

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com

Library of Congress Cataloging-in-Publication Data

SOA governance : governing shared services on-premise and in the cloud /
Thomas Erl ... [et al.].

p. cm.

Includes bibliographical references.

ISBN-13: 978-0-13-815675-6 (hardback : alk. paper)

ISBN-10: (invalid) 0-13-815672-1 (hardback : alk. paper) 1. Service-oriented architecture (Computer science) 2. Business enterprises--Computer networks--Management. I. Erl, Thomas. II. Title: Service-oriented architecture governance.

TK5105.5828.S59 2011

004.6'54--dc22

2011003181

Copyright © 2011 SOA Systems Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671-3447

ISBN-13: 978-0-13-815675-6

ISBN-10: 0-13-815675-1

Text printed in the United States on recycled paper.

First printing April 2011

Editor-in-Chief

Mark L. Taub

Development Editor

Infinet Creative Group

Managing Editor

Kristy Hart

Project Editor

Betsy Harris

Copy Editor

Infinet Creative Group

Senior Indexer

Cheryl Lenser

Proofreaders

Kam Chiu Mok

Shivapriya Nagaraj

Catherine Shaffer

Williams Woods

Publishing

Pamela Janice Yau

Publishing Coordinator

Kim Boedigheimer

Cover Designer

Thomas Erl

Compositors

Bumpy Design

Nonie Ratcliff

Photos

Thomas Erl

Graphics

Infinet Creative Group

*To my supportive wife, Marie, and my children, Sam and Sophia,
for allowing me to take some family time to work on this book.*

—Stephen G. Bennett

*To my family, my colleagues, and my Nespresso® machine,
all of whom supported the creation of this book.*

—Thomas Erl

*Many fine adventures for the past, present, and future for my wife, Amy,
and my sons, Thomas and Jack. You guys make it all worthwhile.
Thanks to all of the great people that I've worked with on SOA and governance.
I've learned a lot from you.*

—Robert Laird

*In memory of my father, Stanley Moores, who died at 45
and whose company and counsel I miss more each year.*

—Robert Moores

To my family in appreciation of their continued support and encouragement.

—Robert D. Schneider

To all my friends and colleagues that helped me make this book a reality.

—Leo Shuster

Thanks and love to my wife, Silke, and my sons, Marc and Jonas.

—Andre Tost

*To my wife, Kelly, and children, Ethan and Sarah, for their loving support;
and to my colleagues for putting up with my endless rants and pontification.*

—Chris Venable

This page intentionally left blank

Contents at a Glance

CHAPTER 1: Introduction	1
CHAPTER 2: Case Study Background	17

PART I: FUNDAMENTALS

CHAPTER 3: Service-Oriented Computing Fundamentals	23
CHAPTER 4: SOA Planning Fundamentals	49
CHAPTER 5: SOA Project Fundamentals	79
CHAPTER 6: Understanding SOA Governance	121

PART II: PROJECT GOVERNANCE

CHAPTER 7: Governing SOA Projects	153
CHAPTER 8: Governing Service Analysis Stages	187
CHAPTER 9: Governing Service Design and Development Stages	221
CHAPTER 10: Governing Service Testing and Deployment Stages	277
CHAPTER 11: Governing Service Usage, Discovery, and Versioning Stages. . .	315

PART III: STRATEGIC GOVERNANCE

CHAPTER 12: Service Information and Service Policy Governance	369
CHAPTER 13: SOA Governance Vitality	411
CHAPTER 14: SOA Governance Technology	425

PART IV: APPENDICES

APPENDIX A: Case Study Conclusion	453
APPENDIX B: Master Reference Diagrams for Organizational Roles	457
APPENDIX C: Service-Oriented Principles Reference	473
APPENDIX D: SOA Design Patterns Reference	489
APPENDIX E: The Annotated SOA Manifesto	577
APPENDIX F: Versioning Fundamentals for Web Services and REST Services. .	591
APPENDIX G: Mapping Service-Oriented to RUP	617
APPENDIX H: Additional Resources	631
About the Authors	635
About the Contributors	641
About the Foreword Contributors	643
Index	645

This page intentionally left blank

Contents

Foreword by Massimo Pezzini	xxxi
Foreword by Roberto Medrano	xxxiii
Acknowledgments	xxxv
CHAPTER 1: Introduction	1
1.1 About this Book	3
Who this Book is For.	3
What this Book Does Not Cover.	4
<i>This is Not a Book About SOA Management</i>	4
<i>This is Not a Book About Cloud Computing Governance.</i>	4
1.2 Recommended Reading	5
1.3 How this Book is Organized	6
Part I: Fundamentals	6
<i>Chapter 3: Service-Oriented Computing Fundamentals</i>	6
<i>Chapter 4: SOA Planning Fundamentals</i>	6
<i>Chapter 5: SOA Project Fundamentals</i>	6
<i>Chapter 6: Understanding SOA Governance</i>	7
Part II: Project Governance	7
<i>Chapter 7: Governing SOA Projects</i>	7
<i>Chapter 8: Governing Service Analysis Stages</i>	7
<i>Chapter 9: Governing Service Design and Development Stages</i>	8
<i>Chapter 10: Governing Service Testing and Deployment Stages</i>	9
<i>Chapter 11: Governing Service Usage, Discovery,</i> <i>and Versioning Stages</i>	9

Part III: Strategic Governance	10
<i>Chapter 12: Service Information and Service Policy Governance</i>	10
<i>Chapter 13: SOA Governance Vitality</i>	11
<i>Chapter 14: SOA Governance Technology</i>	11
Part IV: Appendices	11
<i>Appendix A: Case Study Conclusion</i>	11
<i>Appendix B: Master Reference Diagrams for Organizational Roles</i>	11
<i>Appendix C: Service-Oriented Principles Reference</i>	11
<i>Appendix D: SOA Design Patterns Reference</i>	11
<i>Appendix E: The Annotated SOA Manifesto</i>	11
<i>Appendix F: Versioning Fundamentals for Web Services and REST Services</i>	12
<i>Appendix G: Mapping Service-Oriented to RUP</i>	12
<i>Appendix H: Additional Resources</i>	12
1.4 Symbols, Figures, and Style Conventions	12
Symbol Legend	12
Mapping Diagrams.	12
SOA Principles & Patterns Sections.	13
Capitalization	14
1.5 Additional Information	14
Updates, Errata, and Resources (www.soabooks.com)	14
Master Glossary (www.soaglossary.com)	15
Referenced Specifications (www.soaspecs.com)	15
SOASchool.com® SOA Certified Professional (SOACP)	15
CloudSchool.com™ Cloud Certified Professional (CCP)	15
The SOA Magazine (www.soamag.com)	15
Notification Service	16
CHAPTER 2: Case Study Background	17
2.1 How Case Studies are Used	18
2.2 Raysmoore Corporation	18
History	18
IT Environment	18
Business Goals and Obstacles	19
2.3 Case Study Continuation.	20

PART I: FUNDAMENTALS

CHAPTER 3: Service-Oriented Computing Fundamentals . . .23

- 3.1 Basic Terminology 24
 - Service-Oriented Computing 25
 - Service-Orientation 26
 - Service-Oriented Architecture (SOA). 29
 - Services 31
 - Services as Components* 32
 - Services as Web Services* 32
 - Services as REST Services* 34
 - SOA Manifesto 34
 - Cloud Computing 35
 - IT Resources 35
 - Cloud 36
 - On-Premise 37
 - Cloud Deployment Models. 37
 - Cloud Consumers and Cloud Providers. 38
 - Cloud Delivery Models. 38
 - Service Models. 38
 - Agnostic Logic and Non-Agnostic Logic* 39
 - Service Composition 40
 - Service Inventory 41
 - Service Portfolio 41
 - Service Candidate 42
 - Service Contract. 43
 - Service-Related Granularity 44
 - SOA Design Patterns 46
- 3.2 Further Reading 47

CHAPTER 4: SOA Planning Fundamentals.49

- 4.1 The Four Pillars of Service-Orientation 51
 - Teamwork 52
 - Education 52
 - Discipline 52
 - Balanced Scope 53

4.2 Levels of Organizational Maturity	56
Service Neutral Level	57
Service Aware Level	57
Service Capable Level	57
Business Aligned Level	58
Business Driven Level	58
Service Ineffectual Level	58
Service Aggressive Level	59
4.3 SOA Funding Models	60
Platform (Service Inventory) Funding	60
<i>Project Funding Model (Platform)</i>	61
<i>Central Funding Model (Platform)</i>	64
<i>Usage Based Funding Model (Platform)</i>	66
Service Funding	69
<i>Project Funding Model (Service)</i>	69
<i>Central Funding Model (Service)</i>	71
<i>Hybrid Funding Model (Service)</i>	72
<i>Usage Based Funding Model (Service)</i>	74

CHAPTER 5: SOA Project Fundamentals79

5.1 Project and Lifecycle Stages	81
SOA Adoption Planning	82
Service Inventory Analysis	82
Service-Oriented Analysis (Service Modeling)	84
Service-Oriented Design (Service Contract)	85
Service Logic Design	87
Service Development	87
Service Testing	88
Service Deployment and Maintenance	89
Service Usage and Monitoring	90
Service Discovery	90
Service Versioning and Retirement	91
5.2 Organizational Roles	92
Service Analyst	96
Service Architect	96
Service Developer	97
Service Custodian	98

