

| Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	2
1.2	Terminologie und Begriffsdefinitionen	5
1.3	Stand der Technik	11
1.3.1	Qualitätsmanagement	11
1.3.2	Software-Qualitätssicherung	27
1.3.3	Hardware-Qualitätssicherung	30
1.3.4	Qualitätssicherung softwareintensiver Systeme	32
1.4	Einordnung und Klassifikation der Prüftechniken	33
1.4.1	Dynamischer Test	35
1.4.2	Statische Analyse	40
1.4.3	Formale Techniken: Symbolischer Test und formale Beweisverfahren	41
1.5	Organisation	43
	Checkliste	44
2	Funktionsorientierter Test	45
2.1	Eigenschaften und Ziele des funktionsorientierten Tests	46
2.2	Funktionale Äquivalenzklassenbildung	47
2.2.1	Eigenschaften und Ziele der funktionalen Äquivalenzklassenbildung	47
2.2.2	Beschreibung der funktionalen Äquivalenzklassenbildung	48
2.2.3	Bewertung der funktionalen Äquivalenzklassenbildung	53
2.3	Zustandsbasierter Test	54
2.3.1	Eigenschaften und Ziele des zustandsbasierten Tests	54
2.3.2	Beschreibung des zustandsbasierten Tests	54
2.3.3	Bewertung des zustandsbasierten Tests	60
2.4	Ursache-Wirkungs-Analyse	62
2.5	Weitere funktionsorientierte Testtechniken	69
2.5.1	Syntaxtest	69
2.5.2	Transaktionsflussbasiertes Testen	73
2.5.3	Test auf Basis von Entscheidungstabellen oder Entscheidungsbäumen	75
2.6	Bewertung des funktionsorientierten Tests	77
	Checkliste	77

3	Kontrollflussorientierter, strukturorientierter Test	79
3.1	Eigenschaften und Ziele des kontrollflussorientierten Tests	80
3.2	Anweisungsüberdeckungstest	81
3.2.1	Eigenschaften und Ziele des Anweisungsüberdeckungstests	81
3.2.2	Beschreibung des Anweisungsüberdeckungstests	82
3.2.3	Bewertung des Anweisungsüberdeckungstests	83
3.3	Zweigüberdeckungstest	84
3.3.1	Eigenschaften und Ziele des Zweigüberdeckungstests	84
3.3.2	Beschreibung des Zweigüberdeckungstests	84
3.3.3	Problematiken des Zweigüberdeckungstests	86
3.3.4	Bewertung des Zweigüberdeckungstests	88
3.4	Bedingungsüberdeckungstest	89
3.4.1	Eigenschaften und Ziele des Bedingungsüberdeckungstests	89
3.4.2	Einfacher Bedingungsüberdeckungstest	91
3.4.3	Bedingungs-/Entscheidungsüberdeckungstest	95
3.4.4	Minimaler Mehrfach-Bedingungsüberdeckungstest	96
3.4.5	Modifizierter Bedingungs-/Entscheidungsüberdeckungstest	101
3.4.6	Mehrfach-Bedingungsüberdeckungstest	107
3.4.7	Problematiken	109
3.4.8	Bewertung des Bedingungsüberdeckungstests	112
3.5	Techniken für den Test von Schleifen	112
3.5.1	Eigenschaften und Ziele	112
3.5.2	Strukturierter Pfadtest und <i>boundary interior</i> -Pfadtest	113
3.5.2.1	Beschreibung	113
3.5.2.2	Modifizierte <i>boundary interior</i> -Testtechnik	120
3.5.2.3	Bewertung	123
3.5.3	LCSAJ-Test	127
3.5.3.1	Eigenschaften und Ziele des LCSAJ-Tests	127
3.5.3.2	Erweiterungen	129
3.5.3.3	Bewertung des LCSAJ-Tests	131
3.6	Pfadüberdeckungstest	132
3.6.1	Eigenschaften und Ziele des Pfadüberdeckungstests	132
3.6.2	Bewertung des Pfadüberdeckungstests	133
3.7	Bewertung des kontrollflussorientierten Tests	133

Checkliste	134
------------------	-----

4	Datenflussorientierter, strukturorientierter Test	137
----------	--	------------

4.1	Eigenschaften und Ziele des datenflussorientierten Tests	138
-----	--	-----

4.2	<i>Defs/Uses</i> -Test	141
-----	------------------------------	-----

4.3	<i>Required k-Tuples</i> Test	158
-----	-------------------------------------	-----

