

# Inhalt

<b>Geleitwort .....</b>	<b>XIII</b>
<b>Vorwort .....</b>	<b>XVI</b>
<b>Die Autoren.....</b>	<b>XVIII</b>
<b>1      Softwaremessung.....</b>	<b>1</b>
1.1    Das Wesen von Software.....	1
1.2    Sinn und Zweck der Softwaremessung.....	6
1.2.1    Zum Verständnis (Comprehension) der Software.....	7
1.2.2    Zum Vergleich der Software.....	7
1.2.3    Zur Vorhersage .....	8
1.2.4    Zur Projektsteuerung.....	8
1.2.5    Zur zwischenmenschlichen Verständigung.....	8
1.3    Dimensionen der Substanz Software .....	9
1.3.1    Quantitätsmetrik von Software .....	9
1.3.2    Komplexitätsmetrik von Software .....	10
1.3.3    Qualitätsmetrik von Software .....	10
1.4    Sichten auf die Substanz Software .....	11
1.5    Objekte der Softwaremessung .....	13
1.6    Ziele einer Softwaremessung.....	14
1.7    Zur Gliederung dieses Buches.....	17
<b>2      Softwarequantität .....</b>	<b>21</b>
2.1    Quantitätsmaße .....	21
2.2    Codegrößen .....	23
2.2.1    Codedateien .....	24
2.2.2    Codezeilen .....	25
2.2.3    Anweisungen .....	25
2.2.4    Prozeduren bzw. Methoden .....	25
2.2.5    Module bzw. Klassen.....	25
2.2.6    Entscheidungen.....	26

2.2.7	Logikzweige .....	26
2.2.8	Aufrufe .....	26
2.2.9	Vereinbarte Datenelemente .....	26
2.2.10	Benutzte Datenelemente bzw. Operanden .....	26
2.2.11	Datenobjekte.....	27
2.2.12	Datenzugriffe .....	27
2.2.13	Benutzeroberflächen.....	27
2.2.14	Systemnachrichten.....	27
2.3	Entwurfsgrößen.....	28
2.3.1	Strukturierte Entwurfsgrößen .....	28
2.3.2	Datenmodellgrößen .....	28
2.3.3	Objektmodellgrößen .....	28
2.4	Anforderungsgrößen.....	32
2.4.1	Anforderungen.....	33
2.4.2	Abnahmekriterien .....	34
2.4.3	Anwendungsfälle .....	34
2.4.4	Verarbeitungsschritte .....	34
2.4.5	Oberflächen .....	34
2.4.6	Systemschnittstellen .....	34
2.4.7	Systemakteure .....	35
2.4.8	Relevante Objekte .....	35
2.4.9	Objektzustände .....	35
2.4.10	Bedingungen.....	35
2.4.11	Aktionen .....	35
2.4.12	Testfälle.....	36
2.5	Testgrößen.....	36
2.5.1	Testfälle .....	36
2.5.2	Testfallattribute .....	37
2.5.3	Testläufe .....	37
2.5.4	Testskriptebzw. Testprozeduren .....	38
2.5.5	Testskriptzeilen .....	38
2.5.6	Testskriptanweisungen .....	38
2.5.7	Fehlermeldungen .....	38
2.6	Abgeleitete Größenmaße .....	39
2.6.1	Function-Points.....	39
2.6.2	Data-Points .....	40
2.6.3	Object-Points .....	41
2.6.4	Use-Case-Points .....	42
2.6.5	Testfall-Points .....	42
<b>3</b>	<b>Softwarekomplexität .....</b>	<b>45</b>
3.1	Komplexität in der Softwaremetrik .....	48
3.1.1	Software Komplexität nach dem IEEE-Standard.....	48
3.1.2	Softwarekomplexität aus der Sicht von Zuse .....	49
3.1.3	Softwarekomplexität nach Fenton .....	49
3.1.4	Komplexität als Krankheit der Softwareentwicklung .....	50

