

Inhalt

Geleitwort	X
Vorwort	XVII
Praxisbeispiele	XIX
Die Autoren	XX
1 Agil – ein kultureller Wandel	1
1.1 Der Weg zur agilen Entwicklung	1
1.2 Gründe für eine agile Entwicklung	4
1.3 Die Bedeutung des Agilen Manifests für den Software-Test	7
1.4 Agil setzt Kulturwandel bei den Anwendern voraus	10
1.5 Konsequenzen der agilen Entwicklung für die Software-Qualitätssicherung	12
1.5.1 Räumliche Konsequenzen	12
1.5.2 Zeitliche Konsequenzen	13
2 Agile Vorgehensmodelle und deren Sicht auf Qualitätssicherung	15
2.1 Herausforderungen in der Qualitätssicherung	16
2.1.1 Qualität und Termin	16
2.1.2 Qualität und Budget	17
2.1.3 Der Stellenwert des Software-Tests	18
2.1.4 Fehler aus Vorprojekten (Technical Debt)	20
2.1.5 Testautomatisierung	21
2.1.6 Hierarchische Denkweise	22
2.2 Der Stellenwert des Teams	22
2.3 Audits zur Qualitätssicherung in agilen Projekten	24
2.3.1 Scrum	24
2.3.1.1 Sprint Review Meeting	26
2.3.1.2 Sprint Retrospektive	27
2.3.2 Kanban	31
2.3.2.1 Kaizen – Continuous Improvement	32

2.4	Continuous Integration	33
2.5	Lean Software Development	34
3	Die Organisation des Software-Tests in agilen Projekten	37
3.1	Die Platzierung von Tests in agilen Projekten	38
3.1.1	Der fundamentale Testprozess des ISTQB	38
3.1.1.1	Testplanung und -steuerung	38
3.1.1.2	Testanalyse und Testentwurf	41
3.1.1.3	Testrealisierung und Testdurchführung	42
3.1.1.4	Bewertung von Endekriterien und Bericht	43
3.1.1.5	Abschluss der Testaktivitäten	44
3.1.2	Welcher Test wofür – die vier Testquadranten agilen Testens	46
3.1.2.1	Erster Quadrant: technisch orientiert und teamunterstützend	47
3.1.2.2	Zweiter Quadrant: fachlich orientiert und teamunterstützend	50
3.1.2.3	Dritter Quadrant: fachlich orientiert, aber produkthinterfragend	53
3.1.2.4	Vierter Quadrant: technisch orientiert aber produkthinterfragend	54
3.1.2.5	Der Kontext	56
3.1.3	Tipps für den Software-Test aus agiler Perspektive	57
3.1.4	Agil im Großen mit SAFe oder LeSS	60
3.1.4.1	Testen mit SAFe	60
3.1.4.2	Testen mit LeSS	63
3.1.5	Skalierbare Organisation agiler Teams	65
3.2	Praxisbeispiele	68
3.2.1	Die Rolle des Testers und ihre Veränderung im Laufe der Zeit zum Quality Specialist bei <i>otto.de</i> – ein Erfahrungsbericht	68
3.2.2	Abnahmetest als eigenes Scrum-Projekt/-Team	72
3.2.3	Test Competence Center für agile Projekte	73
3.2.4	Team im Healthcare-Bereich nutzt V-Modell	74
4	Die Rolle des Testers in agilen Projekten	77
4.1	Generalist vs. Spezialist	77
4.2	Der Weg vom zentralen Testcenter in das agile Team	79
4.2.1	Varianten der Testereinbindung in traditionellen Teams	80
4.2.2	Varianten der Testereinbindung in agile Teams	81
4.2.2.1	Die Umstellung von der traditionellen auf die agile Welt ..	82
4.2.2.2	Steigerung von Effizienz und Effektivität	83
4.2.2.3	Teamzusammenstellung	84
4.3	Herausforderungen der Tester im Team	90
4.3.1	Die Tester im agilen Team	90
4.3.1.1	Der Quality Coach	91

4.3.1.2	Aufgaben der agilen Tester	91
4.3.2	Rechtzeitige Problemaufdeckung	93
4.3.3	Die Entstehung technischer Schulden	95
4.4	Teams und Tester im Kampf gegen „technical debt“	96
4.4.1	Was ist „technical debt“?	96
4.4.2	Der Umgang mit technischen Schulden	98
4.5	Erfahrungsbericht: Quality Specialist bei <i>otto.de</i>	100
4.5.1	Wir agieren als Quality-Coach des Teams	100
4.5.2	Wir begleiten den kompletten Story-Lifecycle	101
4.5.3	Wir betreiben Continuous Delivery/Continuous Deployment	101
4.5.4	Wir balancieren die unterschiedlichen Testarten der Testpyramide	102
4.5.5	Wir helfen dem Team, die richtigen Methoden für hohe Qualität einzusetzen	102
4.5.6	Wir sind im Pairing aktiv	103
4.5.7	Wir vertreten unterschiedliche Perspektiven	103
4.5.8	Wir sind Kommunikationstalente	104
4.5.9	Wir sind Quality Specialists	104
4.6	Zu alt für agil? Die mentale Herausforderung	105
4.6.1	Ausgangslage	105
4.6.2	Was führt zur Aussage „Agil ist etwas für junge Leute“?	106
4.6.2.1	Kreativität und Flexibilität	106
4.6.2.2	Verhaftet in alten Denkmustern	106
4.6.2.3	Trägheit, fehlende Beweglichkeit	107
4.6.2.4	Arbeitsumfeld	107
4.6.2.5	Vorteile der Jugend	108
4.6.2.6	Stärken der Senior-Tester/Senior-Manager	108
4.7	Hilfreiche Tipps vom Markt	109
5	Agiles Testmanagement, -methoden und -techniken	111
5.1	Testmanagement	112
5.1.1	Testplanung im traditionellen Umfeld	112
5.1.2	Testplanung im agilen Umfeld	114
5.1.3	Testkonzept	116
5.1.4	Testaktivitäten in Iteration Zero – Initialisierungs-Sprint	119
5.1.5	Externe Unterstützung der Testplanung	120
5.1.6	Testschätzung	120
5.1.7	Testorganisation	122
5.1.8	Testerstellung, Durchführung und Release	123
5.2	Testmethoden im agilen Umfeld	124
5.2.1	Risikobasiertes und valuebasiertes Testen	125
5.2.2	Explorativer Test	128
5.2.3	Session-basiertes Testen	129
5.2.4	Abnahmetestgetriebene Entwicklung	132
5.2.5	Testautomatisierung	132

