

# Inhaltsverzeichnis

<b>Einleitung</b> .....	<b>1</b>
<b>1 Das Wesen von Software</b> .....	<b>5</b>
1.1 Prinzip der Speicherprogrammierbarkeit oder die schnelle Änderbarkeit aufgrund des Baumaterials .....	6
1.2 Komplexe Aufgabenstellungen oder Verlust der Anschaulichkeit.....	8
1.3 Fazit .....	9
<b>2 Dynamik in der Software-Entwicklung</b> .....	<b>11</b>
2.1 Leistungsdynamik .....	11
2.2 Technologiedynamik.....	12
2.3 Teamdynamik.....	12
2.4 Anforderungsdynamik .....	13
2.5 Fazit .....	15
<b>3 Software-Architektur</b> .....	<b>17</b>
3.1 Bedeutung von Architektur in der Software-Entwicklung .	17
3.2 Architektur als Brücke zwischen System und Computerprogramm .....	21
3.3 Hardware-Konfiguration.....	25
3.4 Fazit .....	27
<b>4 Beispiel einer Software-Architektonik</b> .....	<b>29</b>
4.1 Aufgabenstellung .....	29
4.2 Entwurf einer Software-Architektonik für PINT-Services .	31
4.2.1 Basisaufbau .....	32



4.2.2	Einschub: Architekturelemente in der Notation COSPEL..	35
4.2.3	Reaktionszeit und Durchsatz.....	37
4.2.4	Standardschnittstellen.....	46
4.2.5	Offene Schnittstellen .....	48
4.2.6	Verfügbarkeit.....	51
4.2.7	Versionsentwicklung .....	55
4.2.8	Prototyp der Software-Architektonik für PINT-Services...	58
4.2.9	Bewertung der Software-Architektonik des PINT-Gateways.....	60
4.3	Fazit .....	61
<b>5</b>	<b>Zeit- und Kostenschätzungen.....</b>	<b>65</b>
5.1	Grundsätzliches zur Zeit- und Kostenschätzung .....	65
5.2	Produktivität in der Software-Entwicklung .....	66
5.3	Das modifizierte Aron-Modell und das Cocomo-Modell ..	68
5.4	Vergleich mit Einzelangaben aus der Literatur .....	74
5.5	Function Points .....	75
5.6	Aktivitätsorientierte Aufwandsschätzung.....	75
5.7	Fazit .....	76
<b>6</b>	<b>Architekturfaktoren, Prozedur zur Entscheidungsfindung.....</b>	<b>79</b>
6.1	Checkliste .....	80
6.2	Bewertungsmatrix .....	84
6.3	Fazit .....	85
<b>7</b>	<b>Wechselwirkung zwischen Architekturelementen und Implementierung .....</b>	<b>87</b>
7.1	Strukturierungssichten für ein Software-System.....	87
7.2	Auswirkung der Struktur auf das Laufzeitverhalten.....	89
7.3	Durchstich zur Betriebsmittelabschätzung .....	91
7.4	Fazit .....	92
<b>8</b>	<b>Von den Komponenten zur Schicht .....</b>	<b>93</b>
8.1	Ordnung auf Komponenten.....	93
8.2	Konstruktion von Schichten.....	97
8.3	Schichtenmodell für Internet Supplementary Services .....	99
8.4	Architekturregeln.....	100
8.5	EWSD-Serviceschichtung.....	101
8.6	Fazit .....	103