Cloud Service Owner	98
Service Administrator	100
Cloud Resource Administrator	100
Schema Custodian	102
Policy Custodian	104
Service Registry Custodian	105
Technical Communications Specialist	105
Enterprise Architect	106
Enterprise Design Standards Custodian (and Auditor)	107
SOA Quality Assurance Specialist	109
SOA Security Specialist	110
SOA Governance Specialist	111
Other Roles	112
<i>Educator</i>	<i>112</i>
<i>Business Analyst</i>	<i>113</i>
<i>Data Architect</i>	<i>113</i>
<i>Technology Architect</i>	<i>113</i>
<i>Cloud Technology Professional</i>	<i>114</i>
<i>Cloud Architect</i>	<i>114</i>
<i>Cloud Security Specialist</i>	<i>114</i>
<i>Cloud Governance Specialist</i>	<i>114</i>
<i>IT Manager</i>	<i>115</i>
5.3 Service Profiles	115
Service-Level Profile Structure	117
Capability Profile Structure	118
Additional Considerations	119
<i>Customizing Service Profiles</i>	<i>119</i>
<i>Service Profiles and Service Registries</i>	<i>119</i>
<i>Service Profiles and Service Catalogs</i>	<i>119</i>
<i>Service Profiles and Service Architecture</i>	<i>120</i>

CHAPTER 6: Understanding SOA Governance 121

6.1 Governance 101	122
The Scope of Governance	123
<i>Governance and Methodology</i>	<i>124</i>
<i>Governance and Management</i>	<i>124</i>
<i>Methodology and Management</i>	<i>125</i>
<i>Comparisons</i>	<i>125</i>

The Building Blocks of a Governance System	127
<i>Precepts</i>	128
<i>People (Roles)</i>	128
<i>Processes</i>	129
<i>Metrics</i>	129
Governance and SOA	130
6.2 The SOA Governance Program Office (SGPO)	131
6.3 SGPO Jurisdiction Models	133
<i>Centralized Enterprise SGPO</i>	133
<i>Centralized Domain SGPO</i>	134
<i>Federated Domain SGPOs</i>	135
<i>Independent Domain SGPOs</i>	136
6.4 The SOA Governance Program.	137
Step 1: Assessing the Enterprise (or Domain)	137
<i>Current Governance Practices and Management Styles</i>	138
<i>SOA Initiative Maturity</i>	138
<i>Current Organizational Model</i>	139
<i>Current and Planned Balance of On-Premise and Cloud-based IT Resources</i>	139
Step 2: Planning and Building the SOA Governance Program	139
<i>SOA Governance Precepts</i>	139
<i>SOA Governance Processes</i>	141
<i>SOA Governance Roles</i>	143
<i>Additional Components</i>	146
Step 3: Running the SOA Governance Program (Best Practices and Common Pitfalls)	146
<i>Collect the Right Metrics and Have the Right People Use Them</i>	146
<i>Provide Transparency and Foster Collaboration</i>	147
<i>Ensure Consistency and Reliability</i>	147
<i>Compliance and Incentives</i>	147
<i>Education and Communication</i>	148
<i>Common Pitfalls</i>	148

PART II: PROJECT GOVERNANCE

CHAPTER 7: Governing SOA Projects 153

- 7.1 Overview 155
 - Precepts, Processes, and People (Roles) Sections 156
- 7.2 General Governance Controls 157
 - Precepts 157
 - Service Profile Standards* 157
 - Service Information Precepts* 158
 - Service Policy Precepts* 158
 - Logical Domain Precepts* 159
 - Security Control Precepts* 160
 - SOA Governance Technology Standards* 163
 - Metrics 164
 - Cost Metrics* 164
 - Standards-related Precept Metrics* 165
 - Threshold Metrics* 165
 - Vitality Metrics* 166
 - Case Study Example 167
- 7.3 Governing SOA Adoption Planning 169
 - Precepts 169
 - Preferred Adoption Scope Definition* 169
 - Organizational Maturity Criteria Definition* 171
 - Standardized Funding Model* 172
 - Processes 173
 - Organizational Governance Maturity Assessment* 173
 - Adoption Impact Analysis* 176
 - Adoption Risk Assessment* 178
 - People (Roles) 179
 - Enterprise Architect* 179
 - SOA Governance Specialist* 181
 - Case Study Example 182

CHAPTER 8: Governing Service Analysis Stages 187

- 8.1 Governing Service Inventory Analysis 192
 - Precepts 193
 - Service Inventory Scope Definition* 193
 - Processes 195
 - Business Requirements Prioritization* 195
 - People (Roles) 197
 - Service Analyst* 197
 - Enterprise Design Standards Custodian* 198
 - Enterprise Architect* 199
 - SOA Governance Specialist* 200
 - Case Study Example 201
- 8.2 Governing Service-Oriented Analysis
(Service Modeling) 206
 - Precepts 206
 - Service and Capability Candidate Naming Standards* 206
 - Service Normalization* 207
 - Service Candidate Versioning Standards* 209
 - Processes 210
 - Service Candidate Review* 210
 - People (Roles) 212
 - Service Analyst* 212
 - Service Architect* 213
 - Enterprise Design Standards Custodian* 214
 - Enterprise Architect* 215
 - SOA Governance Specialist* 216
 - Case Study Example 217

CHAPTER 9: Governing Service Design and Development Stages 221

- 9.1 Governing Service-Oriented Design (Service Contract) . . 223
 - Precepts 223
 - Schema Design Standards* 223
 - Service Contract Design Standards* 225
 - Service-Orientation Contract Design Standards* 228
 - SLA Template* 229
 - Processes 231
 - Service Contract Design Review* 231
 - Service Contract Registration* 234

People (Roles)	236
<i>Service Architect</i>	236
<i>Schema Custodian</i>	237
<i>Policy Custodian</i>	238
<i>Technical Communications Specialist</i>	239
<i>Enterprise Design Standards Custodian</i>	241
<i>Enterprise Architect</i>	242
<i>SOA Security Specialist</i>	243
<i>SOA Governance Specialist</i>	245
Case Study Example	246
9.2 Governing Service Logic Design	249
Precepts	249
<i>Service Logic Design Standards</i>	249
<i>Service-Orientation Architecture Design Standards</i>	252
Processes	253
<i>Service Access Control</i>	253
<i>Service Logic Design Review</i>	255
<i>Legal Data Audit</i>	257
People (Roles)	259
<i>Service Architect</i>	259
<i>Enterprise Design Standards Custodian</i>	260
<i>Enterprise Architect</i>	261
<i>SOA Security Specialist</i>	262
<i>SOA Governance Specialist</i>	263
Case Study Example	265
9.3 Governing Service Development	267
Precepts	267
<i>Service Logic Programming Standards</i>	267
<i>Custom Development Technology Standards</i>	268
Processes	270
<i>Service Logic Code Review</i>	270
People (Roles)	272
<i>Service Developer</i>	272
<i>Enterprise Design Standards Custodian</i>	273
<i>Enterprise Architect</i>	274
<i>SOA Governance Specialist</i>	275
Case Study Example	276

CHAPTER 10: Governing Service Testing and Deployment Stages277

10.1 Governing Service Testing.....	278
Precepts.....	279
<i>Testing Tool Standards</i>	279
<i>Testing Parameter Standards</i>	280
<i>Service Testing Standards</i>	281
<i>Cloud Integration Testing Standards</i>	283
<i>Test Data Usage Guidelines</i>	285
Processes.....	286
<i>Service Test Results Review</i>	286
People (Roles).....	287
<i>Service Administrator</i>	287
<i>Cloud Resource Administrator</i>	288
<i>Enterprise Architect</i>	289
<i>SOA Quality Assurance Specialist</i>	290
<i>SOA Security Specialist</i>	291
<i>SOA Governance Specialist</i>	292
Case Study Example.....	294
10.2 Governing Service Deployment and Maintenance.....	298
Precepts.....	298
<i>Production Deployment and Maintenance Standards</i>	298
Processes.....	301
<i>Service Certification Review</i>	301
<i>Service Maintenance Review</i>	303
People (Roles).....	304
<i>Service Administrator</i>	304
<i>Cloud Resource Administrator</i>	305
<i>Service Custodian</i>	307
<i>Enterprise Architect</i>	308
<i>SOA Quality Assurance Specialist</i>	309
<i>SOA Security Specialist</i>	310
<i>SOA Governance Specialist</i>	311
Case Study Example.....	312

Chapter 11: Governing Service Usage, Discovery, and Versioning Stages 315

- 11.1 Governing Service Usage and Monitoring 317
 - Precepts 317
 - Runtime Service Usage Thresholds* 317
 - Service Vitality Triggers* 320
 - Processes 323
 - Service Vitality Review* 323
 - People (Roles) 325
 - Enterprise Architect* 325
 - Service Architect* 326
 - Service Administrator* 327
 - Cloud Resource Administrator* 328
 - Service Custodian* 329
 - SOA Security Specialist* 331
 - SOA Governance Specialist* 332
 - Case Study Example 333
- 11.2 Governing Service Discovery 335
 - Precepts 335
 - Centralized Service Registry* 335
 - Processes 337
 - Service Registry Access Control* 337
 - Service Registry Record Review* 339
 - Service Discovery* 340
 - Shared Service Usage Request* 342
 - Shared Service Modification Request* 343
 - People (Roles) 345
 - Service Custodian* 345
 - Service Registry Custodian* 346
 - Technical Communications Specialist* 348
 - SOA Governance Specialist* 348
 - Case Study Example 350
- 11.3 Governing Service Versioning and Retirement 352
 - Precepts 352
 - Service Versioning Strategy* 352
 - SLA Versioning Rules* 354
 - Service Retirement Notification* 356

Processes	357
<i>Service Versioning</i>	357
<i>Service Retirement</i>	359
People (Roles)	360
<i>Enterprise Design Standards Custodian</i>	360
<i>Service Administrator</i>	362
<i>Cloud Resource Administrator</i>	363
<i>Schema Custodian</i>	364
<i>Policy Custodian</i>	364
<i>SOA Governance Specialist</i>	365

