

Peter Liggesmeyer

Qualitätssicherung  
softwareintensiver  
technischer Systeme

Spektrum Akademischer Verlag Heidelberg · Berlin

# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>1</b>
1.1 Motivation	1
1.2 Begriffsdefinitionen	4
1.2.1 Grundlegende Begriffe der Qualitätssicherung	4
Qualität	4
Qualitätsanforderung	5
Qualitätsmerkmal	5
Qualitätsmaß	5
System, technisches System	5
Fehlverhalten	6
Fehler	6
Irrtum	6
1.2.2 Funktionale Qualitätseigenschaften	6
Korrektheit	8
Vollständigkeit	9
1.2.3 Nicht-funktionale Qualitätseigenschaften	9
Sicherheit	9
Zuverlässigkeit	9
Verfügbarkeit	10
Robustheit	11
1.2.4 Wechselwirkungen zwischen Qualitätseigenschaften	11
1.3 Stand der Technik	12
1.3.1 Qualitätsmanagement	12
<i>Total Quality Management</i>	12
<i>Prozeß-Assessments</i>	18
1.3.2 Software-Qualitätssicherung	27
1.3.3 Hardware-Qualitätssicherung	33
1.3.4 Qualitätssicherung softwareintensiver, hybrider Systeme	35
<b>2 Zielsetzung und Vorgehensweise</b>	<b>37</b>
2.1 Systemarten und abgeleitete Qualitätsanforderungen	37
2.1.1 Serienprodukte im Vergleich zu Einzelentwicklungen	38
2.1.2 Eigenentwicklungen im Vergleich zu Fremdentwicklungen	39
2.1.3 Systemumfang	40
2.1.4 Verteilte Systeme im Vergleich zu lokalen Systemen	41
2.1.5 Gewartete im Vergleich zu nicht gewarteten Systemen	41
2.1.6 Investitionsgüter im Vergleich zu Wegwerfprodukten	42

2.1.7 Anwendungsbereiche mit Personengefährdung	43
2.1.8 Reaktive Systeme im Vergleich zu <i>Offline</i> -Systemen	43
2.2 Forderungen an Qualitätssicherungstechniken	44
2.3 Zielsetzung	45
2.4 Software-Qualitätssicherung im Systemumfeld	47
2.5 Qualitätssicherung softwareintensiver Systeme	47
2.5.1 Sicherheits- und Zuverlässigkeitsmodellierung von Systemen	47
2.5.2 Stochastische Zuverlässigkeitsanalyse von Systemen	48
2.5.3 Statistische Interpretation und Optimierung von Messungen in der System-Entwicklung	48
2.5.4 Testen hybrider Systeme	49
<b>3 Software-Qualitätssicherung im Systemumfeld</b>	<b>51</b>
3.1 Motivation	51
3.2 Automatisierte Auswahl von Prüfverfahren	52
3.2.1 Motivation und Lösungsansatz	52
3.2.2 Systematische Bewertung der Eignung von Prüfverfahren	53
Bewertung auf Basis technischer Voraussetzungen	54
Bewertung auf Basis technischer und nicht-technischer Ziele	58
Bewertung auf Basis technischer und nicht-technischer Restriktionen	59
3.2.3 Fallstudie	59
3.2.4 Einsatzerfahrungen und Schlußfolgerungen	63
3.3 Die Prüfung von objektorientierter Software als Bestandteil von Systemen	65
3.3.1 Motivation	65
3.3.2 Objektorientierung in der System-Entwicklung	66
Beherrschung umfangreicher, komplexer Systeme	66
Sicherheit und Zuverlässigkeit	66
Echtzeitanforderungen	67
3.3.3 Objektorientiertes Prüfen mit klassischen Prüfverfahren	67
Klassische Prüfverfahren	67
Anwendbarkeit klassischer Prüfverfahren bei der objektorientierten Prüfung	68
Formale Spezifikationen zur Unterstützung des objektorientierten Prüfens	69
3.3.4 Die Prüfung objektorientierter Komponenten	71
Die Klasse als kleinste prüfbare Einheit	71
Probleme beim Testen von Klassen	71
Ein Ansatz für die Überprüfung von Klassen	73
Zustandstest für die funktionsorientierte Prüfung von Methodensequenzen	73
Funktionale Äquivalenzklassenbildung für den Test von Methoden	76
Strukturorientierte Abdeckung als Vollständigkeitskriterium	78
Datenflußorientiertes Testen von Methodeninteraktionen	84
Test von Unterklassen und Regressionstests	85
Test abstrakter und parametrisierter Klassen	87

