

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	
1.1	Motivation.....	1
1.2	Abgrenzung des Buchs zu ISTQB-Lehrplänen .....	1
1.3	Zur Gliederung dieses Buchs .....	2
1.4	Die wichtigsten Begriffe kurz erklärt .....	3
1.4.1	Definition von Fachbegriffen .....	3
1.4.2	Zu Definitionen und TesternInnen .....	5
1.5	Ein Überblick über das Umfeld des Software-Testing .....	5
1.5.1	Ursachen von Softwarefehlern .....	6
1.5.2	Warum Programmfehler nicht entdeckt werden .....	7
1.5.3	Angebrachter Testaufwand .....	8
1.5.4	Der Tester und der Testprozess .....	9
1.5.5	Modellieren der Software-Umgebung .....	10
1.5.6	Erstellen von Testfällen .....	12
1.5.7	Ausführen und Evaluieren der Tests .....	13
1.5.8	Messen des Testfortschritts .....	14
1.5.9	Testdesign und Testdokumentation im Entwicklungsprozess .....	15
1.5.10	Verschiedene Teststufen und deren Zusammenspiel .....	16
1.5.11	Andere Verifikationsmethoden als Ergänzung zum Test .....	19
1.5.12	Agile Prozessmodelle .....	20
1.5.13	Der Softwaretest in agilen Vorgehensmodellen .....	21
1.5.14	Wer testet die Tester? .....	23
<b>2</b>	<b>Anforderungen und Test</b>	<b>25</b>
2.1	Die Bedeutung textueller Anforderungen .....	25
2.2	Requirements Engineering im Projekt .....	26
2.3	Arten und Quellen von Anforderungen .....	27

2.4	Warum Anforderungen dokumentiert werden sollen . . . . .	28
2.5	Die Review von Anforderungen . . . . .	29
2.5.1	Testbarkeit von Anforderungen . . . . .	30
2.5.2	Modifizierbarkeit und Erweiterbarkeit . . . . .	31
2.5.3	Relevanz von Anforderungen . . . . .	32
2.6	Der Umgang mit natürlicher Sprache . . . . .	32
2.6.1	Einfache Sprache gegen Missverständnisse . . . . .	32
2.6.2	Gelenkte Sprache . . . . .	34
2.7	Hinweise zur Dokumentenform . . . . .	35
2.8	Die Spezifikation an der Schnittstelle zum Test-Team. . . . .	38
2.8.1	Konfiguration von Testdesigns . . . . .	38
2.8.2	Vollständigkeit von Spezifikationen . . . . .	38
2.9	Werkzeuge zur Review von Anforderungen . . . . .	39
2.10	Diskussion . . . . .	41
2.10.1	Verifikation beim Requirements Engineering mit Augenmaß . . . . .	41
2.10.2	Bewertung der Rolle des RE für den Testprozess . . . . .	41
2.11	Fragen und Übungsaufgaben . . . . .	42
<b>3</b>	<b>Review des Designs</b>	<b>45</b>
3.1	Ziele der Review des Architekturdesigns . . . . .	45
3.2	Ziele der Review des Detaildesigns . . . . .	46
3.3	Eigenschaften von gutem Software-Design . . . . .	47
3.4	Hinweise zur Architekturdesignreview . . . . .	47
3.5	Embedded Design. . . . .	51
3.5.1	Sicherheit, Verfügbarkeit & Co . . . . .	51
3.5.2	Wartbarkeit des Geräts. . . . .	51
3.5.3	Ressourcenverbrauch . . . . .	52
3.5.4	Design von Echtzeitsystemen . . . . .	52
3.6	Diskussion . . . . .	52
3.7	Fragen und Übungsaufgaben . . . . .	53
<b>4</b>	<b>Automatische statische Code-Analyse</b>	<b>55</b>
4.1	Motivation zum Einsatz von Analyse-Werkzeugen . . . . .	55
4.2	Techniken von Analyse-Werkzeugen im unteren Preissegment . . . . .	56
4.2.1	Sprachspezifische Fallstricke . . . . .	58
4.2.2	Kontrollflussanalyse . . . . .	59