PART III: STRATEGIC GOVERNANCE

CHAPTER 12: Service Information and Service Policy Governance369

12.1 Overview	371
Service Data vs. Service Information	371
Policies 101	373
12.2 Governance Controls	375
Precepts	375
<i>Enterprise Business Dictionary/Domain Business Dictionary</i>	375
<i>Service Metadata Standards</i>	377
<i>Enterprise Ontology/Domain Ontology</i>	380
<i>Business Policy Standards</i>	382
<i>Operational Policy Standards</i>	384
<i>Policy Centralization</i>	386
Processes	389
<i>Data Quality Review</i>	389
<i>Communications Quality Review</i>	391
<i>Information Alignment Audit</i>	393
<i>Policy Conflict Audit</i>	395
People (Roles)	397
<i>Business Analyst</i>	397
<i>Data Architect</i>	399
<i>Schema Custodian</i>	399
<i>Policy Custodian</i>	401
<i>Service Registry Custodian</i>	402

Technical Communications Specialist 403

SOA Quality Assurance Specialist 405

SOA Governance Specialist 406

12.3 Guidelines for Establishing Enterprise Business Models 408

 Establish a Service Information Governance Council 408

 Assign Business Information Custodians 408

 Assign Value to Business Information 409

 Relate Service Information Governance to Master Data Management 409

CHAPTER 13: SOA Governance Vitality 411

13.1 Vitality Fundamentals 412

13.2 Vitality Triggers 414

Business vs. Technology Changes 415

Types of Vitality Triggers 416

 Strategic Adjustments 416

Strategic Business Adjustment 416

Strategic IT Adjustment 417

 Industry Shifts 417

Business Shift 417

Technology Shift 418

 Metrics 418

Performance Metrics 419

Compliance Metrics 419

 Organizational Shifts 419

 Periodic 420

Milestone 420

Time 420

13.3 SOA Governance Vitality Process 421

 Identify Activity 421

 Assess Activity 422

 Refresh Activity 422

 Approve Activity 423

 Communicate Activity 423

CHAPTER 14: SOA Governance Technology425

14.1 Understanding SOA Governance Technology	426
SOA Governance Task Types	427
<i>Manual Governance</i>	427
<i>Automated Governance</i>	427
<i>Design-time Governance</i>	428
<i>Runtime Governance</i>	428
<i>On-Premise Governance</i>	428
<i>Cloud Governance</i>	428
<i>Passive Governance</i>	428
<i>Active Governance</i>	429
SOA Governance Technology Types	429
<i>Administrative</i>	429
<i>Monitoring</i>	429
<i>Reporting</i>	430
<i>Enforcement</i>	430
14.2 Common SOA Governance Technology Products	431
Service Registries	431
<i>Task Types</i>	432
<i>Technology Types</i>	432
<i>SOA Project Stages</i>	433
Repositories	433
<i>Task Types</i>	434
<i>Technology Types</i>	434
<i>SOA Project Stages</i>	435
Service Agents	435
<i>Task Types</i>	436
<i>Technology Types</i>	437
<i>SOA Project Stages</i>	437
Policy Systems	437
<i>Task Types</i>	438
<i>Technology Types</i>	438
<i>SOA Project Stages</i>	439
Quality Assurance Tools.	439
<i>Task Types</i>	440
<i>Technology Types</i>	440
<i>SOA Project Stages</i>	441
SOA Management Suites	441

- Other Tools and Products. 442
 - Technical Editors and Graphic Tools* 442
 - Content Sharing and Publishing Tools* 442
 - Configuration Management Tools* 443
 - Custom SOA Governance Solutions* 443
- 14.3 Guidelines for Acquiring SOA Governance Technology 444
 - Acquisition Strategies. 444
 - Single Vendor* 444
 - Multiple Vendors* 445
 - Open Source* 446
 - Leased from Cloud Vendor* 447
 - Best Practices 448
 - Establish Criteria Based on Your Specific Requirements* 448
 - Investigate Customizability* 448
 - Investigate APIs* 448
 - Understand Both Initial and Long-Term Costs* 448
 - Understand Actual Governance Support* 449
 - Take the Time to Create a Quality RFP* 449

PART IV: APPENDICES

APPENDIX A: Case Study Conclusion453

**APPENDIX B: Master Reference Diagrams for
Organizational Roles457**

- Service Analyst. 458
- Service Architect 459
- Service Developer 460
- Service Custodian 460
- Service Administrator 461
- Cloud Resource Administrator 462
- Schema Custodian 463
- Policy Custodian 464
- Service Registry Custodian 465
- Technical Communications Specialist 466
- Enterprise Architect 467

Enterprise Design Standards Custodian (and Auditor)	468
SOA Quality Assurance Specialist	469
SOA Security Specialist	470
SOA Governance Specialist (precepts)	471
SOA Governance Specialist (processes)	472

APPENDIX C: Service-Orientation Principles Reference . . 473

APPENDIX D: SOA Design Patterns Reference 489

APPENDIX E: The Annotated SOA Manifesto 577

APPENDIX F: Versioning Fundamentals for Web Services and REST Services 591

F.1 Versioning Basics	593
Versioning Web Services	593
Versioning REST Services	594
Fine and Coarse-Grained Constraints	595
F.2 Versioning and Compatibility	596
Backwards Compatibility	596
<i>Backwards Compatibility in Web Services</i>	596
<i>Backwards Compatibility in REST Services</i>	597
Forwards Compatibility	599
Compatible Changes	602
Incompatible Changes	604
F.3 REST Service Compatibility Considerations	605
F.4 Version Identifiers	608
F.5 Versioning Strategies	611
The Strict Strategy (New Change, New Contract)	611
<i>Pros and Cons</i>	612
The Flexible Strategy (Backwards Compatibility)	612
<i>Pros and Cons</i>	613

The Loose Strategy (Backwards and Forwards Compatibility) . . .613
 Pros and Cons. 614
 Summary Table 614
F.6 REST Service Versioning Considerations 615

APPENDIX G: Mapping Service-Orientation to RUP 617

Compatibility of RUP and SOA 618
Overview of RUP (and MSOAM) 619
The Pillars of Service-Orientation and the RUP Principles. . . 620
Breadth and Depth Roles and Role Mapping 623
Enterprise and Governance Roles 624
 Mapping Service Delivery Project Stages to Disciplines 625
Mapping MSOAM Analysis and Design Stages to
RUP Disciplines 626
Service-Orientation and RUP: Gaps 628
 Related Reading. 628
Bibliography. 629

APPENDIX H: Additional Resources. 631

About the Authors 635

Stephen G. Bennett 635
Thomas Erl. 635
Clive Gee, Ph.D.. 636
Robert Laird. 637
Anne Thomas Manes. 637
Robert Schneider. 638
Leo Shuster 638
Andre Tost 639
Chris Venable. 639

About the Contributors641

Benjamin Carlyle 641

Robert Moores 641

Filippos Santas. 642

**About the Foreword
Contributors643**

Massimo Pezzini 643

Roberto Medrano 643

Index645

Foreword

by Massimo Pezzini

“What are the three key ingredients for successful SOA?” I was asked (in Sweden, if I remember well) by a pretty senior application architect several years ago. It was the time, circa 2004, when SOA was at the peak of what we at Gartner call “the hype cycle.” Every vendor was busily trying to reposition as a SOA player, and users were struggling to understand what SOA was and why they should care about it.

When that application architect asked me the fatal question, I had luckily already investigated SOA, especially its key “dos” and “don’ts,” for quite a while, starting in the late 1990s. I had by then spoken with quite a number of large organizations, in both North America and Europe, that had gone through the painful process of figuring out, through trial and error, how to manage a large-scale and business-critical set of SOA-based projects. Therefore, my answer was spontaneous and also came out with a rather unquestionable tone: “Discipline, discipline, and discipline!”

From my conversation with these leading-edge organizations, it was in fact pretty evident to me that what was later to be called SOA governance was a critical success factor for SOA initiatives. If you think about it for a second, this is obvious: The basic goals of SOA are

1. Reducing application development and maintenance costs, through run-time sharing of services across multiple applications
2. Increasing business agility, by effectively managing service and application life-cycle (discovery, definition, design, implementation, testing, deployment, management, maintenance, and retirement)

There is no way to achieve these goals without applying a proper set of rules and processes, which we now call SOA governance. SOA governance is in charge of making sure that services are designed and implemented to be truly reusable, that there are facilities in place (e.g., a *service repository* or *service inventory*, as it is called in this book) to enable a “reuse first” approach to application development, that ownership of (and accountability for) services is well defined and unambiguous, and that it is clear “who pays for what.” (You would be surprised to know how many SOA initiatives I analyzed came to a stalemate because of cost allocation issues....) SOA pioneers also discovered it was not sufficient to define SOA governance rules and processes. Without an organizational entity (the *SOA Center of Excellence* or *SOA Governance Program Office*, as it is called in this book) in charge of not only defining but also enabling and enforcing these rules and processes, they simply don’t happen.

You will find in this book a comprehensive and richly detailed interpretation of what these rules and processes are all about and how they can be concretely implemented. You may adopt and adapt these suggestions to your actual business and technical requirements, level of SOA maturity, organizational settings, and your company’s business and IT culture. The variety of case studies discussed in the book will also give you a sense of how concretely SOA governance can be implemented to achieve real-life business goals.

Let me conclude with a final “lesson learned” from the SOA governance trenches: Your SOA initiative may be killed by lack of governance, but too much governance can be deadly, too. Figuring out what is the “just enough” amount of governance appropriate for your company is a difficult, but worthwhile task. This book will help you accomplish that goal.

—*Massimo Pezzini*

VP and Research Fellow, Gartner, Inc.

Foreword

by Roberto Medrano

We have spent the better part of the last decade working on SOA governance programs at some of the world's largest and most complex IT organizations. We are very pleased, therefore, to see this important topic addressed in detail by Thomas Erl, one of this generation's truly great software architecture authors. Thomas' book is beyond timely, in our view, as it captures a serious truth that has crept up on even some of the most savvy CIOs. That is, SOA has gone from "nice to have," to "have to have," to today's reality that SOA is *just here*. Period. You have it. You don't have any choice but to have it. And now that you have, you have to govern it.

