ihre Veränderung im Laufe der Zeit zum Quality Specialist bei otto.de – ein Erfahrungsbericht	63
3.2.2	Abnahmetest als eigenes Scrum-Projekt/-Team	66
3.2.3	Test Competence Center für agile Projekte	68
3.2.4	Team im Healthcare-Bereich nutzt V-Modell	69
4	Die Rolle des Testers in agilen Projekten	71
4.1	Generalist vs. Spezialist	71
4.2	Der Weg vom zentralen Testcenter in das agile Team	74
4.2.1	Varianten der Testereinbindung in traditionellen Teams	74
4.2.2	Varianten der Testereinbindung in agilen Teams	76
4.2.2.1	Die Umstellung auf agiles Vorgehen	76
4.2.2.2	Steigerung von Effizienz und Effektivität	77
4.2.2.3	Teamzusammenstellung	78
4.3	Herausforderungen der Tester im Team	85
4.3.1	Die Tester im agilen Team	85
4.3.2	Neues Rollenverständnis finden	86
4.3.2.1	Vom Testmanager zum Quality Coach	86
4.3.2.2	Vom Tester zum Quality Engineer	86
4.3.3	Rechtzeitige Problemaufdeckung	87
4.3.4	Die Entstehung technischer Schulden	89
4.4	Teams und Tester im Kampf gegen „technical debt“	90
4.4.1	Was ist „technical debt“?	90
4.4.2	Der Umgang mit technischen Schulden	92

4.5	Erfahrungsbericht: Quality Specialist bei <i>otto.de</i>	94
4.5.1	Wir agieren als Quality Coach des Teams	94
4.5.2	Wir begleiten den kompletten Story-Lifecycle	95
4.5.3	Wir betreiben Continuous Delivery/Continuous Deployment	95
4.5.4	Wir balancieren die unterschiedlichen Testarten der Testpyramide ...	96
4.5.5	Wir helfen dem Team, die richtigen Methoden für hohe Qualität einzusetzen	96
4.5.6	Wir sind im Pairing aktiv	97
4.5.7	Wir vertreten unterschiedliche Perspektiven	97
4.5.8	Wir sind Kommunikationstalente	98
4.5.9	Wir sind Quality Specialists	98
4.6	Die Herausforderung der Veränderung	99
4.6.1	Ausgangslage	99
4.6.2	Faktoren für die Entwicklung zum agilen Vorgehen	100
4.6.2.1	Kreativität und Flexibilität	100
4.6.2.2	Verhaftet in alten Denkmustern	100
4.6.2.3	Trägheit, fehlende Beweglichkeit	101
4.6.2.4	Arbeitsumfeld	101
4.6.2.5	Veränderte Rollen der Senior-Tester/Senior-Manager	101
4.7	Hilfreiche Tipps aus Projekt- und Community-Erfahrung	102
4.7.1	Zero Testing – Qualität als Haltung	103
5	Agiles Testmanagement, agile Testmethoden und Testtechniken	105
5.1	Testmanagement	106
5.1.1	Testplanung im nicht agilen Umfeld	106
5.1.2	Testplanung im agilen Umfeld	108
5.1.3	Testkonzept	110
5.1.4	Testaktivitäten in Iteration Zero – Initialisierungs-Sprint	112
5.1.5	Externe Unterstützung der Testplanung	114
5.1.6	Testschätzung	115
5.1.7	Testorganisation	116
5.1.8	Testerstellung, Durchführung und Release	117
5.2	Testmethoden im agilen Umfeld	119
5.2.1	Risikobasiertes und valuebasiertes Testen	119
5.2.2	Explorativer Test	122
5.2.3	Session-basiertes Testen	123
5.2.4	Abnahmetestgetriebene Entwicklung	126
5.2.5	Testautomatisierung	126
5.3	Wesentliche Einflussfaktoren auf den Test	127
5.3.1	Continuous Integration (CI)	128
5.3.2	Automatisiertes Konfigurationsmanagement	129
5.4	Die besonderen Herausforderungen beim Test von verteilten Systemen	130
5.4.1	Die Herausforderung für agile Teams im Test von verteilten Systemen	131
5.5	Künstliche Intelligenz, maschinelles Lernen und agiler Test	133
5.5.1	Wie testet man KI/ML-Systeme?	133

