
Inhaltsverzeichnis

1	Einleitung	1
1.1	Lamento	1
1.2	Legacysystem	2
1.3	Assessment	5
1.4	Dualismen	6
2	Messbarkeit	9
2.1	Komplexitätsmetriken	12
2.2	Halstead-Metriken	17
2.3	Funktionspunkt-Metrik	19
2.4	Small-Worlds	20
2.5	Entropie	23
2.6	Volatilität	26
2.7	Maintainability Index	27
2.8	Metrikbasierte Verbesserungen	30
3	Lebenszyklus	33
3.1	Zustände	34
3.2	Versionierung	38
3.3	Operationen	39
4	Softwareevolution	41
4.1	Alterungsprozess	44
4.2	Gesetze der Softwareevolution	48
4.3	Kontinuierliche Veränderung	49
4.4	Wachsende Komplexität	50
4.5	Entropie	52
4.6	Selbstregulierung	54
4.7	Erhaltung der organisatorischen Stabilität	55
4.8	Erhaltung der Ähnlichkeit	56
4.9	Wachstum	57

4.10	Nachlassende Qualität	57
4.11	Volatilität	58
4.12	Konsequenzen aus den Evolutionsgesetzen	61
4.13	Bloating	63
4.14	Taxonomie der Änderung	64
4.15	Anforderungsevolution	80
4.16	Wertentwicklung	82
4.17	Komplexitätskosten	82
4.18	Datenqualität	84
4.19	Architekturevolution	85
4.20	Mitose	86
5	Migration	87
5.1	Enterprisemigration	91
5.2	Organisatorische Aspekte	92
5.3	Technische Migration	92
5.4	Softwareentwicklungsstrategien	96
5.5	Maintenanceende	104
5.6	Reengineering	105
5.7	Business Process Reengineering	107
5.8	Replacement	109
5.9	Software Reengineering	110
5.10	Reverse Engineering	114
5.11	Datenstrategien	120
5.12	Organisatorische Migrationsprobleme	133
6	Legacytransformation	137
6.1	Transformationsprozess	138
6.2	Refactoring	142
6.3	Zielplattformen	143
6.4	Projektmanagement	145
6.5	Transformationsbeispiel	146
7	Maintenance	153
7.1	Softwarequalität	156
7.2	Taxonomie	159
7.3	Kostenverteilung	164
7.4	Maintenanceservices	167
7.5	Maintenanceprozess	173
7.6	Maintenanceprozessverbesserung	177
7.7	Maintenance-Funktionspunkte	180
7.8	Impact-Analyse	181
7.9	Sourcecode	188
7.10	Vorhersagbarkeit	191
7.11	Menschliche Effekte	192

7.12	Stochastische Modelle	197
7.13	Defektraten	201
7.14	Services-Maintenance	202
8	Outsourcing	205
8.1	Vorgehensweisen	209
8.2	Risiken	213
8.3	Insourcing	215
9	Produktlinien	217
9.1	Einsatz	220
9.2	Kognitive Effekte	223
9.3	Assetmining	224
9.4	Architekturmining	229
9.5	Produktlinienwege	230
9.6	Featuremodell	237
9.7	Typische Probleme	238
9.8	Evolution von Produktlinien	239
10	COTS	241
10.1	Teilersatz	244
10.2	Ersatz	258
10.3	Softwareevolution und COTS	260
10.4	Defekte in COTS-Software	262
10.5	COTS-Softwareisolation	265
11	Entwicklungsprozesse	267
11.1	Komplexe Systeme	268
11.2	Rational Unified Process	272
11.3	Enterprise Unified Process	276
11.4	Agiles Manifest	281
11.5	Agile Maintenance	297
12	Architekturen und Sprachen	299
12.1	Legacyarchitekturen	299
12.2	Legacysprachen	318
12.3	Neuere Architekturen	323
12.4	Java 2 Enterprise Edition	330
12.5	.NET	342
12.6	Enterprise Application Integration	346
12.7	MQ-Series	353
12.8	Service Oriented Architecture	355
12.9	Webservices	357
12.10	Systemintegration	368

13 Patterns und Antipatterns	373
13.1 Softwaredarwinismus	374
13.2 Kleine Oberfläche	375
13.3 Service Layer	375
13.4 Gateway	376
13.5 Teile und Modernisiere!	377
13.6 Externalisierung	378
13.7 Legacysoftwareintegration	379
13.8 Façade	380
13.9 Adaptor	380
13.10 Schichten	381
13.11 Model View Controller	382
13.12 Distributed Object	383
13.13 Broker	384
13.14 Conway's Law	385
13.15 Silver Bullet	385
13.16 Batteries not included	386
14 Epilog	387
Literaturverzeichnis	391
Sachverzeichnis	411



<http://www.springer.com/978-3-540-25412-6>

Legacysoftware

Das lange Leben der Altsysteme

Masak, D.

2006, XII, 434 S., Hardcover

ISBN: 978-3-540-25412-6