How did this happen? How did SOA emerge from the egghead shadows to become the de facto enterprise architecture across the globe? Many factors contributed to this situation, but perhaps most important has been the ascendancy of cloud computing. Though still in its infancy, cloud computing has been absolutely transformative in the role that SOA plays in day-to-day enterprise computing. The cloud is inherently service-oriented. Whether an organization is totally cloud-based, a hybrid of on-premise and cloud, or using a private cloud, its applications are now reaching out to consume and expose Web services in ways that would have been hard to imagine even a few years ago. Even organizations that shunned SOA now have one. It's called the cloud, and it's here to stay.

SOA governance and the cloud are vital companions, for better or worse. In a nutshell SOA governance is about making sure the enterprise builds the right things, build them right, and makes sure that what it has built is behaving right. With proper SOA governance, the cloud can be a strategic bonanza, smoothing the way for improving agility,

reducing risks, reducing costs and economies that everyone should want. Companies realizing the most success are those that have built a Unified SOA Governance infrastructure that governs a wide range of assets and artifacts through their entire lifecycle. Without SOA governance, the cloud threatens operational disaster and exposure to multiple levels of risk. And now, we have a thorough and well thought out book on the subject. Thomas has done the industry a great service by delving deeply into this topic in a way that readers of many different backgrounds can understand.

This book works because it gives the reader a sense of SOA governance across the full IT lifecycle and spans the organizations that are charged with managing the SOA. Thomas offers valuable insights and pragmatic tips on how to implement governance that is sensible yet effective, touching on managerial and business issues as much as technology. He probes into the nature of rules and organizations, even human nature, as he lays out the groundwork for good governance. Thomas understands that all of these aspects of governance are relevant to the success of a program. Enjoy this book. If you are involved in IT management, you will find it an indispensable companion in your quest for success with SOA.

—*Roberto Medrano*
EVP, SOA Software

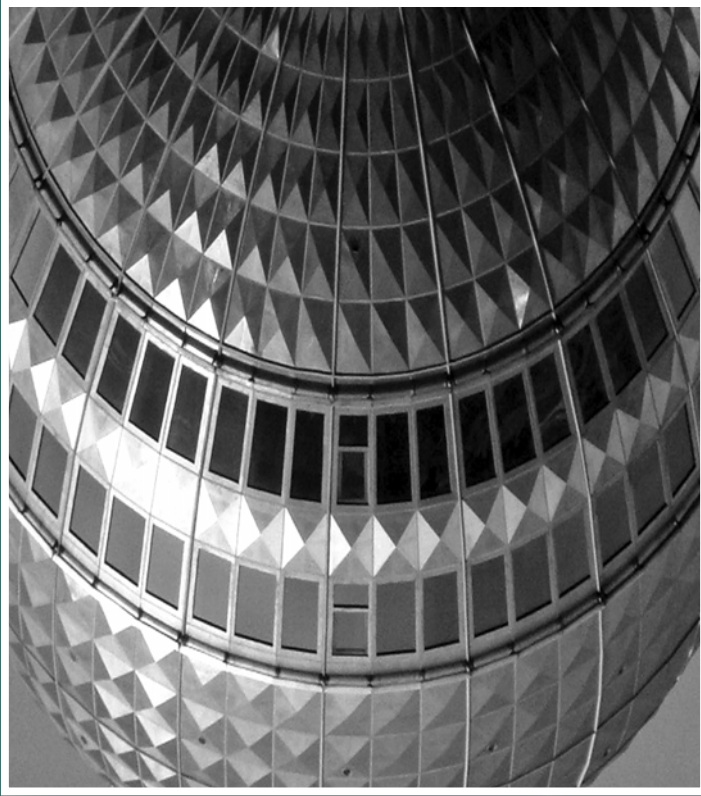
Acknowledgments

Special thanks to the following reviewers who generously volunteered their time and expertise (in alphabetical order):

Mohamad Afshar
Kristofer Ågren
Randy Atkins
Jean-Paul De Baets
Toufic Boubez
Benjamin Robert Carlyle
Pethuru Chelliah
Kevin P. Davis, Ph.D.
Mike Fields
Damian Kleer
Hanu Kommalapati
Nick Laqua
Charles N. Mead, MD, MSc
David E. Michalowicz
Thomas M. Michelbach
Kam Chiu Mok
Robert Moores
Eric Roch
David S. Rogers
Filippos Santas
Mark Sigsworth
Sanjay Kumar Singh
Herbjörn Wilhelmsen
Pamela Janice Yau
Dr. Jure Zakotnik
Dr. Matthias Ziegler

This page intentionally left blank

Chapter 6



Understanding SOA Governance

6.1 Governance 101

6.2 The SOA Governance Program Office (SGPO)

6.3 SGPO Jurisdiction Models

6.4 The SOA Governance Program

The expectation when adopting service-orientation is the realization of a number of specific strategic business benefits, as explained in Chapter 3. To accomplish this requires not only sound technology, mature practices, and sufficient stakeholder support, but also a firm grasp of the strategic target state being realized by the adoption and a firm system of ensuring its attainment and sustainment. Such a system cannot be purchased with technology products labeled as governance tools; it is a system that requires careful definition specific to overarching goals and requirements.

Structured governance is required to carry out and see through the commitments made when embarking on an SOA roadmap. It helps organizations succeed with SOA adoption efforts by mitigating risks through predefined constraints, rules, and the allocation of necessary authority. This chapter provides an introduction to general governance concepts and terms, as well as fundamental topics regarding governance systems for SOA projects.

6.1 Governance 101

Governance is the act of governing or administrating something. By far the most common form of governance is that of an organization. A system of governance is therefore generally a type of organizational system. For example, a society uses an organizational system to govern a public community. A company uses an organizational system to govern its own internal community.

A system for organizational governance exists as a meta-decision system. In other words, it is not just a means by which the organization makes decisions, it is the means by which the organization makes decisions *about* decision-making.

Within this context, a governance system:

- places constraints on decisions
- determines who has responsibility and authority to make decisions
- establishes constraints and parameters that control, guide, or influence decisions
- prescribes consequences for non-compliance

At the highest level in society, governance is established by a constitution. Within a company, it may be declared in the form of a business charter. Founding documents such as these establish a parent level of authority and constraints from which all other decision-making authorities and structures are derived. At deeper levels within the organization, a governance system can further influence the definition of policies, standards, and processes that guide and control day-to-day decision-making activities.

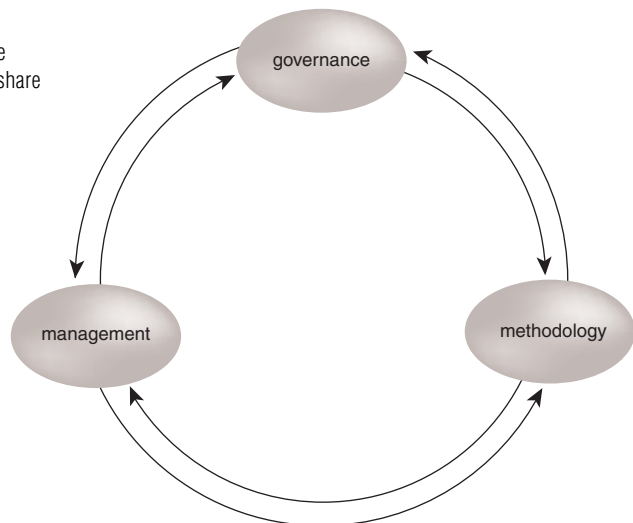
A good system of governance helps the members of an organization carry out responsibilities in a manner supportive of the organization's business goals and vision. It mitigates conflict by clearly defining responsibilities and assignments of authority, and further reduces ambiguity by articulating constraints and parameters in practical forms (such as rules and decision guidelines). It also helps balance tactical and strategic goals by expressing the intents and purposes of its rules.

The Scope of Governance

Within IT, a governance system is responsible for providing organization, direction, and guidance for the creation and evolution of IT assets and resources. To fully understand the scope of a governance system within a given IT department, we need to determine how a governance system relates to and is distinguished from methodology and management (Figure 6.1).

Figure 6.1

Governance, management, and methodology are distinct areas within an IT department that also share distinct relationships.



Governance and Methodology

Methodology represents a system of methods. Within IT, the form of methodology we are generally concerned with is that used to create software programs and business automation solutions. In this context, the methodology determines a system of methods used to conceptualize, design, program, test, and deploy a software program. These methods are generally formalized as a series of step-by-step processes that correspond to project delivery lifecycle stages.

NOTE

The Mainstream SOA Methodology (MSOAM) has established itself as a common, generic methodology for SOA project delivery. This methodology is explained in parts throughout the *Prentice Hall Service-Oriented Computing Series from Thomas Erl*, and is further summarized at www.soamethodology.com. Appendix G provides a supplementary paper that maps MSOAM to the Rational Unified Process (RUP).

Different software delivery methodologies exist. What commonly distinguishes one from the other is how they prioritize tactical and strategic requirements in relation to overarching business goals. These priorities will usually result in different processes (project lifecycle stages) being combined or organized in different ways. In some cases, one methodology may introduce a new process that does not exist in other methodologies—or it may exclude a process that commonly exists in other methodologies. Frequently, however, it comes down to how much time and effort a given process or project lifecycle stage receives, as determined by the tactical and strategic priorities of the methodology.

How a methodology is defined and carried out is heavily influenced by the governance system. Essentially, the methodology must be determined so that it follows the constraints established by the governance system and the corresponding methods (processes) must be carried out in compliance with these constraints, as well as any additional constraints that may be further introduced by the methodology itself.

Governance and Management

Whereas a governance system establishes rules and constraints, it is not responsible for enforcing them or overseeing related activities to ensure compliance. Management refers to the system and resources responsible for day-to-day operations.

Within an IT environment, this basically pertains to the execution of activities. In relation to governance, a management system provides the hands-on means by which the

constraints and goals of the governance system are realized in the real world. Therefore, the management of a governance system represents a subset of the overall management responsibilities.

Management systems are assigned to and carried out by those with authority.