---

3.1.5	Komplexitätsmessung nach Ebert und Dumke .....	52
3.1.6	Die Alpha-Komplexitätsmetrik.....	54
3.2	Steigende Softwarekomplexität .....	56
3.2.1	Codekomplexität – Warum Java komplexer als COBOL ist.....	58
3.2.2	Entwurfskomplexität – Warum verschiedene Entwurfsansätze im Endeffekt gleich komplex sind.....	60
3.2.3	Anforderungskomplexität – Warum die zu lösenden Aufgaben immer komplexer werden .....	63
3.3	Allgemeingültige Maße für die Softwarekomplexität.....	64
3.3.1	Sprachkomplexität.....	64
3.3.2	Strukturkomplexität.....	65
3.3.3	Algorithmische Komplexität.....	65
<b>4</b>	<b>Die Messung der Softwarequalität .....</b>	<b>67</b>
4.1	Qualitätseigenschaften nach Boehm.....	68
4.1.1	Verständlichkeit nach Boehm.....	69
4.1.2	Vollständigkeit nach Boehm.....	70
4.1.3	Portabilität nach Boehm.....	70
4.1.4	Änderbarkeit nach Boehm .....	70
4.1.5	Testbarkeit nach Boehm .....	70
4.1.6	Benutzbarkeit nach Boehm .....	71
4.1.7	Zuverlässigkeit nach Boehm.....	71
4.1.8	Effizienz nach Boehm.....	71
4.2	Gilb und die Quantifizierung der Qualität .....	72
4.2.1	Funktionalitätsmessung nach Gilb .....	73
4.2.2	Performance-Messung nach Gilb.....	73
4.2.3	Zuverlässigungsmessung nach Gilb.....	73
4.2.4	Datensicherungsmessung nach Gilb .....	74
4.2.5	Effizienzmessung nach Gilb .....	74
4.2.6	Verfügbarkeitsmessung nach Gilb .....	74
4.2.7	Wartbarkeitsmessung nach Gilb .....	75
4.3	McCalls Qualitätsbaum .....	75
4.4	Eine deutsche Sicht auf Softwarequalität .....	77
4.4.1	Qualitätsbegriff.....	78
4.4.2	Qualitätsklassifizierung .....	78
4.4.3	Qualitätsmaße .....	79
4.4.4	Qualitätsgrößen.....	79
4.5	Automatisierte Softwarequalitätssicherung .....	79
4.5.1	Automatisierte Messung der Anforderungsqualität .....	80
4.5.2	Automatisierte Messung der Entwurfsqualität.....	81
4.5.3	Automatisierte Messung der Codequalität.....	82
4.5.4	Automatisierte Messung der Testqualität.....	84
4.6	Zielgerichtete Softwarequalitätssicherung .....	85
4.6.1	Qualitätszielbestimmung .....	85
4.6.2	Qualitätszielbefragung .....	85
4.6.3	Qualitätszielbemessung .....	86

