

Auf einen Blick

| | | |
|---|--|-----|
| 1 | Architektur moderner ABAP-Anwendungen | 23 |
| 2 | Das neue ABAP-Programmiermodell für SAP Fiori | 53 |
| 3 | Anwendungsentwicklung im Kontext von SAP S/4HANA | 111 |
| 4 | Entwicklung von Geschäftsobjekten | 163 |
| 5 | Techniken der Anwendungsentwicklung | 237 |
| 6 | Entwicklung transaktionaler Benutzeroberflächen | 327 |
| 7 | Operative Analytik | 459 |
| 8 | Management von ABAP-Entwicklungsprojekten | 521 |

Inhalt

| | |
|------------------|----|
| Geleitwort | 13 |
| Vorwort | 15 |
| Einleitung | 17 |

1 Architektur moderner ABAP-Anwendungen 23

| | |
|--|-----------|
| 1.1 Eigenschaften und Architektur von SAP S/4HANA | 23 |
| 1.1.1 Entwicklungsprinzipien von SAP S/4HANA | 24 |
| 1.1.2 Deployment-Varianten von SAP S/4HANA | 28 |
| 1.2 Entwicklung von Unternehmensanwendungen mit SAP S/4HANA | 34 |
| 1.2.1 Anforderungen an moderne ABAP-Anwendungen | 35 |
| 1.2.2 Standardisierung der Anwendungsarchitektur | 37 |
| 1.2.3 Technologie für die Entwicklung von Benutzeroberflächen | 41 |
| 1.3 Entwicklung auf Basis von SAP NetWeaver 7.50 | 46 |
| 1.3.1 Simplifizierung und Redesign | 47 |
| 1.3.2 Wechsel der UI-Technologie | 48 |

2 Das neue ABAP-Programmiermodell für SAP Fiori 53

| | |
|---|-----------|
| 2.1 ABAP-Entwicklungsumgebung und -Syntax | 53 |
| 2.1.1 ABAP in Eclipse | 54 |
| 2.1.2 Moderne ABAP-Syntax | 55 |
| 2.2 Einsatzszenarien des ABAP-Programmiermodells | 59 |
| 2.2.1 Komponenten des ABAP-Programmiermodells | 59 |
| 2.2.2 Das ABAP-Programmiermodell in der Praxis | 62 |
| 2.3 Core Data Services | 66 |
| 2.3.1 ABAP CDS | 67 |

| | | |
|------------|--|------------|
| 2.3.2 | Typen von ABAP CDS Views | 73 |
| 2.3.3 | CDS-Annotationen | 79 |
| 2.3.4 | Eingangsparameter | 83 |
| 2.3.5 | Assoziationen | 84 |
| 2.3.6 | Performanceprobleme | 86 |
| 2.4 | Entwicklung von OData-Services mit SAP Gateway | 91 |
| 2.4.1 | Optionen zur Entwicklung des OData-Service | 91 |
| 2.4.2 | Registrierung des OData-Service | 94 |
| 2.4.3 | Testen des OData-Service mit dem SAP Gateway Client | 96 |
| 2.5 | Code Pushdown auf die SAP-HANA-Datenbank | 101 |
| 2.5.1 | Goldene Regeln von SAP | 102 |
| 2.5.2 | Strategien für die Realisierung des Code Pushdowns | 104 |

3 **Anwendungsentwicklung im Kontext von SAP S/4HANA** 111

| | | |
|------------|--|------------|
| 3.1 | Simplifizierung und das Principle of One von SAP | 111 |
| 3.1.1 | Principle of One für SAP-Technologien | 112 |
| 3.1.2 | SAP-Technologien des Principles of One | 112 |
| 3.2 | Simplifizierungsgrundsätze für eigene Anwendungsentwicklung | 121 |
| 3.2.1 | Bestehende Anwendungen simplifizieren | 122 |
| 3.2.2 | Legacy-Frameworks und -Datenmodelle simplifizieren ... | 124 |
| 3.2.3 | Schnittstellentechnologien für Prozess- und Datenintegration simplifizieren | 126 |
| 3.2.4 | ABAP-Richtlinien | 127 |
| 3.3 | SAP-S/4HANA-Readiness von Anwendungen | 129 |
| 3.3.1 | Prüfung der Kompatibilität zu SAP S/4HANA vorbereiten | 129 |
| 3.3.2 | SAP-S/4HANA-Kompatibilität prüfen | 131 |
| 3.3.3 | Funktionale Korrektheit von Datenbankzugriffen prüfen | 134 |
| 3.3.4 | Weitere Themen bei der Umstellung von Code auf SAP S/4HANA | 139 |