Methodology and Management

Management relates to methodology the same way it relates to governance. When building software programs according to a pre-defined methodology, a management system is used to ensure the proper execution of processes and project delivery lifecycle stages in compliance with the constraints of the methodology—and the constraints of the governance system.

Comparisons

The following list contains a series of sample distinctions to further help provide a clear separation between governance, methodology, and management:

- Governance establishes rules that control decision-making.
- Methodology establishes processes that comply to governance rules and may introduce additional rules.
- Management makes decisions according to governance rules.
- Governance does not dictate when or how to make a decision. It determines who should make the decision and establishes limits for that person or group.
- Methodology establishes processes that carry out specific types of decision logic that adhere to governance rules.
- Management is responsible for day-to-day operations and for ensuring that decisions made adhere to governance and methodology rules.
- Governance cannot replace management or methodology, nor can it compensate for poor management or poor (or inappropriate) methodology.
- Poorly defined and executed methodology can jeopardize the business goals associated with governance.
- Poor management can undermine a governance system and a methodology and will jeopardize associated business goals.
- Neither management nor methodology can replace governance, nor compensate for poor governance.

- A poor governance system inevitably inhibits the ability of a methodology to fulfill business automation requirement potential.
- A poor governance system inevitably inhibits the ability of management to make correct decisions.

As previously stated, while this book will make many references to management and methodology, it is primarily focused on governance.

STYLES OF GOVERNANCE

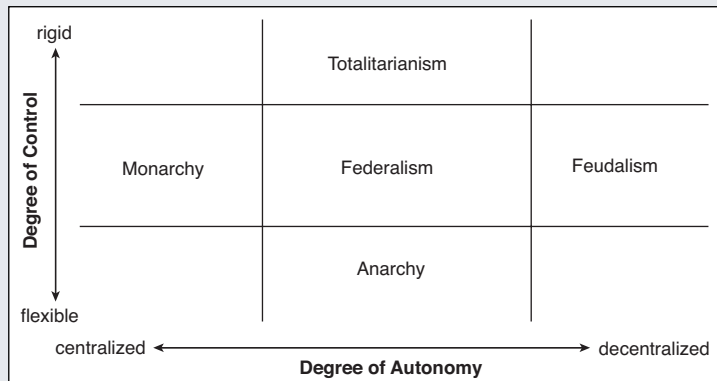
Governance must reflect and complement an organization's culture and structure. For example, when establishing suitable governance rules, considerations such as the following need to be raised:

- How much autonomy should each division, business unit, or department have?
- How much freedom should decision-makers have to delegate responsibilities to others?
- How much freedom should decision-makers have to use their own judgment when making decisions (as opposed to making decisions fully or partially based on pre-determined criteria)?

To determine what style of governance may be the best fit for a given organization, it can be helpful to refer to established forms of governance used historically in society. Figure 6.2 illustrates two dimensions that relate common governance styles.

Figure 6.2

The horizontal axis represents the degree of autonomy given to separate people or groups. The vertical axis represents the degree of control imposed on decision-makers.



Looking at one end of the horizontal spectrum, all decision-making is centralized, which is comparable to a monarchy. At the other end, each group establishes its own policies and procedures, similar to a feudal society. Many IT departments opt for a federated model, which permits the separation of the department into individual business units or cost centers, each of which is given a degree of independence while still maintaining a level of consistency. This helps reduce contention between fiefdoms.

When we study the vertical spectrum, we have a totalitarian type of regime whereby rigid policies dictate required actions, and decision-makers have little freedom to apply their own judgment. Too much rigidity can generate resentment and inhibit creativity in an organization. On the other hand, allowing flexible policies that provide only suggestive guidance leaves decision-makers with so much freedom that there is little chance of achieving meaningful consistency.

Good governance empowers people to do what's right for the business. Poor governance unnecessarily constrains or inhibits decisions, or fails to provide enough decision-making guidance. All governance—whether good or bad—places limits on the decisions and behaviors of the people being governed. It also prescribes consequences for those choosing not to abide. There is no single governance style that is correct for all organizations. Each must strive to find a balance between centralization and decentralization, between rigidity and flexibility, and between its existing culture and its ability to adapt to new approaches.

The Building Blocks of a Governance System

So far we've established that governance provides a systematic way for organizations to make decisions. Let's take a closer look at the primary building blocks that comprise a governance system:

- *precepts* define the rules that govern decision-making
- *people* assume roles and make decisions based on precepts
- *processes* coordinate people and precept-related decision-making activities
- *metrics* measure compliance to precepts

Note that these building blocks can be collectively or individually referred to as *governance controls*.

Precepts

A *precept* is an authoritative rule of action. Precepts are the essence of governance because they determine who has authority to make decisions, they establish constraints for those decisions, and they prescribe consequences for non-compliance.

Precepts codify decision-making rules using:

- *objectives* – broadly define a precept and establish its overarching responsibility, authority, and goals
- *policies* – define specific aspects of a precept and establish decision-making constraints and consequences
- *standards* – specify the mandatory formats, technologies, processes, actions, and metrics that people are required to use and carry out in order to implement one or more policies
- *guidelines* – are non-mandatory recommendations and best practices

NOTE

Within some IT communities, the term “policy” is commonly used instead of “precept” in relation to governance systems. However, as just explained, a policy can be just one aspect of a precept.

Also, even though a precept can contain standards, certain precepts themselves are considered standards. Therefore, it is important to not be confused when the precept name includes the word “standard” (such as Service Design Standard precept), and the precept itself further contains one or more standards that support corresponding precept policies.

People (Roles)

People (and groups of people) make decisions in accordance to and within the constraints stipulated by governance precepts. For a governance system to be successful, people must understand the intents and purposes of the precepts and they must understand and accept the responsibilities and authorities established by the precepts. Governance systems are therefore often closely associated with an organization’s incentive system. This allows the organization to foster a culture that supports and rewards good behavior, while also deterring and punishing poor behavior.

When exploring the involvement of people in relation to governance systems, it is further necessary to identify the role or roles they assume. Organizational roles position people (and groups) in relation to governance models and further affect the relevance of precept compliance and enforcement.

There are two ways that people can relate to precepts and processes: they can help author the precepts and processes and they can be dictated by their application. In this book, we explore both types of relationships.

Processes

A process is an organized representation of a series of activities. It is important to make a distinction between governance processes and other types of processes related to IT. Governance processes provide a means by which to control decisions, enforce policies, and take corrective action in support of the governance system. Other processes, such as those employed to carry out project delivery stages, can be heavily influenced by governance precepts, but are not specifically processes that are directly related to carrying out the governance system. Technically, any process is considered a management activity, but a governance system is dependent on governance processes to ensure compliance with its precepts.

An organization is likely to use a variety of processes to support its precepts. Some may be automated, while others require human effort. Automated processes can help coordinate tasks (such as steps required to collect data for approvals), but can still rely on people to make important decisions (such as making the actual approvals based on the presented data).

Metrics

Metrics provide information that can be used to measure and verify compliance with precepts. The use of metrics increases visibility into the progress and effectiveness of the governance system. By analyzing metrics, we gain insight into the efficacy of governance rules and we can further discover whether particular precepts or processes are too onerous or unreasonable. Metrics also measure trends, such as the number of violations and requests for waivers. A large number of waiver requests may indicate that a given precept might not be appropriate or effective.

Governance and SOA

An organization establishes governance to mitigate risk and to help advance its strategy, goals, and priorities. When the organization invests in an SOA initiative, it expects to gain benefits worth more than the cost of the investment. This return on investment is measured in terms of business outcomes, and, presumably, those outcomes reflect the organization's strategy, goals, and priorities. Therefore, the primary business goal for SOA governance is to ensure that an SOA initiative achieves its targeted business outcome.

An SOA governance system is the meta-decision system that an organization puts in place to control and constrain decision-making responsibilities related to the adoption and application of service-orientation. There are many practices, considerations, models, and frameworks that can comprise a meta-decision system suitable for SOA governance, all of which are explored throughout this book. The foundation of an SOA governance system resides within an SOA Governance Program Office responsible for creating and administering an SOA governance program that encompasses and defines necessary SOA governance models and the tasks required to realize and sustain these models.

NOTE

The term "SOA Governance Program Office" is intentionally capitalized as it represents the official name of an IT department. The term "SOA governance program" is not capitalized, as it refers to a type of program that is commonly assigned its own unique name.

SUMMARY OF KEY POINTS

- There are clear distinctions between governance, methodology, and management.
 - The building blocks of a governance system are precepts, people, processes, and metrics.
 - The fundamental steps to laying the foundation for an SOA governance system are to create an SOA Governance Program Office that creates and administers an SOA governance program.
-

6.2 The SOA Governance Program Office (SGPO)

NOTE

For simplicity's sake this chapter frequently uses the acronym "SGPO" for the "SOA Governance Program Office." This is not an industry-standard acronym, nor is the book proposing it as such. It is an acronym used solely to simplify content by avoiding repeatedly spelling out this term.

The first step in any SOA governance effort is to establish a group (or department) that assumes the responsibility of defining and administering the various parts of an SOA governance system. This group forms the SOA Governance Program Office (SGPO), an organizational entity that is commonly comprised of trained SOA Governance Specialists, Enterprise Architects, and other types of IT decision-makers. The SGPO is given the authority to define and enforce the on-going activities and rules associated with SOA governance.

A primary responsibility of the SGPO is to author a series of formal precepts. In some cases, the SGPO may need to request amendments to existing IT governance precepts to accommodate the distinct needs of SOA projects, as the SGPO needs to avoid inadvertently defining conflicting precepts.

In general, SOA governance precepts are more balanced and more easily accepted when those who are governed have a voice. The SGPO may therefore need to solicit input from major stakeholders, including IT and business managers, senior IT staff, and even the legal department. Those contributing should have an opportunity to comment on pending precepts, propose amendments, and recommend new precepts. However, just because the SGPO solicits input does not imply that it is relinquishing its authority to establish the necessary SOA governance precepts.

