
Inhaltsübersicht

1	Einleitung zum Thema Softwaremigration	1
2	Grundlagen und -begriffe der Softwaremigration	25
3	Migrationsprozesse aus Forschung und Praxis	49
4	Der Reference Migration Process (ReMiP)	85
5	Methoden und Techniken der Softwaremigration	133
6	Softwaremigrationswerkzeuge	171
7	Migrationsfallstudien	205
8	Die Zukunft der Migrationstechnologie	261
	Literatur	285
	Index	301

Inhaltsverzeichnis

1	Einleitung zum Thema Softwaremigration	1
1.1	Die Motivation für Softwaremigration	1
1.2	Zum Zustand der IT in der betrieblichen Praxis	4
1.3	Alternativen zur Bewältigung von Altlasten	7
1.3.1	Neuentwicklung	8
1.3.2	Ablösung durch Standardanwendungssoftware	9
1.3.3	Migration der Altsysteme	9
1.4	Migrationsstrategien	10
1.4.1	Reimplementierungsstrategie	11
1.4.2	Konversionsstrategie	11
1.4.3	Kapselungsstrategie (Wrapping)	12
1.4.4	Auswahl einer geeigneten Strategie	13
1.5	Umstellungs- und Übergabestrategien	13
1.5.1	Punktumstellung und Punktübergabe (Big Bang)	14
1.5.2	Langfristumstellung	15
1.5.3	Inkrementelle Paketumstellung	16
1.5.4	Inkrementelle Paketumstellung mit Komponenten- ersatz	17
1.5.5	Inkrementelle Paketumstellung mit Paralleleinsatz	18
1.6	Zur Wirtschaftlichkeit einer Softwaremigration	19
1.6.1	Betriebswirtschaftliche Sicht	20
1.6.2	Projektmanagement-Sicht	21
1.6.3	Technische Sicht	21
1.7	Zum Aufbau des Buches	22

2	Grundlagen und -begriffe der Softwaremigration	25
2.1	Abgrenzung des Themas Migration	25
2.1.1	Neuentwicklung, Weiterentwicklung und Evolution	25
2.1.2	Integration	26
2.1.3	Wartung und Maintenance	27
2.1.4	Reengineering	28
2.1.5	Reverse Engineering	29
2.1.6	Migration	29
2.2	Migrationsprojekte	32
2.2.1	Falsche Gründe für Migrationsprojekte	33
2.2.2	Wahre Gründe für Migrationsprojekte	34
2.3	Migrationsansätze	34
2.3.1	Reimplementierung	35
2.3.2	Konvertierung	36
2.3.3	Kapselung	36
2.3.4	Ein Beispiel zur Unterscheidung der Migrationsansätze	37
2.4	Migrationsarten	38
2.4.1	Datenmigration	40
2.4.2	Programm-Migration	41
2.4.3	Benutzungsschnittstellen-Migration	42
2.4.4	Systemschnittstellen-Migration	43
2.4.5	Systemmigration	44
2.5	Anwenderoptionen	45
2.5.1	Interne Migration mit eigenen Ressourcen	45
2.5.2	Interne Migration mit fremden Ressourcen	46
2.5.3	Externe Migration mit internem Test	46
2.5.4	Externe Migration einschließlich Test	47
2.5.5	Kriterien für die Entscheidung	47
3	Migrationsprozesse aus Forschung und Praxis	49
3.1	Chicken-Little-Migrationsansatz	50
3.1.1	Zielsetzung und Idee des Chicken-Little-Ansatzes	51
3.1.2	Chicken-Little-Migrationsobjekte	52
3.1.3	Chicken-Little-Vorgehensmodell	53
3.1.4	Bewertung des Chicken-Little-Ansatzes	56

3.2	Butterfly-Migrationsansatz	56
3.2.1	Zielsetzung und Idee des Butterfly-Ansatzes	57
3.2.2	Butterfly-Vorgehensmodell	58
3.2.3	Bewertung des Butterfly-Ansatzes	59
3.3	Der Renaissance-Migrationsprozess	60
3.3.1	Phasen im Renaissance-Migrationsansatz	61
3.3.2	Das Renaissance-Integrationsmodell	62
3.3.3	Bewertung des Renaissance-Ansatzes	63
3.4	Das COREM-Migrationsverfahren	63
3.4.1	Der Top-down-Ansatz	64
3.4.2	Der Bottom-up-Ansatz	64
3.4.3	Verschmelzung der beiden Ansätze	65
3.4.4	Bewertung des COREM-Verfahrens	65
3.5	Die Migrationsfabrik	66
3.5.1	Zielsetzung und Idee der Migrationsfabrik	66
3.5.2	Vorgehensmodell der Migrationsfabrik	67
3.5.3	Bewertung der Migrationsfabrik	69
3.6	SMART-Softwaremigration	70
3.6.1	Zielsetzung und Idee von SMART	70
3.6.2	SMART-Vorgehensmodell	71
3.6.3	Bewertung des SMART-Ansatzes	75
3.7	Objektorientierte Softwaremigration	75
3.7.1	Zielsetzung des objektorientierten Ansatzes	75
3.7.2	Objektorientiertes Vorgehensmodell	76
3.7.3	Bewertung des objektorientierten Ansatzes	79
3.8	Testgetriebene Migrationsprozesse	79
3.9	Werkzeuggetriebene Migrationsprozesse	81
4	Der Reference Migration Process (ReMiP)	85
4.1	Prozessmodelle als Sammlung von Erfahrungswerten	86
4.1.1	Erfahrungswerte aus der Softwaremigration	87
4.1.2	Erfahrungswerte aus der Softwareentwicklung	88
4.1.3	Synthese der Erfahrungswerte aus Entwicklung und Migration	89
4.1.4	ReMiP als Prozess-Framework	91