<b>9</b>	<b>Aufteilung des Software-Systems auf ein verteiltes Computersystem.....</b>	<b>105</b>
9.1	Betriebsmittelzuordnung.....	105
9.2	Graphentheoretischer Lösungsansatz .....	107
9.3	Fazit .....	110
<b>10</b>	<b>Ausführen von Programmen .....</b>	<b>111</b>
10.1	Programme und ihre Realisierung durch Programmiersprachen .....	111
10.2	Prozesse als konkurrierende Ausführungseinheiten.....	113
10.3	Synchronität, Asynchronität und ihr Zusammenwirken ...	116
10.4	Prozeßkonzept in CHILL.....	119
10.5	Abhängigkeit zwischen Programmiersprache und Betriebssystem .....	122
10.6	Auswahlkriterien für eine Programmiersprache.....	123
10.7	Fazit .....	125
<b>11</b>	<b>Formale Grundlage der Architekturelemente.....</b>	<b>127</b>
11.1	Semantik der Signalmodelle .....	127
11.2	Semantik der Komponentenmodelle .....	132
11.2.1	Semantik des Protokollautomaten .....	136
11.2.2	Semantik des asynchronen Interfaces.....	138
11.2.3	Semantik des synchronen Interfaces.....	138
11.2.4	Semantik des Service .....	138
11.3	Einführung in COSPEL .....	139
11.3.1	Begründung und Prinzipien .....	139
11.3.2	Service .....	140
11.3.3	Komponente .....	142
11.3.4	Applikation.....	143
11.4	Sprachbeschreibung COSPEL .....	145
11.4.1	Hinweise zur Metasprache .....	145
11.4.2	COSPEL-Spezifikation .....	146
11.4.3	Applikation.....	146
11.4.4	Komponente .....	149
11.4.5	Service .....	152
11.4.6	Synchrone und asynchrone Interfaces .....	155
11.4.7	Protokollautomat .....	158
11.4.8	Channel.....	160
11.4.9	Signaturen (Formale Parameter).....	161
11.5	Beispiel mit synchronen und asynchronen Schnittstellen.	161
11.6	Click-to-Dial-Beispiel in COSPEL.....	164
11.7	Fazit .....	173

<b>12</b>	<b>Ausfallsicherheit: Redundanzarchitektur bei Echtzeitsystemen .....</b>	<b>175</b>
12.1	Software-Zuverlässigkeit .....	175
12.2	Kombinatorische Regeln .....	177
12.3	Software-Fehlerbehandlung .....	179
12.4	Fazit .....	181
<b>13</b>	<b>Qualitätssicherung.....</b>	<b>183</b>
13.1	Qualität von Software.....	183
13.1.1	Entwicklungsprozeß und Fehlervermeidung .....	186
13.1.2	Fehlerquellen und Fehlerfindung .....	189
13.2	Entwicklungsmethode .....	193
13.3	Inspektionsverfahren .....	197
13.4	Testprozeß.....	201
13.4.1	Testmethoden.....	201
13.4.2	Durchführung der Tests.....	207
13.4.3	Beispiel: Funktionstest der Komponente SIP/PINT-Stack .....	208
13.5	Fazit .....	211
<b>14</b>	<b>Methoden- und Sprachbewertung: UML, SDL, Java, C++, CORBA .....</b>	<b>213</b>
14.1	UML .....	213
14.2	SDL .....	214
14.3	Java .....	215
14.4	C++ .....	216
14.5	CORBA .....	217
14.6	Fazit .....	217
<b>15</b>	<b>Projektmanagement .....</b>	<b>219</b>
15.1	Projekte .....	219
15.2	Entwicklungsprozesse .....	222
15.2.1	Der sequentielle Prozeß mit Rückkopplung .....	222
15.2.2	Extreme Programming .....	224
15.2.3	Der sequentielle Prozeß mit Frühwarnsystem .....	225
15.3	Verwaltung von Entwicklungselementen .....	231
15.4	Fazit .....	237
<b>16</b>	<b>Erfolgreiche Projektkooperation: Soziale Bedingungen .....</b>	<b>239</b>
16.1	Sozialwissenschaftliche Aspekte der Kooperation.....	239
16.2	Erfahrungen aus praktischen Projekten .....	241

16.2.1	Architekturgruppe .....	242
16.2.2	Einheitliche Begriffswelt .....	242
16.2.3	Kommunikation in Entwicklungsgruppen.....	243
16.2.4	Kommunikation zwischen Gruppen .....	244
16.3	Fazit .....	244
<b>17</b>	<b>Resümee.....</b>	<b>245</b>
<b>Anhang.....</b>	<b>.....</b>	<b>251</b>
	Pseudo-Code der Komponentenmodelle.....	251
	Verzeichnis für Bilder und Tabellen .....	258
<b>Glossar .....</b>	<b>.....</b>	<b>261</b>
<b>Literaturverzeichnis .....</b>	<b>.....</b>	<b>265</b>
<b>Index.....</b>	<b>.....</b>	<b>269</b>