4.2.3	Datenflussanalyse, Initialisation Tracking	60
4.2.4	Datenflussanalyse, Value Tracking	61
4.2.5	Semantische Analyse	63
4.2.6	Starke Typenprüfung	64
4.3	Techniken von Analysewerkzeugen im oberen Preissegment	65
4.3.1	Größerer Komfort für den Benutzer	65
4.3.2	Concurrency Checks	66
4.3.3	Stack-Analyse und erweiterte Kontrollflussanalyse	66
4.3.4	Erschöpfende Analyse des Zustandsbaums	67
4.4	Statische Security-Analyse (SSA)	68
4.5	Code-Metriken	69
4.6	Werkzeuge für die Automatische Code-Analyse	71
4.7	Diskussion	73
4.8	Fragen und Übungsaufgaben	74
<b>5</b>	<b>Code-Reviews</b>	<b>75</b>
5.1	Review-Arten	75
5.1.1	Code-Inspektionen	75
5.1.2	Walkthrough	76
5.1.3	Peer-Review	77
5.2	Pair Programming	77
5.3	Werkzeuge zur Code-Review	78
5.4	Diskussion	79
5.5	Fragen und Übungsaufgaben	82
<b>6</b>	<b>Unit-Tests</b>	<b>85</b>
6.1	Der Unit-Test im Entwicklungsprozess	85
6.2	Zur Definition von Unit-Test und Modultest	86
6.3	Black-Box-Testfälle beim White-Box-Test	86
6.3.1	Äquivalenzklassenbildung	87
6.3.2	Grenzwertanalyse	88
6.3.3	Andere Methoden	90
6.4	Stubs und Treiber	91
6.5	Verschiedene Typen von Werkzeugen beim White-Box-Test	99
6.5.1	Unit-Test-Frameworks	99
6.5.2	Werkzeuge zur Testerstellung	99
6.5.3	Werkzeuge zur Messung der Testabdeckung	102

6.6	Testabdeckung . . . . .	103
6.6.1	Statement Coverage . . . . .	103
6.6.2	Branch Coverage und Decision Coverage . . . . .	104
6.6.3	Decision/Condition Coverage . . . . .	104
6.6.4	Modified Condition/Decision Coverage . . . . .	105
6.6.5	Andere Testabdeckungen . . . . .	105
6.6.6	Testabdeckung bei modellbasierter Entwicklung . . . . .	105
6.6.7	Messung der Testabdeckung . . . . .	106
6.7	Basis Path Testing . . . . .	108
6.8	Host oder Target Testing? . . . . .	109
6.9	Den Code immer unverändert testen? . . . . .	111
6.10	Unit-Tests bei objektorientierten Sprachen . . . . .	112
6.11	Grenzen des Unit-Tests . . . . .	112
6.12	Werkzeuge für den Unit-Test . . . . .	113
6.12.1	Unit-Test-Frameworks . . . . .	113
6.12.2	Werkzeuge zur Testerstellung . . . . .	114
6.12.3	Coverage-Analyse . . . . .	115
6.13	Diskussion . . . . .	116
6.13.1	Testabdeckung . . . . .	116
6.13.2	Organisation von Unit-Tests . . . . .	117
6.14	Fragen und Übungsaufgaben . . . . .	118
<b>7</b>	<b>Integrationstests</b>	<b>123</b>
7.1	Software/Software-Integrationstest . . . . .	123
7.1.1	Bottom-up-Unit-Tests als Integrationstest . . . . .	123
7.1.2	Strukturierter Integrationstest . . . . .	126
7.1.3	Testabdeckung der Aufrufe von Unterprogrammen . . . . .	129
7.1.4	Vergleich der Teststrategien . . . . .	130
7.1.5	Grenzen des Software/Software-Integrationstests . . . . .	132
7.1.6	Diskussion des Software/Software-Integrationstests . . . . .	133
7.2	Ressourcentests . . . . .	134
7.2.1	Statischer Ressourcentest . . . . .	134
7.2.2	Dynamischer Ressourcentest . . . . .	136
7.3	Hardware/Software-Integrationstest . . . . .	138
7.3.1	Bottom-up-Verfahren . . . . .	138
7.3.2	Regressionsverfahren . . . . .	139
7.3.3	Black-Box-Verfahren . . . . .	139
7.3.4	Test und Analysen bei Sicherheitsrelevanz . . . . .	139
7.3.5	Diskussion des Hardware/Software-Integrationstests . . . . .	140

