

## Inhaltsverzeichnis:

Abbildungsverzeichnis .....	11
Abkürzungsverzeichnis .....	15
<b>1 Einleitung .....</b>	<b>17</b>
<b>2 Die Problematik der Softwarewartung .....</b>	<b>21</b>
2.1 Arten der Softwarewartung .....	21
2.2 Bedeutung der Wartung im Softwarelebenszyklus .....	23
2.3 Ursachen der Wartungsproblematik .....	26
2.4 Kurzbeschreibung und Wartungsproblematik des Lehrstuhlinformationssystems .....	28
<b>3 Methoden und Werkzeuge der Softwaresanierung .....</b>	<b>33</b>
3.1 Teilgebiete der Softwaresanierung .....	33
3.2 Reverse Engineering .....	37
3.2.1 Ziele des Reverse Engineering .....	37
3.2.2 Disassemblierung und Discompilierung .....	37
3.2.3 Redokumentation .....	39
3.2.3.1 Statische Analyse .....	39
3.2.3.2 Dynamische Analyse .....	42
3.2.3.3 Komplexitätsanalyse .....	44
3.2.4 Redesign .....	48
3.2.4.1 Redesign der Funktionen .....	49
3.2.4.2 Redesign des Datenmodells .....	59
3.3 Restrukturierung .....	62
3.3.1 Ziele der Restrukturierung .....	62
3.3.2 Reformatierung .....	63
3.3.3 Restrukturierung im engeren Sinn .....	64
3.3.4 Modularisierung .....	70
3.3.5 Datenstandardisierung .....	72
3.4 Reengineering .....	73
3.4.1 Ziele des Reengineering .....	73
3.4.2 Integration des Sanierungswerkzeugs mit einem CASE-Tool .....	75
3.4.3 Erleichterung der Wiederverwendung durch Reengineering .....	75
3.4.4 Migrationsunterstützung durch Reengineering-Werkzeuge .....	76

4	Erhöhung der Wartungswirtschaftlichkeit durch den Einsatz von Sanierungsmaßnahmen .....	81
4.1	Bestimmung einer unternehmensspezifischen Gesamtstrategie zur Senkung der Wartungskosten .....	81
4.2	Auswahlverfahren für Sanierungsobjekte .....	84
4.2.1	Software-Inventur .....	84
4.2.2	Daumenregeln für die Auswahl von Sanierungsobjekten .....	85
4.2.3	ABC-Analyse .....	86
4.2.4	Portfolio-Analyse .....	88
4.3	Kosten und Nutzen der Softwaresanierung .....	91
4.3.1	Voraussetzungen zur Bestimmung der Sanierungswirtschaftlichkeit .....	91
4.3.2	Nutzwertanalyse .....	94
4.3.3	Das COCOMO-Modell von BOEHM .....	96
4.3.4	Break-Even-Analyse .....	97
5	Möglichkeiten für die Durchführung eines objektorientierten Reengineering ....	99
5.1	Grundlagen der Objektorientierung .....	99
5.2	Objektorientierte Datenbanken .....	102
5.3	Objektorientiertes Reengineering .....	104
5.3.1	Motivation für ein objektorientiertes Reengineering .....	104
5.3.2	Bestehende Ansätze zum objektorientierten Reengineering .....	108
5.3.3	Umfang des objektorientierten Reengineering .....	110
5.4	Entwicklung einer Methode für das objektorientierte Reengineering .....	112
5.4.1	Objektorientierte Analyse- und Designmethoden .....	112
5.4.2	Kerngedanken der entwickelten Methode .....	115
5.4.3	Das Strukturierte Entity-Relationship-Modell (SERM) als Basis-Datenmodell .....	116
5.4.4	Das Funktionenmodell der Object Modeling Technique (OMT) als Basis-Funktionenmodell .....	125
5.4.5	Integration der Modelle .....	128
5.5	Wartungsunterstützung für objektorientierte Programme .....	139
6	Objektorientiertes Reengineering des Lehrstuhlinformationssystems .....	143
6.1	Werkzeuggestützte Redokumentation und Reformatierung des Lehrstuhlinformationssystems .....	143
6.2	Reverse Engineering des Datenmodells .....	149
6.3	Reverse Engineering des Funktionenmodells .....	153

6.4	Integration von Daten- und Funktionenmodell .....	160
6.5	Beispielhafte Umsetzung des Modells mit der objektorientierten Datenbank POET .....	170
6.6	Konvertierung der Inhalte von dBASE-Datenbankdateien in die POET-Objektbank .....	173
7	Schlußbetrachtung .....	177
	Literaturverzeichnis .....	179