5.3	Wesentliche Einflussfaktoren auf den Test	133
5.3.1	Continuous Integration (CI)	133
5.3.2	Automatisiertes Konfigurationsmanagement	135
5.4	Die besonderen Herausforderungen beim Test von IoT	136
5.4.1	Was ist das Internet of Things?	136
5.4.2	Die Herausforderung für agile Teams im Test	138
6	Agile Testdokumentation	141
6.1	Die Rolle der Dokumentation in der Software-Entwicklung	142
6.2	Der Nutzen der Dokumentation	143
6.3	Dokumentationsarten	146
6.3.1	Anforderungsdokumentation	147
6.3.2	Codedokumentation	148
6.3.3	Testdokumentation	149
6.3.3.1	Testfallbeschreibung	149
6.3.3.2	Testdurchführung	150
6.3.3.3	Testüberdeckung	150
6.3.3.4	Fehlerdokumentation	151
6.3.4	Benutzerdokumentation	152
6.4	Der Tester als Dokumentierer	154
6.5	Stellenwert der Dokumentation im agilen Test	154
7	Agile Testautomatisierung	157
7.1	Die Crux mit den Werkzeugen in agilen Projekten	157
7.2	Testautomatisierung – wie geht man es an?	159
7.3	Testautomatisierung mit zunehmender Integration der Software	161
7.3.1	Unit Test bzw. Komponententest	161
7.3.2	Komponentenintegrationstest	162
7.3.3	Systemtest	162
7.3.4	Systemintegrationstest	162
7.4	xUnit-Frameworks	163
7.5	Einsatz von Platzhaltern	169
7.6	Integrationsserver	170
7.7	Testautomatisierung im fachlich orientierten Test	172
7.7.1	Ein Framework – wozu?	174
7.7.2	Agile versus klassische Automatisierung von Benutzereingaben ..	176
7.7.2.1	Agile Testautomatisierung	176
7.7.2.2	Klassische Testautomatisierung	177
7.7.3	Ein typisches Beispiel: FitNesse und Selenium	179
7.7.4	Behavior Driven Development mit Cucumber und Gherkin	183
7.8	Testautomatisierung im Last- und Performance-Test	186
7.9	Die sieben schlechtesten Ideen für die Testautomatisierung	186
7.9.1	Den Erfolg nach wenigen Sprints erwarten	187

7.9.2	Testwerkzeugen blind vertrauen	187
7.9.3	Schreiben der Testskripts als Nebenbeschäftigung ansehen	188
7.9.4	Testdaten irgendwo in Testfällen vergraben	188
7.9.5	Testautomatisierung nur mit Benutzeroberflächen in Verbindung bringen	189
7.9.6	Soll-Ist-Vergleich unterschätzen	189
7.9.7	(Un-)Testbarkeit der Applikation einfach hinnehmen	190
8	Werkzeugeinsatz in agilen Projekten	191
8.1	Projektmanagement	192
8.1.1	CA Agile Central	194
8.2	Anforderungsmanagement	195
8.2.1	Polarion QA/ALM	198
8.3	Fehlermanagement	201
8.3.1	The Bug Genie	204
8.3.2	Atlassian JIRA	206
8.4	Testplanung und -steuerung	208
8.4.1	Atlassian JIRA	210
8.5	Testanalyse und Testentwurf	213
8.5.1	Risikobasiertes Testen in der TOSCA-Testsuite	214
8.6	Testrealisierung und Testdurchführung	215
8.6.1	Microsoft Test Manager	218
9	Ausbildung und ihre Bedeutung	221
9.1	ISTQB Certified Tester	222
9.2	Certified Agile Tester / CAT	227
9.2.1	Motivation	228
9.2.2	Training-Insights	229
9.3	ISTQB Certified Tester Foundation Level Extension Agile Tester	231
9.4	Individuelle Trainings (Customized Trainings)	232
9.4.1	Empfohlenes Vorgehen bei Einführung der Agilität	232
9.4.1.1	Bestandsaufnahme der Ist-Situation	232
9.4.1.2	Abhängigkeitsanalyse	233
9.4.1.3	Definieren des „neuen“ Ziels	233
9.4.2	Organisatorisches	233
9.4.3	Pilotphase	233
9.4.4	Ausrollen in Unternehmen	234
10	Retrospektive	235
	Literaturverzeichnis	238
	Index	243