7.4	Systemintegrationstest	140
7.5	Werkzeuge für den Integrationstest	141
7.6	Fragen und Übungsaufgaben	142
<b>8</b>	<b>Systemtests</b>	<b>143</b>
8.1	Funktionale Systemtests	143
8.1.1	Zuordnung funktionaler Systemtests zu Anforderungen	143
8.1.2	Äquivalenzklassen und Grenzwerte im Black-Box-Test	144
8.1.3	Zustandsbasierter Test	146
8.1.4	Ursache-Wirkungs-Analyse	150
8.1.5	CECIL-Methode	159
8.1.6	Entscheidungstabellentechnik	160
8.1.7	Paarweises Testen und Klassifikationsbaum-Methode	160
8.1.8	Back To Back Testing	162
8.1.9	Erfahrungsbasierter Test	162
8.1.10	Diskussion des Black-Box-Tests	163
8.1.11	Auswahl eines Black-Box-Testverfahrens für eine Aufgabe	163
8.1.12	Werkzeuge für Funktionstests	164
8.2	Test der Benutzerschnittstelle	164
8.2.1	Grafische Benutzerschnittstelle	165
8.2.2	Werkzeuge für GUI-Tests	165
8.2.3	Eingebettete Benutzerschnittstellen	166
8.2.4	Werkzeuge für den Test von eingebetteten Benutzerschnittstellen	167
8.3	Performanztest und Lasttest	168
8.4	Stresstest	169
8.5	Volumentest	170
8.6	Failover und Recovery Testing	170
8.7	Ressourcentests	172
8.8	Installationstests	173
8.9	Konfigurationstests	174
8.10	Security-Tests	175
8.11	Dokumententests	176
8.12	Testumgebung und Testdaten	176
8.13	Formale Methoden	177
8.13.1	Symbolischer Test	178
8.13.2	Deduktive Verifikation von funktionalen Anforderungen	178
8.13.3	Model Checking	179

8.14	Automation von Systemtests . . . . .	180
8.14.1	Vor- und Nachteile der Testautomation . . . . .	181
8.14.2	Tipps zur Automation von Systemtests . . . . .	182
8.15	Dokumentation des Testdesigns und der Testergebnisse . . . . .	186
8.16	Grenzen des Systemtests . . . . .	187
8.17	Fragen und Übungsaufgaben . . . . .	188
<b>9</b>	<b>Testen von RTOS und Middleware</b>	<b>191</b>
9.1	Definition und Motivation . . . . .	191
9.2	White-Box-Requirements-Test . . . . .	192
9.3	Test eines Interrupt-Managers . . . . .	193
9.4	Test eines Schedulers . . . . .	195
9.5	Fragen und Übungsaufgaben . . . . .	196
<b>10</b>	<b>Race Conditions</b>	<b>197</b>
10.1	Definition von Data Races . . . . .	197
10.2	Dynamische Data-Race-Analyse . . . . .	201
10.2.1	Eraser . . . . .	201
10.2.2	Lamports Happens-Before-Relation . . . . .	205
10.3	Statische Data-Race-Analyse . . . . .	207
10.3.1	Ansätze zur statischen Data-Race-Analyse . . . . .	207
10.3.2	Vergleich zur dynamischen Data-Race-Analyse . . . . .	208
10.4	Werkzeuge für die Data-Race-Analyse . . . . .	209
10.5	Diskussion . . . . .	209
10.6	Fragen und Übungsaufgaben . . . . .	211
<b>11</b>	<b>Deadlocks</b>	<b>213</b>
11.1	Über die Entstehung von Deadlocks . . . . .	213
11.2	Verschiedene Arten der Deadlock-Analyse . . . . .	214
11.3	Dynamische Deadlock-Analyse . . . . .	215
11.4	Statische Deadlock-Analyse . . . . .	215
11.5	Werkzeuge zur Deadlockdetektion . . . . .	216
11.6	Diskussion . . . . .	216
11.7	Fragen und Übungsaufgaben . . . . .	217

<b>12</b>	<b>Echtzeit-Verifikation</b>	<b>219</b>
12.1	Antwortzeiten bei funktionalen Tests . . . . .	219
12.2	WCET-Analyse . . . . .	220
12.2.1	Problemstellung . . . . .	220
12.2.2	Laufzeitanalyse. . . . .	222
12.3	Werkzeuge für die WCET-Analyse. . . . .	225
12.4	Diskussion . . . . .	226
12.5	Fragen und Übungsaufgaben . . . . .	227
<b>13</b>	<b>Schedulability-Analyse</b>	<b>229</b>
13.1	Aufgaben der Schedulability-Analyse . . . . .	230
13.2	Definitionen . . . . .	230
13.3	Diskussion der Scheduling-Strategien . . . . .	231
13.3.1	Statisches Scheduling . . . . .	232
13.3.2	Dynamisches Scheduling . . . . .	233
13.4	Analyse bei Fixed-Priority-Single-CPU-Systemen . . . . .	236
13.4.1	Optimale Prioritätsvergabe. . . . .	236
13.4.2	Rate Monotonic Analysis. . . . .	236
13.4.3	Exakte Antwortzeitenanalyse . . . . .	237
13.4.4	Gegenseitiger Ausschluss . . . . .	242
13.4.5	Aperiodische Aufgaben . . . . .	244
13.4.6	Kontextwechsel . . . . .	245
13.4.7	Cache und Out Of Order Execution. . . . .	245
13.4.8	Input-Jitter . . . . .	245
13.4.9	Interrupts . . . . .	246
13.5	Multi-CPU-Systeme . . . . .	246
13.5.1	Multicore- und Multi-Prozessor-Systeme . . . . .	247
13.5.2	Verteilte Systeme . . . . .	247
13.6	Scheduling-Analyse für CAN . . . . .	249
13.7	Werkzeuge . . . . .	251
13.8	Diskussion . . . . .	251
13.9	Fragen und Übungsaufgaben . . . . .	253