## Abbildungsverzeichnis:

Abb. 1-1: Neuentwicklungen bilden nur die Spitze des Eisbergs der Gründe für die Softwarekrise .....	18
Abb. 2-1: Prozentuale Aufteilung der Wartungsaktivitäten nach Wartungsarten .....	23
Abb. 2-2: Kostenverteilung über den Softwarelebenszyklus .....	24
Abb. 2-3: Ressourcenverbrauch für existierende Systeme .....	24
Abb. 2-4: Zeitaufwand für die einzelnen Aktivitäten des Wartungsprozesses .....	25
Abb. 3-1: Aktivitäten im Rahmen von Forward und Reverse Engineering .....	35
Abb. 3-2: Aktivitäten im Rahmen von Restrukturierung und Reengineering .....	35
Abb. 3-3: Kontrollflußknotenmetrik .....	47
Abb. 3-4: Datenflußdiagramm .....	50
Abb. 3-5: Structure Chart .....	51
Abb. 3-6: Der Zusammenhang zwischen Structure Chart und Struktogramm .....	52
Abb. 3-7: dBASE-Quellcode-Ausschnitt des Lehrstuhlinformationssystems .....	54
Abb. 3-8: Programmablaufplan .....	55
Abb. 3-9: Baumdiagramm .....	56
Abb. 3-10: Struktogramm .....	57
Abb. 3-11: Darstellung von Datenbeziehungen durch Redesign .....	60
Abb. 3-12: Entity-Relationship-Diagramm .....	61
Abb. 3-13: Unstrukturierter Pseudocodeausschnitt .....	65
Abb. 3-14: Restrukturierung durch Codeverdopplung .....	65
Abb. 3-15: Restrukturierung durch Codeauslagerung .....	65
Abb. 3-16: Restrukturierung durch den Status-Variablen-Ansatz .....	66
Abb. 3-17: Codeausschnitt vor der Restrukturierung durch Boolesche Schalter .....	67
Abb. 3-18: Restrukturierung durch Boolesche Schalter .....	67
Abb. 3-19: Codeausschnitt vor der Restrukturierung durch Prädikatverdopplung .....	68
Abb. 3-20: Restrukturierung durch Prädikatverdopplung .....	68
Abb. 3-21: Programmiersprachenwechsel durch Reengineering .....	79
Abb. 4-1: Portfolio-Analyse zur Bestimmung einer unternehmens- spezifischen Gesamtstrategie .....	83
Abb. 4-2: ABC-Analyse .....	87

Abb. 4-3: Portfolio-Analyse zur Bestimmung von Kandidaten für eine Softwaresanierung .....	90
Abb. 4-4: Kostenersparnis durch eine Sanierung .....	98
Abb. 5-1: Verteilung von Wartungsaufträgen auf Systembestandteile .....	105
Abb. 5-2: Vorgehensweise beim selektiven integrationsorientierten Reengineering .....	112
Abb. 5-3: Stammbaum objektorientierter Analyse- und Designmethoden .....	114
Abb. 5-4: Die Überführung von Ergebnissen der Strukturierten Analyse in objektorientiertes Design nach ALABISO .....	115
Abb. 5-5: Präzisierung der Komplexität der Beziehung $b(A, B)$ durch die (min, max)-Notation .....	118
Abb. 5-6: Beispiele zur (min, max)-Notation .....	118
Abb. 5-7: Auflösung der Zwitterrolle von Entity- und Relationshipstypen .....	119
Abb. 5-8: Beispiele für schwache Entitytypen .....	120
Abb. 5-9: Zyklus wechselseitiger Existenzabhängigkeiten im ERM .....	120
Abb. 5-10: Datenobjekttypen im SERM .....	122
Abb. 5-11: Kantensymbole zur Darstellung von Beziehungen zwischen Datenobjekttypen im SERM .....	123
Abb. 5-12: Überführung von Teilstrukturen vom ERM (links) ins SERM (rechts) .....	124
Abb. 5-13: Das Hierarchiemodell der Strukturierten Analyse .....	127
Abb. 5-14: Zuordnung der Begriffe von Daten- und Objektmodell .....	129
Abb. 5-15: Die interacts_with-Beziehung im SOM .....	130
Abb. 5-16: Die is_part_of-Beziehung im SOM .....	131
Abb. 5-17: Die is_a-Beziehung im SOM .....	132
Abb. 5-18: Überführung einer Generalisierungshierarchie in eigenständige Tabellen für Super- und Subklassen .....	133
Abb. 5-19: Überführung einer Generalisierungshierarchie in eigenständige Tabellen für alle Subklassen .....	133
Abb. 5-20: Überführung einer Generalisierungshierarchie in eine einzelne Tabelle für die Superklasse .....	134
Abb. 5-21: Überführung einer is_part_of-Beziehung in eigenständige Tabellen für Ganz- und Teilklassen .....	134
Abb. 5-22: Überführung einer is_part_of-Beziehung in eine einzelne Tabelle .....	135
Abb. 5-23: Konsistenzbedingungen zwischen SOM- und SERM- Beziehungen .....	135