4.7	IEEE- und ISO-Standards für Softwarequalität.....	87
4.7.1	Funktionalität nach ISO 9126.....	88
4.7.2	Zuverlässigkeit nach ISO 9126.....	88
4.7.3	Benutzbarkeit nach ISO 9126 .....	88
4.7.4	Effizienz nach ISO 9126.....	89
4.7.5	Wartbarkeit nach ISO 9126 .....	89
4.7.6	Portabilität nach ISO 9126.....	89
4.8	Folgen fehlender Qualitätssmessung.....	90
<b>5</b>	<b>Anforderungsmessung .....</b>	<b>91</b>
5.1	Tom Gilbs Anstoß der Anforderungsmessung .....	92
5.2	Weitere Ansätze zur Anforderungsmessung .....	94
5.2.1	Der Boehm-Ansatz .....	94
5.2.2	N-Fold Inspektion.....	96
5.2.3	Parnas & Weis Anforderungsprüfung.....	97
5.2.4	Abgleich der Anforderungen nach Fraser und Vaishnavi (Anforderungsprüfung).....	97
5.2.5	Verfolgung der Anforderungen nach Hayes .....	98
5.2.6	Bewertung der Anforderungen nach Glinz.....	99
5.2.7	ISO-Standard 25030 .....	100
5.2.8	Das V-Modell-XT als Referenzmodell für die Anforderungsmessung.....	101
5.3	Eine Metrik für Anforderungen von C. Ebert.....	102
5.3.1	Zahl aller Anforderungen in einem Projekt .....	103
5.3.2	Fertigstellungsgrad der Anforderungen .....	103
5.3.3	Änderungsrate der Anforderungen .....	103
5.3.4	Zahl der Änderungsursachen .....	103
5.3.5	Vollständigkeit des Anforderungsmodells.....	103
5.3.6	Anzahl der Anforderungsmängel.....	104
5.3.7	Anzahl der Mängelarten .....	104
5.3.8	Nutzwert der Anforderungen.....	104
5.4	Die Sophist-Anforderungsmetrik .....	105
5.4.1	Eindeutigkeit der Anforderungen .....	106
5.4.2	Ausschluss der Passivform bei den Anforderungen.....	106
5.4.3	Klassifizierbarkeit der Anforderungen .....	106
5.4.4	Identifizierbarkeit der Anforderungen .....	106
5.4.5	Lesbarkeit .....	107
5.4.6	Selektierbarkeit.....	107
5.5	Werkzeuge für die Anforderungsmessung .....	107
5.5.1	Anforderungsmessung in den früheren CASE-Werkzeugen.....	107
5.5.2	Anforderungsmessung im CASE-Tool SoftSpec.....	108
5.5.3	Anforderungsmessung in den gegenwärtigen Requirements Management Tools...109	109
5.5.4	Anforderungsmetrik aus dem Werkzeug TextAudit .....	110
5.5.5	Darstellung der Anforderungsmetrik .....	113
5.6	Gründe für die Anforderungsmessung .....	114

---

<b>6</b>	<b>Entwurfsmessung.....</b>	<b>115</b>
6.1	Erste Ansätze zu einer Entwurfsmetrik .....	116
6.1.1	Der MECCA-Ansatz von Tom Gilb .....	116
6.1.2	Der Structured-Design-Ansatz von Yourdon und Constantine .....	116
6.1.3	Der Datenflussansatz von Henry und Kafura.....	118
6.1.4	Der Systemgliederungsansatz von Belady und Evangelisti .....	119
6.2	Entwurfsmessung nach Card und Glass.....	120
6.2.1	Entwurfsqualitätsmaße.....	121
6.2.2	Entwurfskomplexitätsmaße.....	123
6.2.3	Erfahrung mit der ersten Entwurfsmetrik .....	126
6.3	Die SOFTCON Entwurfsmetrik .....	127
6.3.1	Formale Vollständigkeits- und Konsistenzprüfung.....	128
6.3.2	Technische Qualitätsmaße für den Systementwurf.....	129
6.4	Objektorientierte Entwurfsmetrik.....	132
6.4.1	Die OO-Metrik von Chidamer und Kemerer .....	133
6.4.2	MOOD-Entwurfsmetrik.....	137
6.5	Entwurfsmetrik in UMLAudit .....	140
6.5.1	Entwurfsquantitätsmetrik.....	141
6.5.2	Entwurfskomplexitätsmetrik.....	143
6.5.3	Entwurfsqualitätsmetrik.....	147
6.5.4	Entwurfsgrößenmetrik .....	153
6.6	Entwurfsmetrik für Webapplikationen .....	156
<b>7</b>	<b>Codemetrik.....</b>	<b>159</b>
7.1	Programmaufbau .....	159
7.2	Ansätze zur Messung von Codekomplexität.....	162
7.2.1	Halsteads Software Science .....	162
7.2.2	McCabe's Zyklomatische Komplexität.....	164
7.2.3	Chapins Q-Komplexität .....	165
7.2.4	Elshofs Referenzkomplexität.....	166
7.2.5	Prathers Verschachtelungskomplexität .....	167
7.2.6	Weitere Codekomplexitätsmaße .....	169
7.3	Ansätze zur Messung von Codequalität.....	169
7.3.1	Der Codequalitätsindex von Simon .....	169
7.3.2	Der Maintainability-Index von Oman .....	171
7.3.3	Zielorientierte Codequalitätsmessung .....	172
7.4	Codemetrik im SoftAudit-System .....	180
7.4.1	Codequantitätsmetrik .....	181
7.4.2	Codekomplexität.....	182
7.4.3	Codequalität.....	185
7.5	Werkzeuge für die Codemessung .....	190
7.5.1	Die ersten Codemessungswerkzeuge .....	190
7.5.2	Codemessungswerkzeuge der 90er Jahre .....	191
7.5.3	Heutige Codemessungswerkzeuge.....	193
7.6	Beispiel einer Codemessung.....	195