4.2	Struktur des ReMiP-Prozessmodells	91
4.2.1	ReMiP-Phasen	92
4.2.2	ReMiP-Meilensteine	95
4.2.3	Kern- und Basisbereiche des ReMiP	96
4.2.4	Prozessmodell des ReMiP	97
4.3	Kernbereich Anforderungsanalyse	98
4.3.1	Analyse der Anforderungen	99
4.3.2	Dokumentation der Anforderungen	101
4.4	Kernbereich Legacy-Analyse und -Aufbereitung	101
4.4.1	Analyse des Legacy-Systems	102
4.4.2	Globale Bewertung des Legacy-Systems	103
4.4.3	Reverse Engineering des Legacy-Systems	105
4.4.4	Vorbereitung des Legacy-Systems	106
4.5	Kernbereich Zielsystementwurf	108
4.5.1	Definition der Zielsystemkandidaten	109
4.5.2	Verfeinerung der Zielsystemarchitektur	110
4.5.3	Definition der Übergangsarchitektur	110
4.5.4	Entwurf der Programme	111
4.5.5	Entwurf der Benutzungsschnittstellen	111
4.5.6	Entwurf der Datenbanken	112
4.5.7	Spezifikation der Transformationen	112
4.6	Kernbereich Strategiewahl	113
4.6.1	Erarbeitung der Migrationsstrategien für Zielsystemkandidaten	114
4.6.2	Bewertung der Migrationsstrategien für Zielsystemkandidaten	116
4.6.3	Auswahl der Migrationsstrategie	116
4.6.4	Verwaltung der Migrationsstrategie	117
4.7	Kernbereich Transformation	117
4.7.1	Isolation des Migrationspakets	118
4.7.2	Transformation des Migrationspakets	119
4.7.3	Durchführung der Deltamigration	120
4.7.4	Implementierung neuer Komponenten	120
4.8	Kernbereich Test	121
4.8.1	Definition der globalen Teststrategie	122
4.8.2	Testspezifikation für das Migrationspaket	123
4.8.3	Testdurchführung für das Migrationspaket	123
4.8.4	Auswertung der Testergebnisse	124

4.9	Kernbereich Übergabe	125
4.9.1	Einrichtung der Zielumgebung	126
4.9.2	Planung der Übergabe	127
4.9.3	Entwicklung des Supportmaterials	127
4.9.4	Durchführung der Übergabe	128
4.9.5	Ablösung des Legacy-Systems	128
4.10	Basisbereich Konfigurations- und Änderungsmanagement	129
4.11	Basisbereich Projektmanagement	129
4.12	Basisbereich Mitarbeiterqualifizierung	130
4.13	Basisbereich Migrationsumgebung	131
5	Methoden und Techniken der Softwaremigration	133
5.1	Softwaremesstechnik	134
5.2	Systembewertung	136
5.3	Migrationsaufwandsschätzung	139
5.3.1	Schätzung nach COCOMO II	140
5.3.2	Schätzung mit Object Points	143
5.3.3	Schätzung mit Data Points	144
5.4	Migrationsplanung	145
5.5	Migrationssteuerung	148
5.6	Entwurfsmethoden	150
5.6.1	Entwurf der Istarchitektur	150
5.6.2	Entwurf der Sollarchitektur	151
5.7	Code-Transformationstechniken	153
5.8	Code-Reimplementierungstechnik	155
5.9	Code-Kapselungstechniken	157
5.9.1	Prozesskapselung	158
5.9.2	Transaktionskapselung	159
5.9.3	Programmkapselung	159
5.9.4	Modulkapselung	159
5.9.5	Prozedurkapselung	160
5.10	Datenkonvertierungstechnik	160
5.11	Datenkapselungstechnik	163
5.12	Regressionstesttechniken	164
5.12.1	Voller Regressionstest	165
5.12.2	Selektiver Regressionstest	166