Abb. 5-24: Notation von Klassen&Objekten und Klassen nach COAD/YOURDON .....	136
Abb. 5-25: Notation der is_part_of-Beziehung nach COAD/YOURDON .....	137
Abb. 5-26: Notation für a) Subklassen- und b) Generalisierungshierarchien im engeren Sinne in der Notation von COAD/YOURDON .....	137
Abb. 5-27: Notation einer Instanzverbindung nach COAD/YOURDON .....	138
Abb. 5-28: Gegenüberstellung von Instanzverbindungen a) im SERM und b) in der Notation von COAD/YOURDON .....	138
Abb. 5-29: Notation einer interacts_with-Beziehung nach COAD/YOURDON .....	139
Abb. 6-1: Beschreibung der Struktur der Datenbankdatei MIK.DBF .....	144
Abb. 6-2: Aufrufliste des Programms MIKRO.PRG .....	145
Abb. 6-3: Ausschnitt der Aufrufhierarchie des Lehrstuhl- informationssystems .....	146
Abb. 6-4: Cross-Reference Liste der globalen Variablen ANZAHL .....	147
Abb. 6-5: Von SNAP! reformatierter Quellcode .....	148
Abb. 6-6: ERM des Lehrstuhlinformationssystems nach der (1,M,N)-Notation .....	151
Abb. 6-7: ERM des Lehrstuhlinformationssystems nach der (min, max)-Notation .....	152
Abb. 6-8: SERM des Lehrstuhlinformationssystems ohne Aggregation und Generalisierung .....	154
Abb. 6-9: Structure Chart eines fiktiven Programms .....	156
Abb. 6-10: Hierarchie von Datenflußdiagrammen, hergeleitet aus dem Structure Chart der Abbildung 6-9 .....	157
Abb. 6-11: Kontextdiagramm für das Lehrstuhlinformationssystem (Ebene 0) .....	158
Abb. 6-12: DFD für das Lehrstuhlinformationssystem (Ebene 1) .....	159
Abb. 6-13: DFD für den Prozeß Seminarverwaltung (Prozeß 6, Ebene 2) .....	160
Abb. 6-14: Datenflußdiagramm des Prozesses Computerpraktikum_Schein_Erstellung .....	161
Abb. 6-15: Attribute der Veranstaltungsdateien .....	162
Abb. 6-16: Darstellung der is_part_of-Beziehung zwischen Seminar und Seminartermin in ERM und SERM .....	163
Abb. 6-17: Klassenhierarchie der Veranstaltungen .....	164
Abb. 6-18: Attribute der Veranstaltungsteilnahmedateien .....	165
Abb. 6-19: Vererbungshierarchie der Attribute der Veranstaltungsteil- nahmedateien .....	166

Abb. 6-20: Die für die Vererbung relevanten Methoden der Veranstaltungsteilnahmen .....	167
Abb. 6-21: Vererbungshierarchie der Methoden der Veranstaltungsteilnahmedateien .....	167
Abb. 6-22: SOM des Lehrstuhlinformationssystems inklusive Aggregation und Generalisierung .....	168
Abb. 6-23: Darstellung des Lehrstuhlinformationssystems in der Notation von COAD/YOURDON .....	169
Abb. 6-24: Darstellung der Datenverwaltungskomponente am Beispiel der persistenten Klasse Student .....	172
Abb. 6-25: Vererbungshierarchie der Veranstaltungsteilnahmen nach der Überführung der multiplen Vererbung in einfache Vererbung .....	172
Abb. 6-26: Vorspann einer dBASE-Bestandsdatei .....	173
Abb. 6-27: Inhalt des Arrays der Feldbeschreibungen einer dBASE- Bestandsdatei .....	174
Abb. 6-28: Aufbau und Inhalt der Datei SEMINART.DBF und ihre Überführung in eine Klassenstruktur .....	175