3.3.5 Objektorientierter Integrationstest	87
Integrationstest von Basisklassen	88
Integrationstest von abgeleiteten Klassen	89
Integrationstest von dienst anbietenden abgeleiteten Klassen	90
Integrationstest von dienstnutzenden abgeleiteten Klassen	92
Integrationstest dienst anbietender und dienstnutzender abgeleiteter Klassen	93
Integrationstest und Testumgebungen	93
3.3.6 Objektorientierter Systemtest	94
3.3.7 Prüfung von objektorientierten Systemen im Vergleich zur Prüfung von objektorientierter Software	96
3.3.8 Schlußfolgerungen	97
<b>4 Sicherheits- und Zuverlässigkeitsmodellierung von Systemen</b>	<b>99</b>
4.1 Motivation	99
4.2 Stand der Technik	100
4.2.1 FMECA	101
4.2.2 Zuverlässigkeitsblockdiagramme	102
Serienschaltung	102
Parallelschaltung	103
Kombinierte Serien- und Parallelschaltung	104
4.2.3 Fehlerbaumanalyse	104
Grundlagen	104
Durchführung und Auswertung	105
Ursache-Wirkungs-Graphen als Verallgemeinerung von Fehlerbäumen	108
4.2.4 Markov-Modellierung	111
4.3 Sicherheits- und Zuverlässigkeitsmodellierung hybrider Systeme mit Fehlerbäumen	113
4.4 Automatisierte Zuverlässigkeitsanalysen auf Basis zustandsendlicher Beschreibungen	114
4.4.1 Motivation	114
4.4.2 Lösungsansatz	114
4.4.3 Systemdarstellung	116
4.4.4 Voraussetzungen und Ergebnisse	116
4.4.5 Durchführung der formalen Sicherheitsanalyse	117
4.4.6 Fallstudie	120
Überblick	120
Anwendungen der formalen Sicherheitsanalyse	123
4.4.7 Schlußfolgerungen	128
4.5 Fehlerbaumgenerierung auf Basis der FMECA	129
4.5.1 Motivation	129
4.5.2 Voraussetzungen und Durchführung	129
4.5.3 Nutzen	130
4.5.4 Anwendung	131
4.6 Fehlerbaumgenerierung für Software	132