5.13	Systemnachdokumentation	167
5.14	Systemverwaltung	168
6	Softwaremigrationswerkzeuge	171
6.1	Codeanalysewerkzeuge	173
6.1.1	Codeprüfung	173
6.1.2	Codemessung	174
6.1.3	Verfügbare Analysewerkzeuge	176
6.2	Schätzwerkzeuge	178
6.2.1	COCOMO-II	179
6.2.2	SoftCalc	180
6.3	Projektmanagementwerkzeuge	180
6.4	Entwurfswerkzeuge	181
6.5	Programmtransformationswerkzeuge	183
6.5.1	Die prozeduralen Werkzeuge	183
6.5.2	Die objektorientierten Werkzeuge	186
6.6	Programmkapselungswerkzeuge	189
6.6.1	Screen Scraping Tools	189
6.6.2	Transaction Wrapping Tools	191
6.6.3	Function Wrapping Tools	192
6.7	Datenkonvertierungswerkzeuge	193
6.8	Datenkapselungswerkzeuge	196
6.8.1	Kapselung der Legacy-Datenbanksysteme	196
6.8.2	Kapselung relationaler Datenbanken	197
6.9	Regressionstestwerkzeuge	198
6.9.1	Kommerzielle Testwerkzeuge	198
6.9.2	Regressionstest einzelner Module	199
6.9.3	Regressionstest der migrierten Systeme	200
6.10	Nachdokumentationswerkzeuge	202
6.11	Systemverwaltungswerkzeuge	204
7	Migrationsfallstudien	205
7.1	RCOST: Migration von COBOL-Client/Server-Systemen in eine Webumgebung	206
7.1.1	Projekthintergrund	207
7.1.2	Migrationsvorbereitung	207

7.1.3	Migrationsdurchführung	209
7.1.4	Lehren aus dem RCOST-Projekt	211
7.2	Migration von Mainframe-COBOL-Programmen in eine UNIX-J2EE-Umgebung	211
7.2.1	Projekthintergrund	212
7.2.2	Schichtenarchitektur	213
7.2.3	Erfahrungen mit der Kapselung alter Programme und deren Daten	216
7.3	Eine inkrementelle Systemablösung	217
7.4	Eine schrittweise COBOL-Migration in drei Akten	219
7.4.1	Der erste Akt – Beseitigung der Altlasten	220
7.4.2	Der zweite Akt – Zurück zu Standard-COBOL	222
7.4.3	Der dritte Akt – Der Übergang zum IBM-Mainframe	224
7.5	Migration einer Mainframe-Anwendung auf UNIX	226
7.5.1	Machbarkeitsstudie	227
7.5.2	Grundprämissen für das Projekt	227
7.5.3	Gegenstände der Migration	228
7.5.4	Regressionstest des migrierten Billing-Systems	229
7.5.5	Erfahrungen aus der Migration des Billing-Systems	229
7.6	Massenmigration von einem Bull-Mainframe auf verteilte UNIX-Rechner	230
7.6.1	Projektorganisation	231
7.6.2	Einrichtung von Gateways zwischen den Rechnern	231
7.6.3	Aufbau einer Migrationsfabrik	232
7.6.4	Projektresümee	234
7.7	Das ARNO-Projekt : Beispiel einer Migration von einem proprietären Mainframe-Rechner in eine offene UNIX-Umgebung	235
7.7.1	Die Migration beginnt mit einer Studie	236
7.7.2	Die Projektorganisation	237
7.7.3	Die Vorbereitung der Migration	237
7.7.4	Der Wechsel der Programmiersprache	238
7.7.5	Der Konvertierungsprozess	241
7.7.6	Der Testprozess	244
7.7.7	»Migration« der Mitarbeiter	246
7.7.8	Übergabe an die Linienorganisation	247
7.7.9	Fazit einer erfolgreichen Migration	247

7.8	Automatisierte Migration von COBOL in Java	248
7.8.1	Die Ausgangssituation	250
7.8.2	Alternative Lösungen	251
7.8.3	Die Spezifikation der Reimplementierung	252
7.8.4	Der Code-Konvertierungsprozess	254
7.8.5	Das Konvertierungsergebnis	256
7.8.6	Das Ergebnis als Zwischenstufe	258
7.8.7	Eine testgetriebene Migration	259
8	Die Zukunft der Migrationstechnologie	261
8.1	Die aktuelle Migrationswelle – serviceorientierte Architekturen (SOA)	263
8.2	Alternative Wege zur SOA	264
8.2.1	Top-down-Migrationsansatz	264
8.2.2	Bottom-up-Migrationsansatz	265
8.2.3	Gemischter Migrationsansatz	266
8.2.4	Voraussetzung für eine Migration zur SOA	267
8.3	Die Bereitstellung von Web Services	268
8.3.1	Web Services einkaufen	268
8.3.2	Web Services mieten (Software on Demand)	269
8.3.3	Web Services ausleihen	270
8.3.4	Web Services entwickeln	271
8.3.5	Web Services aus vorhandenen Systemen wiedergewinnen	272
8.4	Ansätze zur Wiedergewinnung von Web Services	273
8.4.1	Entdecken potenzieller Web Services	274
8.4.2	Bewertung potenzieller Web Services	275
8.4.3	Extrahierung des Codes für den Web Service	277
8.4.4	Anpassung des Web-Service-Codes	278
8.5	Test der Web Services	279
8.6	Migration als Outsourcing-Geschäft	280
8.7	Widerstände gegen Migration	281
8.8	Das Rad dreht sich weiter	283
	Literatur	285
	Index	301