4.4	Datenkontext-Überdeckung	164
-----	--------------------------------	-----

4.5	Bewertung des datenflussorientierten Tests	169
-----	--	-----

Checkliste	171
------------------	-----

5	Spezielle dynamische Testtechniken	173
----------	---	------------

5.1	Diversifizierender Test	174
-----	-------------------------------	-----

5.1.1	Eigenschaften und Ziele des diversifizierenden Tests	174
-------	--	-----

5.1.2	<i>Back to Back</i> -Test	174
-------	---------------------------------	-----

5.1.2.1	Eigenschaften und Ziele des <i>Back to Back</i> -Tests	174
---------	--	-----

5.1.2.2	Beschreibung des <i>Back to Back</i> -Tests	175
---------	---	-----

5.1.2.3	Bewertung des <i>Back to Back</i> -Tests	178
---------	--	-----

5.1.3	Mutationen-Test	179
-------	-----------------------	-----

5.1.3.1	Eigenschaften und Ziele des Mutationen- Tests	179
---------	---	-----

5.1.3.2	Beschreibung des Mutationen-Tests	180
---------	---	-----

5.1.3.3	Bewertung des Mutationen-Tests	186
---------	--------------------------------------	-----

5.1.4	Regressionstest	187
-------	-----------------------	-----

5.1.4.1	Eigenschaften und Ziele des Regressionstests	187
---------	--	-----

5.1.4.2	Beschreibung des Regressionstests	187
---------	---	-----

5.1.4.3	Bewertung des Regressionstests	189
---------	--------------------------------------	-----

5.1.5	Bewertung des diversifizierenden Tests	189
-------	--	-----

5.2	Bereichstest (<i>Domain Testing</i>)	190
-----	--	-----

5.2.1	Eigenschaften und Ziele der Bereichstests	190
-------	---	-----

5.2.2	Pfadbereichstest	191
-------	------------------------	-----

5.2.3	Test fehleroffenbarer Unterbereiche	197
-------	---	-----

5.2.4	Partition-Analyse	201
-------	-------------------------	-----

5.2.5	Bewertung der Bereichstests	202
-------	-----------------------------------	-----

5.3	Zufallstest	203
-----	-------------------	-----

5.4	<i>Error guessing</i>	204
-----	-----------------------------	-----

5.5	Verwendung von Zusicherungen	205
-----	------------------------------------	-----

5.6	Bewertung	208
	Checkliste	208
6	Software-Messung	209
6.1	Eigenschaften und Ziele der Software-Messung	210
6.2	Maße und Metriken	212
6.3	Maßtypen	212
6.4	Forderungen an Maße	214
6.5	Maßskalen	216
6.5.1	Grundlagen	216
6.5.2	Skalendiskussion	218
6.5.2.1	Die Ordinalskala	219
6.5.2.2	Die Rationalskala	219
6.5.2.3	Die empirische Relation	220
6.6	Datenerfassung für Maßsysteme	223
6.7	Zielgerichtete Definition von Maßen	224
6.8	Auswertung von Messungen	225
6.8.1	Darstellung von Messwerten	227
6.8.2	Auswertung mit Erfahrungswissen	231
6.8.3	Auswertung mit statistischen Techniken	232
6.9	Wichtige Maße für Software	235
6.9.1	Die zyklomatische Komplexität	236
6.9.2	Die Halstead-Maße	240
6.9.3	Das Maß <i>Live Variables</i>	242
6.9.4	Das Maß „Variablenspanne“	243
6.9.5	Die MTBF	243
6.10	Fallstudie zur Software-Messung	244
6.11	Bewertung der Software-Messung	247
	Checkliste	248
7	Werkzeugunterstützte statische Codeanalyse	249
7.1	Eigenschaften und Ziele der werkzeugunterstützten statischen Codeanalyse	250
7.2	Stilanalyse	251
7.2.1	Eigenschaften und Ziele der Stilanalyse	251
7.2.2	Prüfung der Einhaltung von Programmierkonventionen	252