<b>14</b>	<b>Hardware/Software-Interaktionsanalyse</b>	<b>255</b>
14.1	Die FMEA als Grundlage der HSIA .....	255
14.2	Die HSIA als Quelle für Software-Anforderungen .....	259
14.3	Software-Kritikalitätsanalyse .....	260
14.4	Software-FMEA .....	261
14.5	Werkzeuge .....	262
14.6	Diskussion .....	262
14.7	Fragen und Übungsaufgaben .....	263
<b>15</b>	<b>Modellbasierter Test</b>	<b>265</b>
15.1	Begriffsdefinition .....	265
15.2	MBT und Testautomation .....	266
15.3	Modelle .....	266
15.3.1	Statecharts .....	266
15.3.2	SDL .....	266
15.3.3	Message Sequence Charts .....	267
15.3.4	UML Version 2 .....	267
15.3.5	SysML .....	268
15.3.6	Funktions-Modellierung .....	268
15.4	Testmodell vs. Implementierungsmodell .....	268
15.5	Werkzeuge .....	269
15.6	Diskussion .....	270
15.7	Fragen und Übungsaufgaben .....	270
<b>16</b>	<b>Testmanagement</b>	<b>271</b>
16.1	Testplanung .....	271
16.2	Teststeuerung .....	273
16.3	Abweichungsmanagement .....	275
16.4	Bewertung und Anpassung des Testprozesses .....	277
16.4.1	Formale Reifegradmodelle für den Software-Test .....	277
16.4.2	Prozessbewertung in agilen Projekten .....	278
16.4.3	Mit Augenmaß ins Kostenoptimum .....	278
16.5	Risikobasierter Test .....	282
16.6	Werkzeuge .....	283



16.7	Diskussion .....	284
16.8	Fragen und Übungsaufgaben .....	285
<b>17</b>	<b>Qualitätsmanagement</b>	<b>287</b>
17.1	Definition .....	287
17.2	Qualitätsmanagement-Standards .....	288
17.3	Kosten und Haftungsrelevanz des Qualitätsmanagement .....	291
17.4	Umsetzung des Qualitätsmanagements .....	292
17.5	Die Rolle des Qualitätsmanagers .....	293
17.6	Mit Metriken die Qualität steuern .....	294
17.7	Die Wirtschaftlichkeit von Qualitätsmanagement .....	297
17.8	Werkzeuge .....	297
17.9	Diskussion .....	298
17.10	Fragen und Übungsaufgaben .....	299
<b>18</b>	<b>Softwaretest und Haftungsrisiko</b>	<b>301</b>
18.1	Ein Softwarefehler im Sinne des Gesetzes .....	301
18.2	Vertragliche Gewährleistung und Haftung .....	302
18.3	Vertragliche Beschränkung der Haftung .....	303
18.4	Produzentenhaftung bei Software .....	304
18.5	Produkthaftung .....	304
18.6	Sorgfaltspflicht des Software-Herstellers .....	306
18.7	Technische Normen mit Bezug zum Softwaretest .....	307
18.7.1	DIN IEC 56/575/CD .....	307
18.7.2	IEEE Std 1012 .....	308
18.7.3	IEEE Std 829 .....	308
18.7.4	IEEE Std 1008-1987 .....	309
18.7.5	ISO/IEC 29119 .....	309
18.7.6	IEC/EN 61508 .....	309
18.7.7	ISO 26262 .....	312
18.7.8	Normenreihe 250XX .....	314
18.8	Tipps vom Rechtsanwalt und vom Techniker .....	316
18.9	Fragen und Übungsaufgaben .....	318

<b>Nachwort</b>	<b>319</b>
<b>Anhang</b>	
<b>Anhang A – Lösungen zu Übungsaufgaben</b>	<b>323</b>
<b>Anhang B – Dokumentation des Testdesigns</b>	<b>349</b>
<b>Anhang C – Software-Verifikationsplan</b>	<b>351</b>
<b>Anhang D – Software-Verifikations-Report</b>	<b>355</b>
<b>Quellenverzeichnis</b>	<b>357</b>
<b>Index</b>	<b>367</b>