<b>8</b>	<b>Testmetrik .....</b>	<b>197</b>
8.1	Testmessung in der früheren Projektpraxis .....	198
8.1.1	Das ITS-Projekt bei Siemens .....	198
8.1.2	Das Wella-Migrationsprojekt .....	199
8.2	Testmetrik nach Hetzel.....	201
8.3	Testmetrik bei IBM Rochester .....	203
8.4	Maßzahlen für den Systemtest.....	206
8.4.1	Testzeit .....	206
8.4.2	Testkosten.....	207
8.4.3	Testfälle .....	207
8.4.4	Fehlermeldungen .....	207
8.4.5	Systemtestüberdeckung .....	208
8.4.6	Empfehlungen von Hutcheson.....	208
8.4.7	Test-Points.....	209
8.5	Testmetrik im GEOS-Projekt .....	210
8.5.1	Messung der Testfälle.....	211
8.5.2	Messung der Testüberdeckung .....	214
8.5.3	Messung der Fehlerfindung.....	214
8.5.4	Auswertung der Testmetrik .....	216
8.6	Testmetrik nach Sneed und Jungmayr.....	217
8.6.1	Testbarkeitsmetrik .....	217
8.6.2	Testplanungsmetrik .....	220
8.6.3	Testfortschrittsmetrik.....	223
8.6.4	Testqualitätsmetrik .....	224
<b>9</b>	<b>Produktivitätssumierung von Software.....</b>	<b>229</b>
9.1	Produktivitätssumierung – Ein umstrittenes Thema .....	232
9.2	Softwareproduktivität im Rückblick .....	233
9.2.1	Dokumentenmessung mit dem Fog-Index .....	233
9.2.2	Produktivitätssumierung bei der Standard Bank of South Africa .....	234
9.2.3	Die Entstehung der Function-Point-Methode .....	235
9.2.4	Das COCOMO-I-Modell von Boehm.....	237
9.2.5	Putnams Softwaregleichung .....	238
9.2.6	Die Data-Point-Methode.....	241
9.2.7	Die Object-Point-Methode.....	243
9.2.8	Die Use-Case-Point-Methode .....	247
9.3	Alternative Produktivitätsmaße .....	249
9.4	Produktivitätsberechnung anhand der Softwaregröße .....	251
9.5	Aufwandserfassung .....	252
9.6	Softwareproduktivitätsarten .....	253
9.6.1	Programmierproduktivität.....	253
9.6.2	Designproduktivität .....	254
9.6.3	Analyseproduktivität .....	254
9.6.4	Testproduktivität.....	255
9.6.5	Gesamtproduktivität .....	255