7.2.3	Bewertung der Stilanalyse	255
7.3	Diagramme und Tabellen	256
7.3.1	Eigenschaften und Ziele der Nutzung von Diagrammen und Tabellen	256
7.3.2	Diagramme	256
7.3.2.1	Kontrollflussgraph	257
7.3.2.2	Programmablaufplan	261
7.3.2.3	Nassi-Shneiderman-Diagramm	262
7.3.2.4	Strukturdiagramm	263
7.3.3	Tabellen	264
7.3.4	Bewertung der Nutzung von Diagrammen und Tabellen	265
7.4	<i>Slicing</i>	266
7.4.1	Eigenschaften und Ziele des <i>Slicings</i>	266
7.4.2	Statisches <i>Slicing</i>	266
7.4.3	Dynamisches <i>Slicing</i>	270
7.4.4	Bewertung des <i>Slicings</i>	272
7.5	Datenflussanomalieanalyse	272
7.5.1	Eigenschaften und Ziele der Datenflussanomalieanalyse	272
7.5.2	Durchführung der Datenflussanomalieanalyse	274
7.5.3	Mögliche Probleme der Datenflussanomalieanalyse und ihre Behebung	279
7.5.4	Bewertung der Datenflussanomalieanalyse	283
7.6	Bewertung der werkzeugunterstützten statischen Codeanalyse	284
	Checkliste	284
8	Software-Inspektionen und Reviews	285
8.1	Eigenschaften und Ziele von Software-Inspektionen und Reviews	286
8.2	Formale Inspektionstechniken	288
8.2.1	Eigenschaften und Ziele der formalen Inspektionstechniken	288
8.2.2	Beschreibung der formalen Inspektionstechniken	289
8.2.3	Bewertung der formalen Inspektionstechniken	295
8.3	Konventionelles Review in Sitzungstechnik: <i>Structured Walkthrough</i>	297
8.4	Review in Kommentartechnik	298
8.5	Bewertung von Software-Inspektionen und Reviews	299
	Checkliste	300

9	Formale Techniken: Symbolischer Test und formaler Korrektheitsbeweis	301
9.1	Eigenschaften und Ziele der formalen Techniken	302
9.2	Symbolischer Test	302
9.2.1	Eigenschaften und Ziele des symbolischen Tests	302
9.2.2	Beschreibung des symbolischen Tests	306
9.2.3	Bewertung des symbolischen Tests	314
9.3	Formaler Korrektheitsbeweis	316
9.3.1	Eigenschaften und Ziele des formalen Korrektheitsbeweises	316
9.3.2	Zusicherungsverfahren	316
9.3.2.1	Das Floyd'sche Verifikationsverfahren	316
9.3.2.2	Der Hoare-Kalkül	325
9.3.2.3	Totale Korrektheit	331
9.3.3	Algebraische Techniken	332
9.3.4	Automatenbasierte Techniken	336
9.3.5	Bewertung des formalen Korrektheitsbeweises	339
9.4	Bewertung der formalen Techniken	341
	Checkliste	342
10	Prozesse und Prüfstrategien	343
10.1	Eigenschaften und Ziele	344
10.2	Software-Entwicklungsprozesse	345
10.3	Die Entwicklung	347
10.3.1	Die Analyse	350
10.3.2	Der Entwurf	351
10.3.3	Die Implementierung	352
10.4	Die Prüfung	352
10.4.1	Die Modulprüfung	353
10.4.2	Die Integration und die Integrationsprüfung	354
10.4.3	Die Systemprüfung	359
10.5	Organisatorische Aspekte	360
10.6	Dokumentation und Auswertung der Prüfung	363
10.7	Standards	365
10.7.1	Bedeutung von Standards	365
10.7.2	Prozessorientierte Standards	368

10.7.2.1	DIN EN ISO 9001 und V-Modell	368
10.7.2.2	ISO/IEC TR 15504: SPICE	368
10.7.2.3	AQAP-Century-Standards	369
10.7.3	Anwendungsbereichsunabhängige technische Standards: Der Standard IEC 61508	369
10.7.4	Anwendungsbereichsspezifische technische Standards	371
10.7.4.1	DIN EN 50128 und Mü 8004	371
10.7.4.2	RTCA/DO 178B	372
10.8	Bewertung	373
	Checkliste	373
11	Werkzeuge	375
11.1	Eigenschaften und Ziele der Nutzung von Werkzeugen	376
11.2	Werkzeugtypen	377
11.2.1	Dynamische Testwerkzeuge	377
11.2.2	Statische Analysewerkzeuge	381
11.2.3	Formale Verifikationswerkzeuge	383
11.2.4	Modellierende und analysierende Werkzeuge	384
11.3	Verfügbarkeit von Werkzeugen	385
11.3.1	Abdeckung von Techniken durch Werkzeuge	386
11.3.2	Abdeckung von Programmiersprachen durch Werkzeuge	386
11.3.3	Abdeckung von Entwicklungs- und Zielplattformen durch Werkzeuge	387
11.4	Informationsquellen über Werkzeuge	387
11.5	Bewertung der Nutzung von Werkzeugen	388
	Checkliste	389
12	Prüfen von objektorientierter Software	391
12.1	Eigenschaften und Ziele des Prüfens von objektorientierter Software	392
12.2	Hinweise für die objektorientierte Entwicklung	394
12.3	Objektorientierter Modultest	395
12.3	Klassentest als objektorientierter Modultest	396
12.3.2	Ein Ansatz für die Überprüfung von Klassen	398
12.3.3	Funktionsorientierter Test	399
12.3.3.1	Zustandsbasierter Test von Operationssequenzen	400
12.3.3.2	Funktionsorientierte Äquivalenzklassenbildung für den Test von Operationen	402
12.3.4	Strukturorientierter Test	403

