

Inhalt

Geleitwort	13
Vorwort	15
Einleitung	17

1 Architektur moderner ABAP-Anwendungen 21

1.1 Eigenschaften und Architektur von SAP S/4HANA	21
1.1.1 Entwicklungsprinzipien von SAP S/4HANA	22
1.1.2 Deployment-Varianten von SAP S/4HANA	26
1.2 Entwicklung von Unternehmensanwendungen mit SAP S/4HANA	30
1.2.1 Anforderungen an moderne ABAP-Anwendungen	31
1.2.2 Standardisierung der Anwendungsarchitektur	33
1.2.3 Technologie für die Entwicklung von Benutzeroberflächen	37
1.3 Entwicklung auf Basis von SAP NetWeaver 7.50	42
1.3.1 Simplifizierung und Redesign	42
1.3.2 Wechsel der UI-Technologie	43

2 Das neue ABAP-Programmiermodell für SAP Fiori 47

2.1 ABAP-Entwicklungsumgebung und -Syntax	48
2.1.1 ABAP in Eclipse	48
2.1.2 Moderne ABAP-Syntax	49
2.2 Einsatzszenarien des ABAP-Programmiermodells	52
2.2.1 Komponenten des ABAP-Programmiermodells	52
2.2.2 Das ABAP-Programmiermodell in der Praxis	56
2.3 Core Data Services	60
2.3.1 ABAP CDS	60
2.3.2 Typen von ABAP CDS Views	66
2.3.3 CDS-Annotationen	72
2.3.4 Eingangsparameter	75

2.3.5	Assoziationen	77
2.3.6	Performanceprobleme	79
2.4	Entwicklung von OData-Services mit SAP Gateway	83
2.4.1	Optionen zur Entwicklung des OData-Service	84
2.4.2	Registrierung des OData-Service	87
2.4.3	Testen des OData-Service mit dem SAP Gateway Client	88
2.5	Code Pushdown auf die SAP-HANA-Datenbank	94
2.5.1	Goldene Regeln von SAP	95
2.5.2	Strategien für die Realisierung des Code Pushdowns	96
2.6	Simplifizierung und das Principle of One von SAP	101
2.6.1	Principle of One für SAP-Technologien	101
2.6.2	SAP-Technologien des Principle of One	102
2.7	Simplifizierungsgrundsätze für eigene Anwendungs- entwicklung	110
2.7.1	Bestehende Anwendungen simplifizieren	110
2.7.2	Legacy-Frameworks und -Datenmodelle simplifizieren ...	112
2.7.3	Schnittstellentechnologien für Prozess- und Daten- integration simplifizieren	114
2.7.4	ABAP-Richtlinien	115
2.8	SAP-S/4HANA-Readiness von Anwendungen	117
2.8.1	Prüfung der Kompatibilität zu SAP S/4HANA vorbereiten	117
2.8.2	SAP-S/4HANA-Kompatibilität prüfen	119
2.8.3	Funktionale Korrektheit von Datenbankzugriffen prüfen	122
2.8.4	Weitere Themen bei der Umstellung von Code auf SAP S/4HANA	126
2.9	Sicherheit von Anwendungen	129
2.9.1	Sicherheit von SAP-Fiori-Apps	130
2.9.2	Sicherheit auf Anwendungsebene	130
2.9.3	Zugriffsschutz für CDS Views	133
2.10	Anwendungsentwicklung mit dem ABAP-Paketkonzept	138
2.10.1	Strukturierung durch Pakete	138
2.10.2	ABAP-Anwendungen strukturieren	142
2.11	Anwendungsentwicklung und der SAP-Standard	148

3 Entwicklung von Geschäftsobjekten 151

3.1 Datenmodellierung	152
3.1.1 Datenmodell definieren	152
3.1.2 Basic CDS Views erstellen	155
3.2 Transaktionale Core Data Services	162
3.2.1 Transaktionale CDS Views erstellen	165
3.2.2 Generiertes BOPF-Geschäftsobjekt testen	168
3.2.3 Transaktionale CDS Views klassifizieren	170
3.3 BOPF-Metamodell	172
3.3.1 Automatische Datenmodellierung	175
3.3.2 Standardisierte Umsetzung von Geschäftslogik	176
3.3.3 BOPF-Transaktions- und -Servicemanager	183
3.4 BOPF-Implementierung	187
3.4.1 Alternativen Schlüssel implementieren	187
3.4.2 Ermittlungen implementieren	190
3.4.3 Validierungen implementieren	193
3.4.4 Aktionen implementieren	196
3.4.5 Spezielle Techniken anwenden	199
3.5 Namenskonventionen	211
3.6 Entwicklung einer Benutzeroberfläche mit SAP Fiori Elements	214
3.6.1 Consumption Views anlegen	214
3.6.2 Consumption View über OData-Service veröffentlichen	216
3.6.3 Benutzeroberfläche erstellen	220