| | | |
|------------|---|-----|
| 3.4 | Sicherheit von Anwendungen | 141 |
| 3.4.1 | Sicherheit von SAP-Fiori-Apps | 142 |
| 3.4.2 | Sicherheit auf Anwendungsebene | 143 |
| 3.4.3 | Zugriffsschutz für CDS Views | 147 |
| 3.5 | Anwendungsentwicklung und der SAP-Standard | 152 |
| 3.6 | Automatisierte und manuelle Qualitätsprüfungen | 154 |
| 3.6.1 | Automatisierte Codeprüfungen | 154 |
| 3.6.2 | Manuelle Codeprüfungen | 160 |

4 Entwicklung von Geschäftsobjekten 163

| | | |
|------------|--|-----|
| 4.1 | Datenmodellierung | 164 |
| 4.1.1 | Datenmodell definieren | 164 |
| 4.1.2 | Basic CDS Views erstellen | 167 |
| 4.2 | Transaktionale Core Data Services | 173 |
| 4.2.1 | Transaktionale CDS Views erstellen | 176 |
| 4.2.2 | Generiertes BOPF-Geschäftsobjekt testen | 180 |
| 4.2.3 | Transaktionale CDS Views klassifizieren | 182 |
| 4.3 | BOPF-Metamodell | 184 |
| 4.3.1 | Automatische Datenmodellierung | 187 |
| 4.3.2 | Standardisierte Umsetzung von Geschäftslogik | 188 |
| 4.3.3 | BOPF-Transaktions- und -Servicemanager | 196 |
| 4.4 | BOPF-Implementierung | 200 |
| 4.4.1 | Alternativen Schlüssel implementieren | 200 |
| 4.4.2 | Ermittlungen implementieren | 202 |
| 4.4.3 | Validierungen implementieren | 206 |
| 4.4.4 | Aktionen implementieren | 209 |
| 4.4.5 | Spezielle Techniken anwenden | 212 |
| 4.5 | Namenskonventionen | 223 |
| 4.6 | Entwicklung einer Benutzeroberfläche mit SAP Fiori Elements | 226 |
| 4.6.1 | Consumption Views anlegen | 226 |
| 4.6.2 | Consumption View über OData-Service veröffentlichen | 229 |
| 4.6.3 | Benutzeroberfläche erstellen | 232 |

5 Techniken der Anwendungsentwicklung 237

| | | |
|------------|--|-----|
| 5.1 | Sperrkonzept | 237 |
| 5.1.1 | Implementierung eines optimistischen Sperrkonzepts | 241 |
| 5.1.2 | Technische Details des optimistischen Sperrkonzepts | 243 |
| 5.2 | BOPF-Puffermanagement | 250 |
| 5.2.1 | Puffer bei direkten lesenden und schreibenden Operationen | 252 |
| 5.2.2 | Puffer mit alternativem Schlüssel | 254 |
| 5.2.3 | Puffer bei Zugriffen über Assoziationen | 254 |
| 5.2.4 | Puffer bei Zugriffen über QUERY-Funktionen | 255 |
| 5.2.5 | Performanceoptimierung durch BOPF-Puffermanagement | 256 |
| 5.3 | LUW-Konzept | 265 |
| 5.3.1 | Das LUW-Konzept im ABAP-Programmiermodell | 266 |
| 5.3.2 | Herausforderungen beim Einsatz des LUW-Konzepts für Legacy-Datenmodelle | 267 |
| 5.4 | Protokollierung | 269 |
| 5.4.1 | Logging oder Journal | 271 |
| 5.4.2 | Technische Änderungsnachverfolgung | 275 |
| 5.4.3 | Fachliche Änderungshistorie | 278 |
| 5.5 | Berechtigungssteuerung | 289 |
| 5.5.1 | Implementierung einer Berechtigungsprüfung | 291 |
| 5.5.2 | Performanceoptimierung bei der Durchführung von Berechtigungsprüfungen | 293 |
| 5.5.3 | Privilegierter Modus und Berechtigungsprüfungen | 295 |
| 5.6 | Tests | 296 |
| 5.6.1 | Allgemeine Hinweise zu Tests | 297 |
| 5.6.2 | Automatisiertes Testen in ABAP | 302 |
| 5.6.3 | Automatisiertes Testen in SAPUI5 | 320 |

