

Inhalt

Geleitwort	IX
Vorwort	XVII
Die Autoren	XXI
Danksagungen	XXIII
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	9
1.5 Konsequenzen der agilen Entwicklung für die Software-Qualitätssicherung	11
1.5.1 Räumliche Konsequenzen	11
1.5.2 Zeitliche Konsequenzen	12
2 Agile Vorgehensmodelle und ihre 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
2.3.2 Kanban	31
2.4 Continuous Integration	33
2.5 Lean Software Development	33

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
3.1.2	Welcher Test wofür – Die vier Testquadranten agilen Testens	46
3.1.3	Tipps für den Software-Test aus agiler Perspektive	56
3.1.4	Skalierbare Organisation agiler Teams	59
3.2	Praxisbeispiele	66
3.2.1	Abnahmetest als eigenes Scrum-Projekt/-Team	66
3.2.2	Test Competence Center für agile Projekte	68
3.2.3	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	73
4.2.1	Varianten der Testereinbindung in traditionellen Teams	74
4.2.2	Varianten der Testereinbindung in agile Teams	75
4.3	Herausforderungen der Tester im Team	84
4.3.1	Die Tester im agilen Team	84
4.3.2	Rechtzeitige Problemaufdeckung	86
4.3.3	Die Entstehung technischer Schulden	88
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	91
4.5	Zu alt für agil? Die mentale Herausforderung	93
4.5.1	Ausgangslage	93
4.5.2	Was führt zur Aussage „Agil ist etwas für junge Leute“?	94
4.6	Hilfreiche Tipps vom Markt	97
5	Agiles Testmanagement, -methoden und -techniken	101
5.1	Testmanagement	102
5.1.1	Testplanung im traditionellen Umfeld	102
5.1.2	Testplanung im agilen Umfeld	103
5.1.3	Testkonzept	106
5.1.4	Testaktivitäten in Iteration Zero – Initialisierungs-Sprint	108
5.1.5	Externe Unterstützung der Testplanung	110
5.1.6	Testschätzung	110
5.1.7	Testorganisation	111
5.1.8	Testerstellung, Durchführung und Release	112
5.2	Testmethoden im agilen Umfeld	114
5.2.1	Risikobasiertes und valuebasiertes Testen	114
5.2.2	Explorativer Test	117
5.2.3	Session-basiertes Testen	118
5.2.4	Abnahmetestgetriebene Entwicklung	121
5.2.5	Testautomatisierung	122

5.3	Wesentliche Einflussfaktoren auf den Test	122
5.3.1	Continuous Integration (CI)	123
5.3.2	Automatisiertes Konfigurationsmanagement	125
6	Agile Testdokumentation	127
6.1	Die Rolle der Dokumentation in der Software-Entwicklung	128
6.2	Der Nutzen der Dokumentation	129
6.3	Dokumentationsarten	132
6.3.1	Anforderungsdokumentation	132
6.3.2	Code-Dokumentation	134
6.3.3	Testdokumentation	135
6.3.4	Benutzerdokumentation	138
6.4	Der Tester als Dokumentierer	139
6.5	Stellenwert der Dokumentation im agilen Test	140
7	Agile Testautomatisierung	141
7.1	Die Crux mit den Werkzeugen in agilen Projekten	141
7.2	Testautomatisierung – Wie geht man es an?	143
7.3	Testautomatisierung mit zunehmender Integration der Software	145
7.3.1	Unit Test bzw. Komponententest	145
7.3.2	Komponentenintegrationstest	146
7.3.3	Systemtest	146
7.3.4	Systemintegrationstest	146
7.4	xUnit-Frameworks	147
7.5	Einsatz von Platzhaltern	152
7.6	Integrationsserver	153
7.7	Testautomatisierung im fachlich orientierten Test	154
7.7.1	Ein Framework – wozu?	157
7.7.2	Agile versus klassische Automatisierung von Benutzereingaben	158
7.7.3	Ein typisches Beispiel: FitNesse und Selenium	161
7.8	Testautomatisierung im Last- und Performance-Test	166
7.9	Die sieben schlechtesten Ideen für die Testautomatisierung	167
7.9.1	Den Erfolg nach wenigen Sprints erwarten	167
7.9.2	Testwerkzeugen blind vertrauen	168
7.9.3	Schreiben der Testskripts als Nebenbeschäftigung ansehen	168
7.9.4	Testdaten irgendwo in Testfällen vergraben	169
7.9.5	Testautomatisierung nur mit Benutzeroberflächen in Verbindung bringen	169
7.9.6	Soll-Ist-Vergleich unterschätzen	170
7.9.7	(Un-)Testbarkeit der Applikation einfach hinnehmen	170
8	Werkzeugeinsatz in agilen Projekten	173
8.1	Projektmanagement	174
8.1.1	Rally	176

8.2	Anforderungsmanagement	178
8.2.1	Polarion QA/ALM	180
8.3	Fehlermanagement	183
8.3.1	The Bug Genie	186
8.3.2	Atlassian JIRA	189
8.4	Testplanung und -steuerung	191
8.4.1	Atlassian JIRA	193
8.5	Testanalyse und Testentwurf	195
8.5.1	Risikobasiertes Testen in der TOSCA-Testsuite	196
8.6	Testrealisierung und Testdurchführung	197
8.6.1	Microsoft TFS Test Manager	200
9	Ausbildung und ihre Bedeutung	203
9.1	ISTQB Certified Tester	205
9.2	Certified Agile Tester/CAT	210
9.2.1	Motivation	210
9.2.2	Training-Insights	211
9.3	Individuelle Trainings (Customized Trainings)	213
9.3.1	Empfohlenes Vorgehen bei Einführung der Agilität	214
9.3.2	Organisatorisches	215
9.3.3	Pilotphase	215
9.3.4	Ausrollen in Unternehmen	216
10	Retrospektive	217
	Literaturverzeichnis	221
	Index	227