4.6.1	Motivation	132
4.6.2	Die Technik zur Generierung von Fehlerbäumen aus Software	134
	Grundlagen	134
	<i>Program Slicing</i>	134
	Fehlerbaumgenerierung auf Basis des statischen <i>Slicing</i>	138
	Aufbau von Fehlerbäumen aus Fehlerbaummustern	143
4.6.3	Nutzen	144
4.6.4	Anwendung	147
4.7	Fehlerbaumgenerierung für elektronische Schaltungen	148
4.7.1	Motivation	148
4.7.2	Durchführung	148
	Zielsetzung	148
	Fehlerbaumgenerierung mit Komponenten-Fehlerbäumen	148
	Fehlerbaumgenerierung durch Gleichungslösen	152
4.7.3	Nutzen	158
4.7.4	Anwendung	158
4.8	Sicherheits- und Zuverlässigkeitsmodellierung hybrider Systeme durch Kombination von Komponenten-Fehlerbäumen	160
4.9	Nutzen	161
<b>5</b>	<b>Stochastische Zuverlässigkeitsanalyse von Systemen</b>	<b>163</b>
5.1	Motivation und Ziele	163
5.2	Grundlagen der stochastischen Zuverlässigkeitsanalyse	164
5.3	Hardware-Zuverlässigkeitsmodellierung	169
	5.3.1 Grundlagen	169
	5.3.2 Die Weibullverteilung	170
	5.3.3 Die logarithmische Normalverteilung	171
5.4	Software-Zuverlässigkeitsmodellierung	172
	5.4.1 Software-Zuverlässigkeitsmodelle	172
	5.4.2 Bestimmung von Modellparametern	172
	Verfahren der kleinsten Fehlerquadrate	172
	<i>Maximum-Likelihood</i> -Verfahren	175
	5.4.3 Modellauswahl auf Basis von Ausfallbeobachtungen	177
	<i>U-Plot</i> und Hypothesentests	177
	<i>Prequential-Likelihood</i> -Verfahren	178
	<i>Holdout</i> -Bewertung	180
5.5	Hardware- und Software-Zuverlässigkeitsanalyse im Vergleich	181
5.6	Analyse der Zuverlässigkeit von hybriden Systemen	184
	5.6.1 Die Poissonverteilung	184
	Der ausfallfreie Fall	184
	Der ausfallbehaftete Fall	185
	5.6.2 Nutzung der Poissonverteilung zur Zuverlässigkeitsmodellierung	187
	5.6.3 Beispiel eines Modells:	
	Musas elementares Ausführungszeiten-Modell	188
	Modellbildung	189
	Beispiele für die Modellierung	194

5.7	Werkzeugunterstützte Zuverlässigkeitsmodellierung auf Basis von Ausfallbeobachtungen	196
5.7.1	Einführung	196
5.7.2	Das Zuverlässigkeitsanalysewerkzeug RAT	196
	Werkzeugbeschreibung	196
	Beispielanalyse	197
5.8	Einsatzerfahrungen	198
5.8.1	Beschreibung der Werkzeuganwendungen	198
5.8.2	Empirische Ergebnisse	201
	Modellvorauswahl	201
	Auswahlkriterien	201
	Ausfalldaten	202
	Skalen: Ausführungszeit vs. Kalenderzeit	202
	Anwendbarkeit von Software-Zuverlässigkeitsmodellen auf hybride Systeme	203
	Ergebnisqualität	204
5.9	Nutzen	204
<b>6</b>	<b>Statistische Interpretation und Optimierung von Messungen in der System-Entwicklung</b>	<b>207</b>
6.1	Motivation	207
6.2	Maße und Metriken	208
6.2.1	Grundlagen	208
6.2.2	Maßtypen	210
6.2.3	Forderungen an Maße	211
6.2.4	Maßskalen	211
6.2.5	Datenerfassung für Maßsysteme	213
6.2.6	Zielgerichtete Definition von Maßen	214
6.3	Die Berechnung statistisch abgesicherter Auswertemodelle	214
6.3.1	Geeignete statistische Techniken	214
6.3.2	Die Diskriminanzfunktion nach Fischer	215
6.3.3	Bewertung der erzielten Prognosequalität	215
6.3.4	Eine Weiterentwicklung des Fisher-Verfahrens für umfangreiche Maßsysteme	216
6.4	Eine industrielle Fallstudie	218
6.4.1	Einführung	218
6.4.2	Ausgangsdaten und Maße	218
6.4.3	Verwendete statistische Methoden	219
6.4.4	Ergebnisse der Modellermittlung	220
	Voraussetzungen	220
	Prognosemodell für Budgetüberziehungen	221
6.4.5	Interpretation der Ergebnisse	224
6.4.6	Zusammenfassung	225
6.5	Nutzen	225

<b>7 Testen hybrider Systeme</b>	<b>227</b>
7.1 Motivation	227
7.2 Testen technischer Systeme	228
7.2.1 Testtechniken für Systeme	228
7.2.2 <i>Hardware-in-the-Loop</i> -Testen	230
7.3 Nutzen	230
<b>8 Zusammenfassung und Ausblick</b>	<b>231</b>
<b>Literatur</b>	<b>235</b>
<b>Glossar</b>	<b>253</b>
<b>Personenregister</b>	<b>263</b>
<b>Sachregister</b>	<b>267</b>