Following are some basic guidelines for incorporating the SGPO into an IT environment:

- The SGPO must have the responsibility and authority to develop and manage the SOA governance system, and other teams must accept the SGPO's authority.
- The SGPO must ensure that the SOA governance system aligns with the organization's incentive and disciplinary systems.
- The SGPO must develop collaborative working relationships with other governance teams whose responsibilities intersect with those of the SGPO.

- The SGPO must ensure that its precepts align with other governance systems (Figure 6.3) within the company, or they must work with the other governance program offices to amend the conflicting precepts.
- The SGPO must have access to communication channels to disseminate information about the governance precepts and to provide training to people affected by them.

Corporate Governance

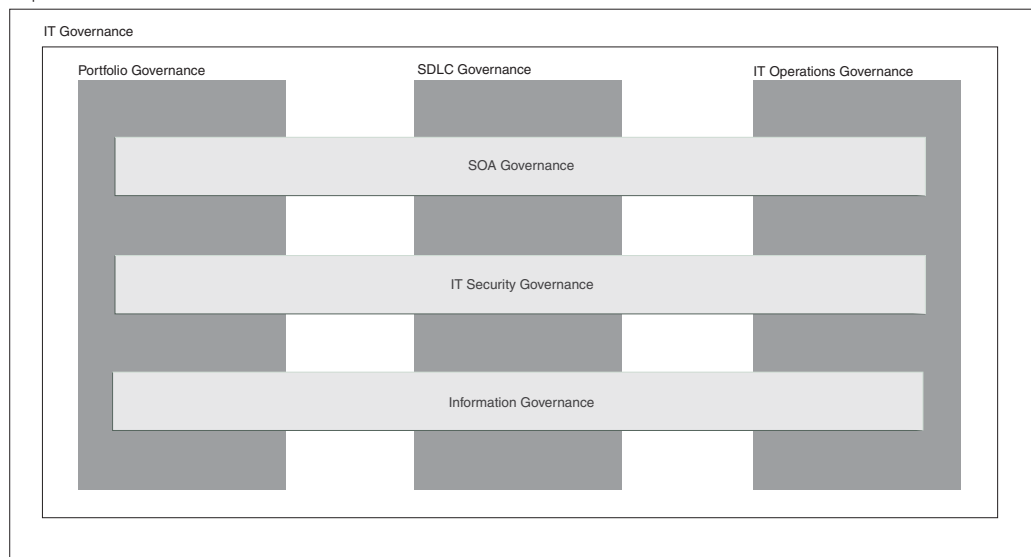


Figure 6.3

SOA governance must be defined through a program that can harmoniously co-exist alongside other IT governance programs.

What's of critical importance is that an appropriate scope be established for the SGPO. There are two primary factors that determine this scope: the reach of the SGPO within the overall IT enterprise and the areas of responsibility assumed by the SGPO within whatever domain it operates.

6.3 SGPO Jurisdiction Models

As explained in Chapter 3, a given IT enterprise can have one or more service inventories. Each service inventory represents a collection of independently standardized and governed services. When an IT enterprise has multiple service inventories, each is (ideally) associated with a well-defined domain, such as a line of business. In this case, service inventories are further qualified with the word “domain.”

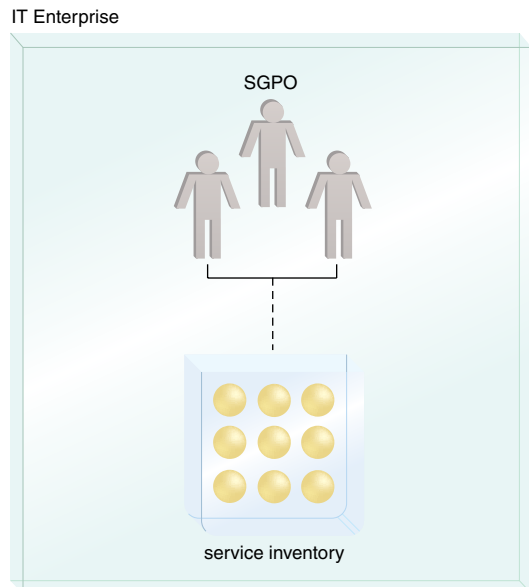
Depending on whether domain service inventories are being used and depending on how cooperative relations are between different service inventory owners, there may or may not be the opportunity to have one SGPO assume responsibility for multiple domain service inventories. As a result, different jurisdiction models exist, as follows:

Centralized Enterprise SGPO

If a single enterprise service inventory has been established, then it is generally expected that SOA governance responsibilities will be assigned to a single SGPO that oversees SOA governance on behalf of the entire IT enterprise.

Figure 6.4

A single SGPO responsible for the enterprise service inventory.



Centralized Domain SGPO

Even though individual domain service inventories can be independently standardized, managed, and owned, with enough cooperation between the owners, the IT department may be able to establish a single, enterprise-wide SGPO that subjects all service inventories to a common SOA governance system.

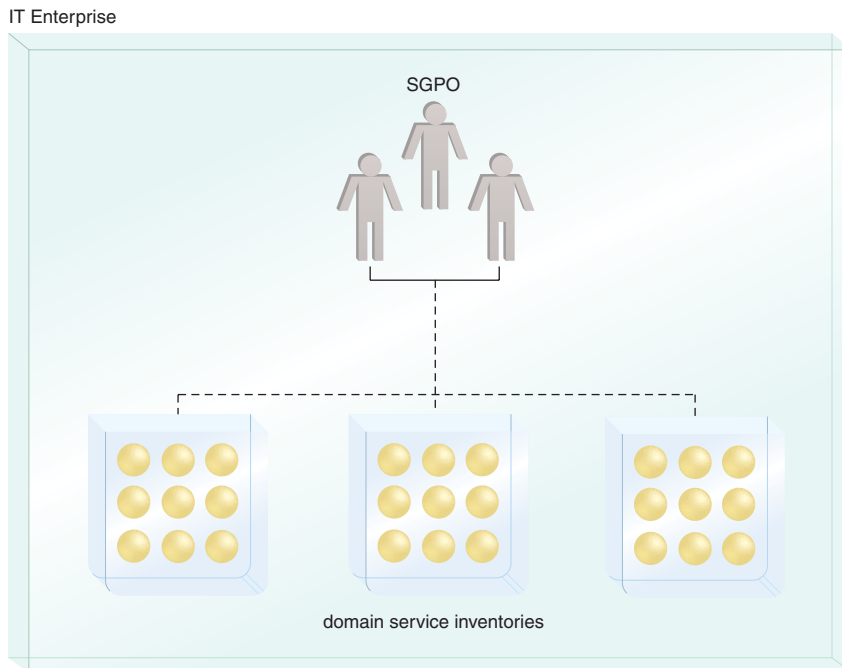


Figure 6.5

A single SGPO responsible for multiple domain service inventories.

Alternatively, different SOA governance programs can be created for each or select domain service inventories. With this model, separate programs can still be defined and maintained by the same central SGPO. The primary benefit of doing so is to maintain consistency and enterprise-wide alignment of how SOA governance programs are created and carried out, despite the fact that the respective SOA governance systems vary.

Federated Domain SGPOs

In this model, a central overarching SGPO exists in addition to individual SGPOs, each responsible for a separate domain service inventory. The domain SGPOs carry out individual SOA governance programs; however, these programs are required to comply to a set of conventions and standards defined by a single parent SGPO. The intent of this model is to strike a balance between domain-level independence and enterprise-wide consistency.

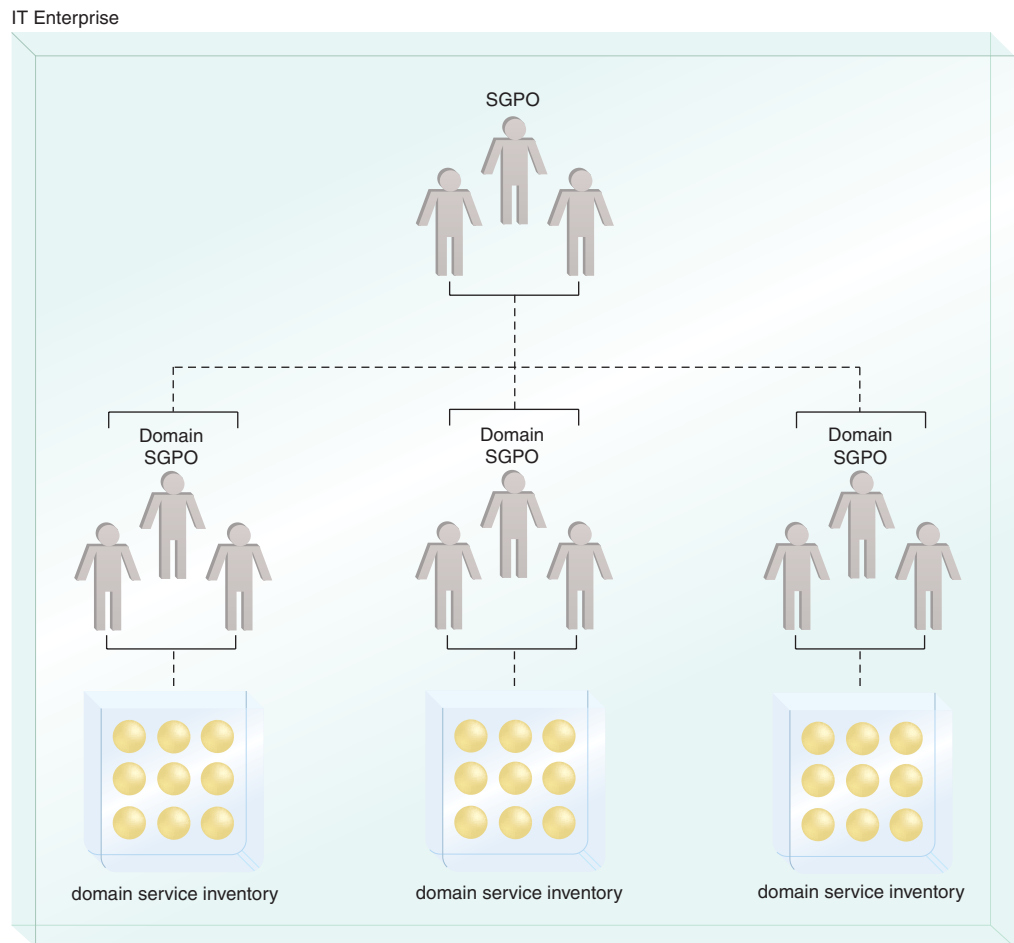


Figure 6.6

Multiple domain SGPOs are further “governed” by a central overarching SGPO.

