

| Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	2
1.2	Terminologie und Begriffsdefinitionen	5
1.3	Stand der Technik	10
1.3.1	Qualitätsmanagement	10
1.3.1.1	Kontinuierliche Ansätze	13
1.3.1.2	Modellbasierte Ansätze	16
1.3.1.3	Unterstützende Techniken	28
1.3.2	Software-Qualitätssicherung	30
1.3.3	Hardware-Qualitätssicherung	32
1.3.4	Qualitätssicherung softwareintensiver Systeme	35
1.4	Einordnung und Klassifikation der Prüftechniken	37
1.4.1	Dynamischer Test	39
1.4.2	Statische Analyse	43
1.4.3	Formale Techniken: Symbolischer Test und formale Beweisverfahren	44
1.5	Organisation	46
	Checkliste	48
2	Funktionsorientierter Test	49
2.1	Eigenschaften und Ziele des funktionsorientierten Tests	50
2.2	Funktionale Äquivalenzklassenbildung	51
2.2.1	Eigenschaften und Ziele der funktionalen Äquivalenzklassenbildung	51
2.2.2	Beschreibung der funktionalen Äquivalenzklassenbildung	52
2.2.3	Bewertung der funktionalen Äquivalenzklassenbildung	57
2.3	Zustandsbasierter Test	58
2.3.1	Eigenschaften und Ziele des zustandsbasierten Tests	58
2.3.2	Beschreibung des zustandsbasierten Tests	58
2.3.3	Bewertung des zustandsbasierten Tests	64
2.4	Ursache-Wirkungs-Analyse	66
2.5	Weitere funktionsorientierte Testtechniken	73
2.5.1	Syntaxtest	73
2.5.2	Transaktionsflussbasiertes Testen	75
2.5.3	Test auf Basis von Entscheidungstabellen oder Entscheidungsbäumen	79
2.6	Bewertung des funktionsorientierten Tests	81
	Checkliste	81
3	Kontrollflussorientierter, strukturorientierter Test	83
3.1	Eigenschaften und Ziele des kontrollflussorientierten Tests	84
3.2	Anweisungsüberdeckungstest	85
3.2.1	Eigenschaften und Ziele des Anweisungsüberdeckungstests	85
3.2.2	Beschreibung des Anweisungsüberdeckungstests	86
3.2.3	Bewertung des Anweisungsüberdeckungstests	87
3.3	Zweigüberdeckungstest	88

3.3.1	Eigenschaften und Ziele des Zweigüberdeckungstests	88
3.3.2	Beschreibung des Zweigüberdeckungstests	88
3.3.3	Problematiken des Zweigüberdeckungstests	90
3.3.4	Bewertung des Zweigüberdeckungstests	91
3.4	Bedingungsüberdeckungstest	93
3.4.1	Eigenschaften und Ziele des Bedingungsüberdeckungstests	93
3.4.2	Einfacher Bedingungsüberdeckungstest	95
3.4.3	Bedingungs-/Entscheidungsüberdeckungstest	99
3.4.4	Minimaler Mehrfach-Bedingungsüberdeckungstest	101
3.4.5	Modifizierter Bedingungs-/Entscheidungsüberdeckungstest	106
3.4.6	Mehrfach-Bedingungsüberdeckungstest	111
3.4.7	Problematiken	114
3.4.8	Bewertung des Bedingungsüberdeckungstests	117
3.5	Techniken für den Test von Schleifen	117
3.5.1	Eigenschaften und Ziele	117
3.5.2	Strukturierter Pfadtest und <i>boundary interior</i> -Pfadtest	118
3.5.2.1	Beschreibung	118
3.5.2.2	Modifizierte <i>boundary interior</i> -Testtechnik	124
3.5.2.3	Bewertung	128
3.5.3	LCSAJ-Test	132
3.5.3.1	Eigenschaften und Ziele des LCSAJ-Tests	132
3.5.3.2	Erweiterungen	134
3.5.3.3	Bewertung des LCSAJ-Tests	136
3.6	Pfadüberdeckungstest	136
3.6.1	Eigenschaften und Ziele des Pfadüberdeckungstests	136
3.6.2	Bewertung des Pfadüberdeckungstests	137
3.7	Bewertung des kontrollflussorientierten Tests	138
	Checkliste	138
4	Datenflussorientierter, strukturorientierter Test	141
4.1	Eigenschaften und Ziele des datenflussorientierten Tests	142
4.2	<i>Defs/Uses</i> -Test	144
4.3	<i>Required k-Tuples</i> Test	163
4.4	Datenkontext-Überdeckung	171
4.5	Bewertung des datenflussorientierten Tests	177
	Checkliste	178
5	Spezielle dynamische Testtechniken	179
5.1	Diversifizierender Test	180
5.1.1	Eigenschaften und Ziele des diversifizierenden Tests	180
5.1.2	<i>Back to Back</i> -Test	180
5.1.2.1	Eigenschaften und Ziele des <i>Back to Back</i> -Tests	180
5.1.2.2	Beschreibung des <i>Back to Back</i> -Tests	181
5.1.2.3	Bewertung des <i>Back to Back</i> -Tests	183
5.1.3	Mutationen-Test	185
5.1.3.1	Eigenschaften und Ziele des Mutationen-Tests	185