6 Entwicklung transaktionaler Benutzeroberflächen 327

| | | |
|------------|---------------------------------|-----|
| 6.1 | Floorplans | 329 |
| 6.2 | SAP Fiori Elements | 333 |

| | | |
|------------|---|-----|
| 6.3 | Annotations | 338 |
| 6.3.1 | Annotationen für den List Report | 339 |
| 6.3.2 | Annotationen der Object Page | 353 |
| 6.3.3 | Annotationen für übergreifende Konzepte | 362 |
| 6.3.4 | Weitere Annotationen | 370 |
| 6.3.5 | Umgang mit Problemen | 391 |
| 6.3.6 | Erweiterungsmöglichkeiten | 403 |
| 6.4 | SAP Fiori Tools | 417 |
| 6.5 | Freestyle-Entwicklung von SAP-Fiori-Apps | 422 |
| 6.5.1 | Wahl des Entwicklungsansatzes | 423 |
| 6.5.2 | Freestyle-Entwicklung | 424 |
| 6.5.3 | Gemischte Entwicklung | 425 |
| 6.6 | Draft-Konzept | 430 |
| 6.6.1 | Neues Sperrkonzept | 439 |
| 6.6.2 | Draft-Konzept bei Legacy-Anwendungen | 442 |
| 6.7 | Ausblick auf das ABAP-RESTful-Programmiermodell | 444 |
| 6.7.1 | Vom klassischen ABAP zum ABAP-RESTful- Programmiermodell | 445 |
| 6.7.2 | Einsatz des ABAP-RESTful-Programmiermodells | 448 |
| 6.7.3 | Zukünftige Features des ABAP-RESTful-Programmier- modells für die On-Premise-Entwicklung | 455 |

7 Operative Analytik 459

| | | |
|------------|---|-----|
| 7.1 | Grundlagen | 460 |
| 7.2 | Analytische Core Data Services | 461 |
| 7.2.1 | Das analytische Datenmodell | 462 |
| 7.2.2 | Definition des analytischen Datenmodells | 467 |
| 7.2.3 | Parametrisierung des analytischen Modells und Filterung | 481 |
| 7.2.4 | Wichtige Annotationen | 485 |
| 7.3 | Visualisierung mit der Analytical List Page | 488 |
| 7.3.1 | Vorbereitende Schritte | 490 |
| 7.3.2 | Analytical List Page implementieren | 492 |
| 7.3.3 | Annotationen für die Analytical List Page implementieren | 494 |
| 7.3.4 | Analytical List Page testen | 502 |

| | | |
|------------|---|------------|
| 7.3.5 | Filter implementieren | 504 |
| 7.3.6 | Daten aus der Analytical List Page exportieren | 510 |
| 7.3.7 | Key Performance Indicators implementieren | 512 |
| 7.4 | Weitere Möglichkeiten der Visualisierung | 514 |
| 7.4.1 | SAP Analytics Cloud | 515 |
| 7.4.2 | SAP Analysis for Microsoft Office | 519 |

8 Management von ABAP-Entwicklungsprojekten 521

| | | |
|------------|---|------------|
| 8.1 | Notwendige Skills | 522 |
| 8.2 | Architekturmanagement | 523 |
| 8.3 | Agile Entwicklung, Continuous Integration und Delivery | 526 |
| 8.3.1 | Code-Repository | 528 |
| 8.3.2 | Parallele Entwicklungen | 531 |
| 8.3.3 | Beispiel für einen agilen Entwicklungsprozess mit abapGit | 537 |
| 8.3.4 | Automatisierung der Prozesse | 539 |
| 8.4 | Technische Schulden | 545 |

Anhang 549

| | | |
|----------|---|------------|
| A | Technical Architecture Modeling | 551 |
| B | Das Enterprise Procurement Model | 555 |
| C | Literaturverzeichnis | 561 |
| D | Die Autoren | 563 |

| | |
|-------------|-----|
| Index | 565 |
|-------------|-----|