4 Techniken der Anwendungsentwicklung 225

4.1 Sperrkonzept	225
4.1.1 Implementierung eines optimistischen Sperrkonzepts	229
4.1.2 Technische Details des optimistischen Sperrkonzepts	231
4.2 BOPF-Puffermanagement	238
4.2.1 Puffer bei direkten lesenden und schreibenden Operationen	240
4.2.2 Puffer mit Alternativem Schlüssel	242
4.2.3 Puffer bei Zugriffen über Assoziationen	242

4.2.4	Puffer bei Zugriffen über QUERY-Funktionen	243
4.2.5	Performanceoptimierung durch BOPF-Puffer- management	244
4.3	LUW-Konzept	252
4.3.1	Das LUW-Konzept im ABAP-Programmiermodell	253
4.3.2	Herausforderungen beim Einsatz des LUW-Konzepts für Legacy-Datenmodelle	254
4.4	Protokollierung	256
4.4.1	Logging oder Journal	258
4.4.2	Technische Änderungsnachverfolgung	262
4.4.3	Fachliche Änderungshistorie	265
4.5	Berechtigungssteuerung	276
4.5.1	Implementierung einer Berechtigungsprüfung	279
4.5.2	Performanceoptimierung bei der Durchführung von Berechtigungsprüfungen	281
4.5.3	Privilegierter Modus und Berechtigungsprüfungen	282
4.6	Tests	284
4.6.1	Allgemeine Hinweise zu Tests	284
4.6.2	Automatisiertes Testen in ABAP	290
4.6.3	Automatisiertes Testen in SAPUI5	308

5 Entwicklung transaktionaler Benutzeroberflächen

5.1	Floorplans	317
5.2	SAP Fiori Elements	321
5.3	Annotationen	325
5.3.1	Annotationen für den List Report	327
5.3.2	Annotationen der Object Page	340
5.3.3	Annotationen für übergreifende Konzepte	346
5.3.4	Weitere Annotationen	355
5.3.5	Umgang mit Problemen	375
5.3.6	Erweiterungsmöglichkeiten	385
5.4	Freestyle-Entwicklung von SAP-Fiori-Anwendungen	399
5.4.1	Wahl des Entwicklungsansatzes	399
5.4.2	Freestyle-Entwicklung	400

5.5	Draft-Konzept	402
5.5.1	Neues Sperrkonzept	410
5.5.2	Draft-Konzept bei Legacy-Anwendungen	413
5.6	Ausblick auf das ABAP-RESTful-Programmiermodell	415
5.6.1	Vom klassischen ABAP zum ABAP-RESTful-Programmiermodell	416
5.6.2	Einsatz des ABAP-RESTful-Programmiermodells	418

6 Operative Analytik 425

6.1	Grundlagen	426
6.2	Analytische Core Data Services	427
6.2.1	Das analytische Datenmodell	428
6.2.2	Definition des analytischen Datenmodells	433
6.2.3	Parametrisierung des analytischen Modells und Filterung	447
6.2.4	Wichtige Annotationen	452
6.3	Visualisierung mit der Analytical List Page	454
6.3.1	Vorbereitende Schritte	457
6.3.2	Analytical List Page implementieren	458
6.3.3	Annotationen für die Analytical List Page implementieren	460
6.3.4	Analytical List Page testen	469
6.3.5	Filter implementieren	471
6.3.6	Daten aus Analytical List Page exportieren	477
6.3.7	Key Performance Indicators implementieren	478
6.4	Weitere Möglichkeiten der Visualisierung	481
6.4.1	SAP Lumira Discovery	482
6.4.2	SAP Analysis for Microsoft Office	483

7 Management von ABAP-Entwicklungsprojekten 485

7.1	Notwendige Skills	486
7.2	Architekturmanagement	487

7.3 Agile Entwicklung, Continous Integration und Delivery	491
7.3.1 Code-Repository	492
7.3.2 Parallelle Entwicklungen	495
7.3.3 Beispiel für einen agilen Entwicklungsprozess mit abapGit	501
7.3.4 Automatisierung der Prozesse	503
7.4 Qualitätsmanagement	509
7.4.1 Technische Schulden	509
7.4.2 Automatisierte und manuelle Qualitätsprüfungen	512
 Anhang	521
 A Technical Architecture Modeling	523
B Das Enterprise Procurement Model	527
C Literaturverzeichnis	533
D Die Autoren	535
 Index	537