5.1.3.2	Beschreibung des Mutationen-Tests	185
5.1.3.3	Bewertung des Mutationen-Tests	191
5.1.4	Regressionstest	192
5.1.4.1	Eigenschaften und Ziele des Regressionstests	192
5.1.4.2	Beschreibung des Regressionstests	192
5.1.4.3	Bewertung des Regressionstests	194
5.1.5	Bewertung des diversifizierenden Tests	195
5.2	Bereichstest (<i>Domain Testing</i>)	195
5.2.1	Eigenschaften und Ziele der Bereichstests	195
5.2.2	Pfadbereichstest	196
5.2.3	Test fehleroffenbarer Unterbereiche	203
5.2.4	Partition-Analyse	206
5.2.5	Bewertung der Bereichstests	208
5.3	Zufallstest	208
5.4	<i>Error guessing</i>	210
5.5	Verwendung von Zusicherungen	211
5.6	Bewertung	213
	Checkliste	214
6	Modellbasiertes Testen	215
6.1	Eigenschaften und Ziele des modellbasierten Tests	216
6.2	Beschreibung des modellbasierten Tests	218
6.2.1	Charakterisierung verschiedener Modelltypen	218
6.2.1.1	Zustandsautomaten	220
6.2.1.2	Vor- und Nachbedingungen	221
6.2.1.3	Temporallogiken	223
6.2.1.4	Markov-Ketten	225
6.2.1.5	Prozesskalküle	227
6.3	Bewertung des modellbasierten Tests	229
	Checkliste	230
7	Software-Messung	231
7.1	Eigenschaften und Ziele der Software-Messung	232
7.2	Maße und Metriken	233
7.3	Maßtypen	234
7.4	Forderungen an Maße	236
7.5	Maßskalen	237
7.5.1	Grundlagen	237
7.5.2	Skalendiskussion	240
7.5.2.1	Die Ordinalskala	240
7.5.2.2	Die Rationalskala	241
7.5.2.3	Die empirische Relation	242
7.6	Datenerfassung für Maßsysteme	244
7.7	Zielgerichtete Definition von Maßen	245
7.8	Auswertung von Messungen	245
7.8.1	Darstellung von Messwerten	247