12.3.4.1	Kontrollflussorientierter Test	404
12.3.4.2	Datenflussorientierter Test	409
12.3.5	Formale Spezifikationen zur Unterstützung des objektorientierten Prüfens	410
12.3.6	Test von parametrisierten Klassen	411
12.3.7	Test von Unterklassen und Regressionstests	412
12.4	Objektorientierter Integrationstest	413
12.4.1	Integrationstest von Basisklassen	413
12.4.2	Integrationstest und Vererbung	415
12.4.2.1	Integrationstest von Vererbung bei dienst anbietenden Klassen	416
12.4.2.2	Integrationstest von Vererbung bei dienst nutzenden Klassen	418
12.4.2.3	Integrationstest von Vererbung bei dienst nutzenden und dienst anbietenden Klassen ...	419
12.4.2.4	Integrationstest und Testumgebungen	419
12.5	Objektorientierter Systemtest	420
12.6	Bewertung des Prüfens von objektorientierter Software	423
	Checkliste	423
13	Prüfen von eingebetteter Software	425
13.1	Eigenschaften und Ziele des Prüfens von eingebetteter Software	426
13.2	Wichtige Eigenschaften von eingebetteter Software	426
13.2.1	Sicherheitskritikalität	426
13.2.2	Zuverlässigkeit und Verfügbarkeit	428
13.2.3	Echtzeitfähigkeit	429
13.3	Dynamisches Testen von sicherheitskritischer Software	430
13.4	Sicherheits- und Zuverlässigkeitsmodellierung	432
13.4.1	Eigenschaften und Ziele der Sicherheits- und Zuverlässigkeitsmodellierung	432
13.4.2	Software-FMECA	433
13.4.3	Fehlerbaumanalyse	434
13.4.4	Markov-Modellierung	438
13.4.5	Bewertung der Sicherheits- und Zuverlässigkeitsmodellierung	440
13.5	Stochastische Software-Zuverlässigkeitsanalyse	440
13.5.1	Eigenschaften und Ziele der stochastischen Software-Zuverlässigkeitsanalyse	440
13.5.2	Grundlagen der stochastischen Zuverlässigkeitsanalyse	442
13.5.3	Hardware- und Software-Zuverlässigkeitsanalyse im Vergleich	448
13.5.4	Software-Zuverlässigkeitsmodelle	451
13.5.4.1	Bestimmung von Modellparametern	451

13.5.4.2	Modellauswahl	456
13.5.5	Beispiel eines Modells: Musas elementares Ausführungszeiten-Modell	461
13.5.5.1	Modellbildung	461
13.5.5.2	Beispiele für die Modellierung	465
13.5.6	Bewertung der stochastischen Software-Zuverlässigkeitsanalyse	465
13.6	Bewertung des Prüfens von eingebetteter Software	468
	Checkliste	469
14	Ein Praxisleitfaden	471
14.1	Organisatorische Hinweise	472
14.2	Technische Hinweise	473
14.2.1	Eine einfache praxisgeeignete Prüfstrategie	476
14.2.2	Beachtung spezieller Anforderungen	477
14.3	Zusammenfassung	479
	Checkliste	481
	Literaturverzeichnis	483
	Glossar	501
	Index	517

Übersicht

1.1	Motivation	2
1.2	Terminologie und Begriffsdefinitionen	5
1.3	Stand der Technik	11
1.4	Einordnung und Klassifikation der Prüftechniken	33
1.5	Organisation	43
	Checkliste	44