---

9.7	Produktivitätsstudien .....	256
9.7.1	Studien über Softwareproduktivität in den USA.....	256
9.7.2	Studien über Softwareproduktivität in Europa.....	258
9.7.3	Probleme beim Produktivitätsvergleich .....	261
9.8	Produktivitätsmessung nach Wertbeitrag .....	261
<b>10</b>	<b>Die Messung der Wartungsproduktivität.....</b>	<b>263</b>
10.1	Frühere Ansätze zur Messung der Wartbarkeit von Software .....	265
10.1.1	Stabilitätsmaße von Yau und Collofello .....	266
10.1.2	Maintenance-Umfrage bei der U.S. Air Force .....	267
10.1.3	Die Wartbarkeitsstudie von Vessey und Weber.....	268
10.1.4	Bewertung der Softwarewartbarkeit nach Berns.....	269
10.1.5	Die Wartungsuntersuchung von Gremillion .....	270
10.1.6	Wartungsmetrik bei Hewlett-Packard.....	271
10.1.7	Wartungsmessung nach Rombach .....	272
10.1.8	Messung der Wartbarkeit kommerzieller COBOL Systeme .....	273
10.1.9	Der Wartbarkeitsindex von Oman .....	275
10.2	Ansätze zur Messung der Wartbarkeit objektorientierter Software .....	277
10.2.1	Erste Untersuchung der Wartbarkeit objektorientierter Programme.....	278
10.2.2	Chidamer/Kemerers OO-Metrik für Wartbarkeit.....	279
10.2.3	MOOD-Metrik als Indikator der Wartbarkeit .....	280
10.2.4	Eine empirische Validation der OO-Metrik für die Schätzung des Wartungsaufwands.....	281
10.2.5	Der Einfluss einer zentralen Steuerung auf die Wartbarkeit eines OO-Systems....	281
10.2.6	Kalkulation vom Wartungsaufwand aufgrund der Programmkomplexität.....	282
10.2.7	Vergleich der Wartbarkeit objektorientierter und prozeduraler Software .....	283
10.2.8	Zur Änderung der Wartbarkeit im Laufe der Softwareevolution .....	285
10.3	Wartungsproduktivitätsmessung.....	287
10.3.1	Erste Ansätze zur Messung von Wartungsproduktivität .....	287
10.3.2	Messung von Programmwartbarkeit im ESPRIT-Projekt MetKit.....	290
10.3.3	Wartungsproduktivitätsmessung in der US-Marine .....	292
10.3.4	Messung der Wartungsproduktivität bei Martin-Marietta.....	294
10.3.5	Vergleich der Wartungsproduktivität repräsentativer Schweizer Anwender .....	295
<b>11</b>	<b>Softwaremessung in der Praxis .....</b>	<b>299</b>
11.1	Dauerhafte Messverfahren.....	301
11.1.1	Beteiligung der Betroffenen.....	301
11.1.2	Aufbauen auf vorhandener Metrik .....	302
11.1.3	Transparenz des Verfahrens.....	302
11.2	Beispiele dauerhafter Messverfahren.....	303
11.2.1	Die Initiative von Hewlett-Packard zur Softwaremessung.....	303
11.2.2	Prozess- und Produktmessung in der Siemens AG .....	307
11.2.3	Die eingebaute Softwaremessung im GEOS-Projekt.....	311
11.3	Übergreifende Software-Cockpits und Dashboards.....	314
11.3.1	Aufbau und Funktionalität des Software-Cockpits .....	316
11.3.2	Dashboard.....	317

## **Inhalt**

---

11.3.3	Scorecard.....	318
11.3.4	Interaktive Analysen und Berichte .....	319
11.4	Einmalige Messverfahren.....	319
11.4.1	Vereinbarung der Messziele .....	321
11.4.2	Auswahl der Metrik.....	321
11.4.3	Bereitstellung der Messwerkzeuge .....	322
11.4.4	Übernahme der Messobjekte .....	322
11.4.5	Durchführung der Messung .....	323
11.4.6	Auswertung der Messergebnisse .....	323
11.5	Beispiel einer einmaligen Messung.....	324
<b>Literatur .....</b>		<b>329</b>
<b>Register .....</b>		<b>351</b>