7.8.2	Auswertung mit Erfahrungswissen	251
7.8.3	Auswertung mit statistischen Techniken	253
7.9	Wichtige Maße für Software	255
7.9.1	Die zyklomatische Komplexität	256
7.9.2	Die Halstead-Maße	260
7.9.3	Das Maß <i>Live Variables</i>	262
7.9.4	Das Maß „Variablenspanne“	263
7.9.5	Die MTBF	263
7.10	Fallstudie zur Software-Messung	263
7.11	Bewertung der Software-Messung	266
	Checkliste	268
8	Werkzeugunterstützte statische Codeanalyse	269
8.1	Eigenschaften und Ziele der werkzeugunterstützten statischen Codeanalyse	270
8.2	Stilanalyse	271
8.2.1	Eigenschaften und Ziele der Stilanalyse	271
8.2.2	Prüfung der Einhaltung von Programmierkonventionen	272
8.2.3	Bewertung der Stilanalyse	275
8.3	Diagramme und Tabellen	276
8.3.1	Eigenschaften und Ziele der Nutzung von Diagrammen und Tabellen	276
8.3.2	Diagramme	276
8.3.2.1	Kontrollflussgraph	277
8.3.2.2	Programmablaufplan	281
8.3.2.3	Nassi-Shneiderman-Diagramm	282
8.3.2.4	Strukturdiagramm	282
8.3.3	Tabellen	284
8.3.4	Bewertung der Nutzung von Diagrammen und Tabellen	285
8.4	<i>Slicing</i>	285
8.4.1	Eigenschaften und Ziele des <i>Slicings</i>	285
8.4.2	Statisches <i>Slicing</i>	286
8.4.3	Dynamisches <i>Slicing</i>	289
8.4.4	Bewertung des <i>Slicings</i>	291
8.5	Datenflussanomalieanalyse	292
8.5.1	Eigenschaften und Ziele der Datenflussanomalieanalyse	292
8.5.2	Durchführung der Datenflussanomalieanalyse	293
8.5.3	Mögliche Probleme der Datenflussanomalieanalyse und ihre Behebung	298
8.5.4	Bewertung der Datenflussanomalieanalyse	302
8.6	Bewertung der werkzeugunterstützten statischen Codeanalyse	303
	Checkliste	303
9	Software-Inspektionen und Reviews	305
9.1	Eigenschaften und Ziele von Software-Inspektionen und Reviews	306
9.2	Formale Inspektionstechniken	308
9.2.1	Eigenschaften und Ziele der formalen Inspektionstechniken	308
9.2.2	Beschreibung der formalen Inspektionstechniken	308
9.2.3	Bewertung der formalen Inspektionstechniken	315

9.3	Konventionelles Review in Sitzungstechnik: <i>Structured Walkthrough</i>	317
9.4	Review in Kommentartechnik	318
9.5	Bewertung von Software-Inspektionen und Reviews	318
	Checkliste	319
10	Formale Techniken: Symbolischer Test und formaler Korrektheitsbeweis	321
10.1	Eigenschaften und Ziele der formalen Techniken	322
10.2	Symbolischer Test	322
10.2.1	Eigenschaften und Ziele des symbolischen Tests	322
10.2.2	Beschreibung des symbolischen Tests	325
10.2.3	Bewertung des symbolischen Tests	333
10.3	Formaler Korrektheitsbeweis	335
10.3.1	Eigenschaften und Ziele des formalen Korrektheitsbeweises	335
10.3.2	Zusicherungsverfahren	335
10.3.2.1	Das Floyd'sche Verifikationsverfahren	335
10.3.2.2	Der Hoare-Kalkül	344
10.3.3	Totale Korrektheit	349
10.3.4	Algebraische Techniken	351
10.3.5	Automatenbasierte Techniken	354
10.3.6	Bewertung des formalen Korrektheitsbeweises	357
10.4	Bewertung der formalen Techniken	358
	Checkliste	360
11	Prozesse und Prüfstrategien	361
11.1	Eigenschaften und Ziele	362
11.2	Software-Entwicklungsprozesse	362
11.3	Die Entwicklung	364
11.3.1	Die Analyse	368
11.3.2	Der Entwurf	369
11.3.3	Die Implementierung	370
11.4	Die Prüfung	370
11.4.1	Die Modulprüfung	371
11.4.2	Die Integration und die Integrationsprüfung	372
11.4.3	Die Systemprüfung	376
11.5	Organisatorische Aspekte	378
11.6	Dokumentation und Auswertung der Prüfung	381
11.7	Standards	382
11.7.1	Bedeutung von Standards	382
11.7.2	Prozessorientierte Standards	385
11.7.2.1	DIN EN ISO 9001 und V-Modell	385
11.7.2.2	ISO/IEC 15504: SPICE	385
11.7.2.3	AQAP-Century-Standards	386
11.7.3	Anwendungsbereichsunabhängige technische Standards: Der Standard IEC 61508	386
11.7.4	Anwendungsbereichsspezifische technische Standards	387
11.7.4.1	DIN EN 50128	387
11.7.4.2	RTCA/DO 178B	388