Independent Domain SGPOs

Each domain service inventory has its own SGPO, which has full governance authority and jurisdiction over that domain. With the absence of a centralized SGPO presence, independent domain-level SGPOs have complete freedom to define and execute respective SOA governance programs.

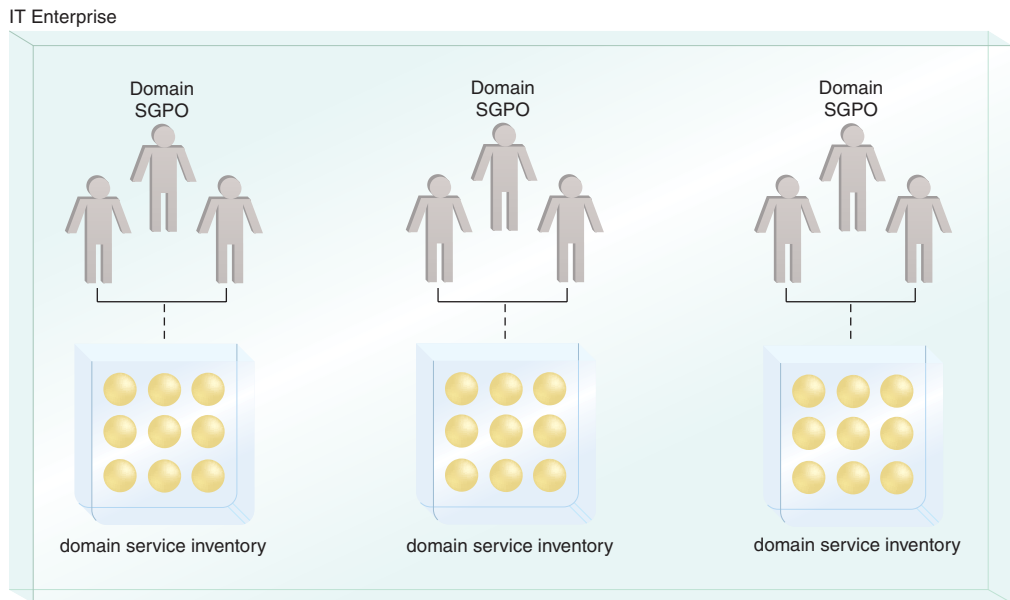


Figure 6.7

Multiple domain SGPOs independently govern multiple domain service inventories.

SUMMARY OF KEY POINTS

- The SGPO is an organizational entity responsible for defining and administering the SOA governance program.
 - The SGPO needs to be carefully positioned within the overall IT department to ensure alignment with existing governance groups and programs.
 - Different SGPO jurisdiction models can be considered, depending on the SOA adoption approach taken by an organization.
-

6.4 The SOA Governance Program

The SGPO exists to create and maintain an *SOA governance program*. This program encompasses the SOA governance system and all associated responsibilities for planning, implementing, and evolving this system. The best way to distinguish the program from the system is to view the SOA governance system as a set of formal precepts, roles, processes, metrics, and any associated models. The SOA governance program is dedicated to establishing and evolving the SOA governance system and therefore further provides real-world planning and implementation considerations, such as project plans, budgets, schedules, milestones, and further deliverables that map the SOA governance system to other parts of the existing IT enterprise (including already established IT governance systems).

The task of realizing an SOA governance program can be divided into three basic steps:

1. Assessing the Enterprise (or Domain)
2. Planning and Building the SOA Governance Program
3. Running the SOA Governance Program

Step 1: Assessing the Enterprise (or Domain)

Before creating appropriate precepts and formalizing the overall SOA governance system, the SGPO must first evaluate specific aspects of the current organizational state of the IT enterprise or whatever domain thereof for which that SOA adoption is being planned. This assessment may be limited to the domain in which the SGPO operates, but often also encompasses broader, organization-wide considerations that apply to most or all domains.

The assessment generally focuses on several specific areas:

- Current Governance Practices and Management Styles
- SOA Initiative Maturity
- Current Organizational Model
- Current and Planned Balance of On-Premise and Cloud-based IT Resources

Current Governance Practices and Management Styles

The organization's existing governance practices and management styles need to be studied to determine how best to introduce SOA governance-related processes and precepts. As previously described, no one governance model is suitable for every organization. A successful SOA governance program must take into account the organization's culture and management preferences.

Common issues that need to be addressed include:

- Are decisions tightly controlled by a central authority or widely delegated?
- Do the various groups within the organization collaborate or do they typically work autonomously?
- How do other governance program offices in the company work?
- How well does the organization articulate and disseminate governance precepts?
- How rigorously do people within the organization adhere to standard practices and processes?
- How much flexibility do managers and project leaders have in adapting to processes to meet the needs of a specific project?
- How much flexibility does management have to establish or modify incentive systems?

Concrete, well-researched answers to these questions can significantly influence an SOA governance program in that they can identify both strengths and weaknesses in relation to the types of governance and management practices required to see through a successful SOA initiative. This, in turn, helps determine the nature of precepts required and to what extent the existing IT culture will be impacted by the SOA governance system.

SOA Initiative Maturity

Ideally, an SOA governance program is established prior to the launch of an SOA initiative. However, in situations where existing SOA projects or activities are already underway, a further analysis of their progress and maturity is required to ensure that the introduction of the SOA governance program ends up supporting and aligning these efforts with overarching strategic goals. The SGPO may also need to spend time assessing existing SOA initiatives in relation to an IT department's readiness for SOA governance.

NOTE

Visit www.soaspecs.com for a list of industry maturity models relevant to the adoption of service-orientation and SOA.

Current Organizational Model

An organizational model defines roles and responsibilities within an organization. A given IT department will have a distinct organizational model that usually establishes a hierarchy with levels of authority. The SGPO must assess existing roles and responsibilities in order to identify how new roles and responsibilities specific to SOA governance will affect the organizational model.

Current and Planned Balance of On-Premise and Cloud-based IT Resources

In order to take an appropriate range of considerations into account when authoring SOA governance precepts and supporting processes, the SGPO needs to have a clear understanding of what cloud-based IT resources relevant to the SOA project currently exist, and to what extent the organization is planning to explore or proceed with cloud-based deployment of services and/or related IT resources. These considerations usually lead to additional standards, additional factors that apply to review processes, and additional organizational roles and skill-sets required for the definition of precepts and processes.

Step 2: Planning and Building the SOA Governance Program

After assessing the organization, the SGPO can get to work on actually planning and creating a concrete program for SOA governance. As previously established, the SOA governance program encompasses the SOA governance system and further provides supporting components to help establish and maintain this system.

To identify the primary components of an SOA governance program, we therefore begin by revisiting the precepts, people, and processes that are part of a governance system.

SOA Governance Precepts

The assessment completed in the previous stage is intended primarily to identify the aspects of a current or planned SOA initiative that pose the most risk and have the most urgent need for structured governance.

The following precepts are described individually in Chapters 7 to 12, where they are further associated with project lifecycle stages, processes, and organizational roles:

- Service Profile Standards (Chapter 7)
- SOA Governance Technology Standards (Chapter 7)
- Preferred Adoption Scope Definition (Chapter 7)
- Organizational Maturity Criteria Definition (Chapter 7)
- Standardized Funding Model (Chapter 7)
- Service Inventory Scope Definition (Chapter 8)
- Service and Capability Candidate Naming Standards (Chapter 8)
- Service Normalization (Chapter 8)
- Service Candidate Versioning Standards (Chapter 8)
- Schema Design Standards (Chapter 9)
- Service Contract Design Standards (Chapter 9)
- Service-Orientation Contract Design Standards (Chapter 9)
- SLA Template (Chapter 9)
- Service Logic Design Standards (Chapter 9)
- Service-Orientation Architecture Design Standards (Chapter 9)
- Service Logic Programming Standards (Chapter 9)
- Custom Development Technology Standards (Chapter 9)
- Testing Tool Standards (Chapter 10)
- Testing Parameter Standards (Chapter 10)
- Service Testing Standards (Chapter 10)
- Cloud Integration Testing Standards (Chapter 10)
- Test Data Usage Guidelines (Chapter 10)
- Production Deployment and Maintenance Standards (Chapter 10)
- Runtime Service Usage Thresholds (Chapter 11)

- Service Vitality Triggers (Chapter 11)
- Centralized Service Registry (Chapter 11)
- Service Versioning Strategy (Chapter 11)
- SLA Versioning Rules (Chapter 11)
- Service Retirement Notification (Chapter 11)
- Enterprise Business Dictionary/Domain Business Dictionary (Chapter 12)
- Service Metadata Standards (Chapter 12)
- Enterprise Ontology/Domain Ontology (Chapter 12)
- Business Policy Standards (Chapter 12)
- Operational Policy Standards (Chapter 12)
- Policy Centralization (Chapter 12)

It is important to document the reasoning behind each precept and define the circumstances in which it does or does not apply. Precepts need to be codified with clarifying policies and standards and consequences for non-compliance need to be further established. Also, supporting guidelines and compliance metrics are required. Where appropriate, conditions that might warrant a waiver need to be identified and a separate precept for allowing or denying waivers may further be required.

SOA Governance Processes

Depending on the size of the SGPO, internal processes may be required to coordinate activities within the group running the office. Governance process definition is another area of focus for the SOA governance program.