6	Agile Testdokumentation	137
6.1	Die Rolle der Dokumentation in der Softwareentwicklung	137
6.2	Der Nutzen der Dokumentation	139
6.3	Dokumentationsarten	142
6.3.1	Anforderungsdokumentation	142
6.3.2	Codedokumentation	144
6.3.3	Testdokumentation	145
6.3.3.1	Testfallbeschreibung	145
6.3.3.2	Testdurchführung	146
6.3.3.3	Testüberdeckung	147
6.3.3.4	Fehlerdokumentation	147
6.3.4	Benutzerdokumentation	149
6.4	Der Tester als Dokumentierer	150
6.5	Stellenwert der Dokumentation im agilen Test	151
7	Agile Testautomatisierung	153
7.1	Die Crux mit den Werkzeugen in agilen Projekten	153
7.2	Testautomatisierung – wie geht man es an?	156
7.3	Testautomatisierung mit zunehmender Integration der Software	157
7.3.1	Unittest bzw. Komponententest	158
7.3.2	Komponentenintegrationstest	158
7.3.3	Systemtest	158
7.3.4	Systemintegrationstest	159
7.4	xUnittest-Frameworks	159
7.5	Einsatz von Platzhaltern	165
7.6	Integrationsserver	166
7.7	Testautomatisierung im fachlich orientierten Test	168
7.7.1	Testautomatisierungs-Frameworks	171
7.7.2	Schlanke versus umfassende Automatisierung von Benutzereingaben	172
7.7.2.1	Schlanke Testautomatisierung	172
7.7.2.2	Umfassende Testautomatisierung	174
7.7.3	Ein typisches Beispiel: FitNesse und Selenium	175
7.7.4	Behavior-Driven Development mit Cucumber und Gherkin	180
7.8	Testautomatisierung im Last- und Performanztest	183
7.9	Die sieben schlechtesten Ideen für die Testautomatisierung	183
7.9.1	Den Erfolg nach wenigen Sprints erwarten	184
7.9.2	Testwerkzeugen blind vertrauen	184
7.9.3	Schreiben der Testskripts als Nebenbeschäftigung ansehen	185
7.9.4	Testdaten irgendwo in Testfällen vergraben	185
7.9.5	Testautomatisierung nur mit Benutzeroberflächen in Verbindung bringen	186
7.9.6	Soll-Ist-Vergleich unterschätzen	186
7.9.7	(Un-)Testbarkeit der Applikation einfach hinnehmen	187

8	Werkzeugeinsatz in agilen Projekten	189
8.1	Projektmanagement	190
8.1.1	Broadcom Rally	192
8.2	Anforderungsmanagement	193
8.2.1	Polarion QA/ALM	196
8.3	Fehlermanagement	199
8.3.1	Pachno	202
8.3.2	Atlassian JIRA	205
8.4	Testplanung und -steuerung	207
8.4.1	Atlassian JIRA	209
8.5	Testanalyse und Testentwurf	211
8.5.1	Risikobasiertes Testen in der Tricentis-Testsuite	213
8.6	Testrealisierung und Testdurchführung	214
8.6.1	Azure Test Plans	217
9	Ausbildung und ihre Bedeutung	219
9.1	ISTQB® Certified Tester	220
9.2	A4Q Practitioner in Agile Quality 2.0	222
9.2.1	Motivation	223
9.2.2	Training-Insights	223
9.2.3	A4Q Software Development Engineer in Test (SDET)	224
9.3	Individuelle Trainings (Customized Trainings)	225
9.3.1	Empfohlenes Vorgehen bei Einführung der Agilität	225
9.3.1.1	Bestandsaufnahme der Ist-Situation	225
9.3.1.2	Abhängigkeitsanalyse	226
9.3.1.3	Definieren des „neuen“ Ziels	226
9.3.2	Organisatorisches	226
9.3.3	Pilotphase	226
9.3.4	Ausrollen in Unternehmen	227
10	Retrospektive	229
	Literaturverzeichnis	233
	Index	239