11.8	Bewertung	389
	Checkliste	389
12	Werkzeuge	391
12.1	Eigenschaften und Ziele der Nutzung von Werkzeugen	392
12.2	Werkzeugtypen	393
12.2.1	Dynamische Testwerkzeuge	393
12.2.2	Statische Analysewerkzeuge	397
12.2.3	Formale Verifikationswerkzeuge	398
12.2.4	Modellierende und analysierende Werkzeuge	399
12.3	Verfügbarkeit von Werkzeugen	401
12.3.1	Abdeckung von Techniken durch Werkzeuge	401
12.3.2	Abdeckung von Programmiersprachen durch Werkzeuge	402
12.3.3	Abdeckung von Entwicklungs- und Zielplattformen durch Werkzeuge	402
12.4	Informationsquellen über Werkzeuge	403
12.5	Bewertung der Nutzung von Werkzeugen	403
	Checkliste	404
13	Prüfen von objektorientierter Software	407
13.1	Eigenschaften und Ziele des Prüfens von objektorientierter Software	408
13.2	Hinweise für die objektorientierte Entwicklung	410
13.3	Objektorientierter Modultest	411
13.3.1	Klassentest als objektorientierter Modultest	412
13.3.2	Ein Ansatz für die Überprüfung von Klassen	413
13.3.3	Funktionsorientierter Test	415
13.3.3.1	Zustandsbasierter Test von Operationssequenzen	415
13.3.3.2	Funktionsorientierte Äquivalenzklassenbildung für den Test von Operationen	417
13.3.4	Strukturorientierter Test	417
13.3.4.1	Kontrollflussorientierter Test	419
13.3.4.2	Datenflussorientierter Test	421
13.3.5	Formale Spezifikationen zur Unterstützung des objektorientierten Prüfens	422
13.3.6	Test von parametrisierten Klassen	426
13.3.7	Test von Unterklassen und Regressionstests	427
13.4	Objektorientierter Integrationstest	428
13.4.1	Integrationstest von Basisklassen	428
13.4.2	Integrationstest und Vererbung	430
13.4.2.1	Integrationstest von Vererbung bei dienst anbietenden Klassen	431
13.4.2.2	Integrationstest von Vererbung bei dienst nutzenden Klassen	432
13.4.2.3	Integrationstest von Vererbung bei dienst nutzenden und dienst anbietenden Klassen ...	433
13.4.2.4	Integrationstest und Testumgebungen	434
13.5	Objektorientierter Systemtest	435
13.6	Bewertung des Prüfens von objektorientierter Software	437
	Checkliste	438
14	Prüfen von eingebetteter Software	439
14.1	Eigenschaften und Ziele des Prüfens von eingebetteter Software	440
14.2	Wichtige Eigenschaften von eingebetteter Software	440

14.2.1	Sicherheitskritikalität	440
14.2.2	Zuverlässigkeit und Verfügbarkeit	441
14.2.3	Echtzeitfähigkeit	443
14.3	Dynamisches Testen von sicherheitskritischer Software	443
14.4	Sicherheits- und Zuverlässigkeitsmodellierung	445
14.4.1	Eigenschaften und Ziele der Sicherheits- und Zuverlässigkeitsmodellierung	445
14.4.2	Software-FMECA	446
14.4.3	Fehlerbaumanalyse	447
14.4.4	Markov-Modellierung	451
14.4.5	Bewertung der Sicherheits- und Zuverlässigkeitsmodellierung	452
14.5	Stochastische Software-Zuverlässigkeitsanalyse	453
14.5.1	Eigenschaften und Ziele der stochastischen Software-Zuverlässigkeitsanalyse	453
14.5.2	Grundlagen der stochastischen Zuverlässigkeitsanalyse	454
14.5.3	Hardware- und Software-Zuverlässigkeitsanalyse im Vergleich	460
14.5.4	Software-Zuverlässigkeitsmodelle	463
14.5.4.1	Bestimmung von Modellparametern	463
14.5.4.2	Modellauswahl	467
14.5.5	Beispiel eines Modells: Musas elementares Ausführungszeiten-Modell	471
14.5.5.1	Modellbildung	472
14.5.5.2	Beispiele für die Modellierung	475
14.5.6	Bewertung der stochastischen Software-Zuverlässigkeitsanalyse	476
14.6	Bewertung des Prüfens von eingebetteter Software	478
	Checkliste	479
15	Ein Praxisleitfaden	481
15.1	Organisatorische Hinweise	482
15.2	Technische Hinweise	483
15.2.1	Eine einfache praxisgeeignete Prüfstrategie	485
15.2.2	Beachtung spezieller Anforderungen	487
15.3	Zusammenfassung	489
	Checkliste	490
	Literaturverzeichnis	493
	Glossar	509
	Index	521



<http://www.springer.com/978-3-8274-2056-5>

Software-Qualität

Testen, Analysieren und Verifizieren von Software

Liggesmeyer, P.

2009, XV, 526 S., Hardcover

ISBN: 978-3-8274-2056-5