The following processes are covered in Chapters 7 to 12, where they are mapped to project lifecycle stages, precepts, and organizational roles:

- Organizational Governance Maturity Assessment (Chapter 7)
- Adoption Impact Analysis (Chapter 7)
- Adoption Risk Assessment (Chapter 7)
- Business Requirements Prioritization (Chapter 8)
- Service Candidate Review (Chapter 8)

- Service Contract Design Review (Chapter 9)
- Service Contract Registration (Chapter 9)
- Service Access Control (Chapter 9)
- Service Logic Design Review (Chapter 9)
- Legal Data Audit (Chapter 9)
- Service Logic Code Review (Chapter 9)
- Service Test Results Review (Chapter 10)
- Service Certification Review (Chapter 10)
- Service Maintenance Review (Chapter 10)
- Service Vitality Review (Chapter 11)
- Service Registry Access Control (Chapter 11)
- Service Registry Record Review (Chapter 11)
- Service Discovery (Chapter 11)
- Shared Service Usage Request (Chapter 11)
- Shared Service Modification Request (Chapter 11)
- Service Versioning (Chapter 11)
- Service Retirement (Chapter 11)
- Data Quality Review (Chapter 12)
- Communications Quality Review (Chapter 12)
- Information Alignment Audit (Chapter 12)
- Policy Conflict Audit (Chapter 12)

You may have noticed how several of these processes end with “review.” Many SOA governance processes are designed specifically to support and enforce compliance to precepts, and therefore are carried out subsequent to other project delivery tasks as a formal review.

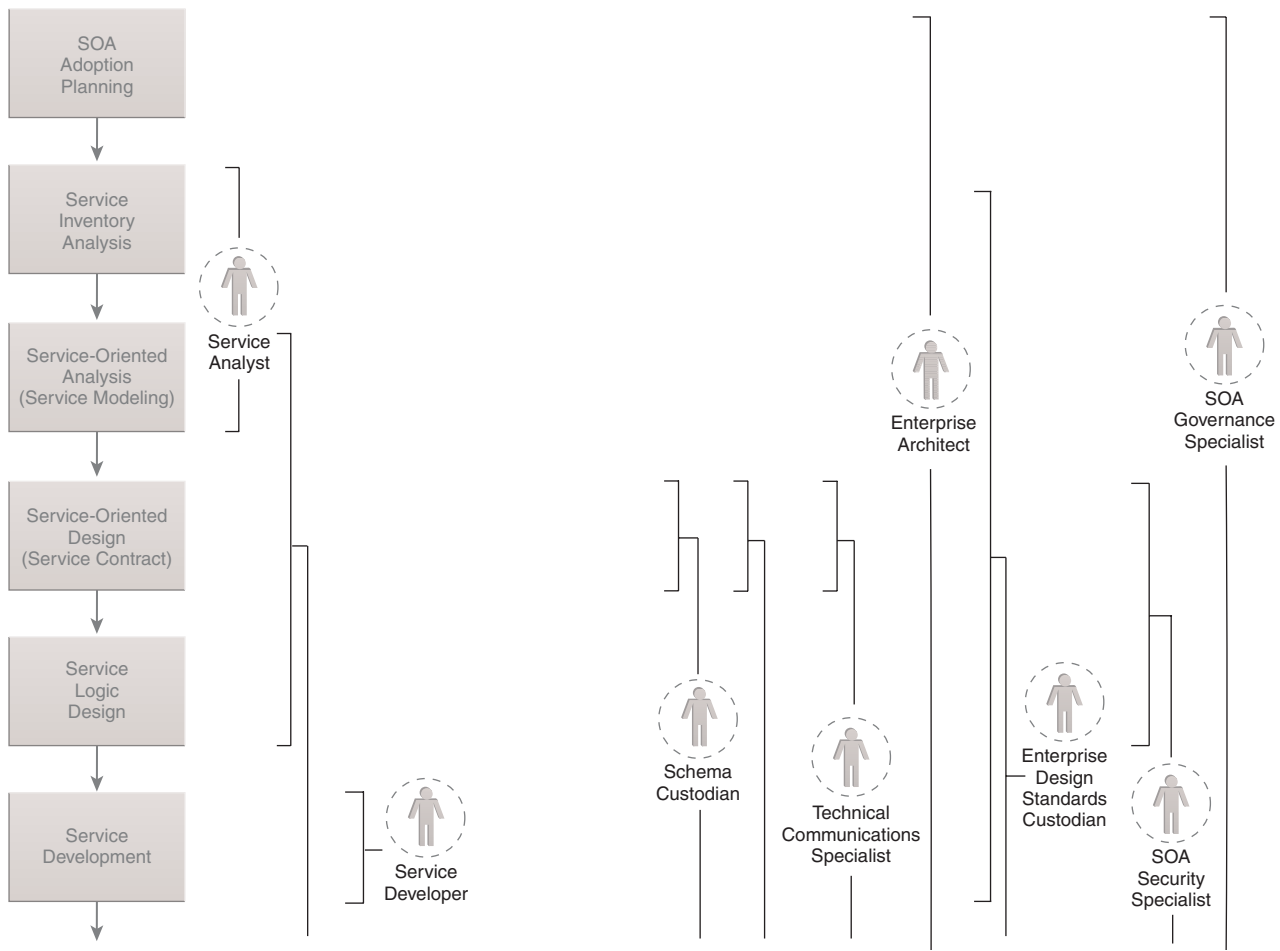
SOA Governance Roles

Organizational roles associated with SOA initiatives are of great interest to the SGPO because the various project stages for which governance precepts and processes can be defined will involve these roles in a governance capacity.

The following organizational roles were introduced in Chapter 5 and are further explored in Chapters 7 to 12, where they are associated with project lifecycle stages and SOA governance precepts and processes:

- Service Analyst
- Service Architect
- Service Developer
- Service Custodian
- Service Administrator
- Cloud Resource Administrator
- Schema Custodian
- Policy Custodian
- Service Registry Custodian
- Technical Communications Specialist
- Enterprise Architect
- Enterprise Design Standards Custodian (and Auditor)
- SOA Quality Assurance Specialist
- SOA Security Specialist
- SOA Governance Specialist

Figure 6.8 provides an overview of how these roles commonly map to SOA project lifecycle stages.



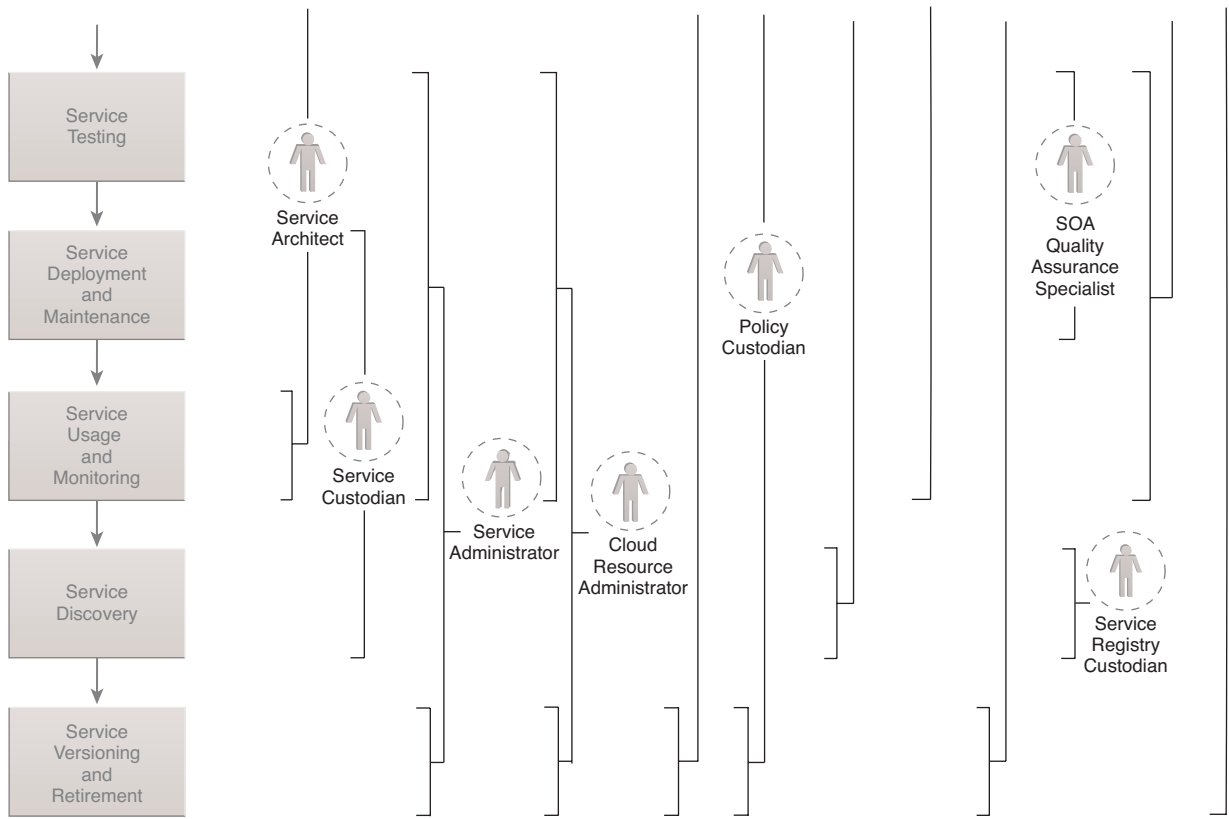


Figure 6.8

Each role can be involved in governance activities pertaining to multiple SOA project stages. Appendix B further provides master reference diagrams that illustrate the cross-project stage relationships of these roles with precepts and processes.

Additional Components

As previously stated, the scope of the SOA governance program goes beyond the definition of the SOA governance system. Some of the areas that the program will likely need to further address in support of pre-defined precepts and processes include:

- *SOA Governance Tools* – Products and technologies that enable the automation of SOA governance processes or that can monitor and collect relevant statistical data need to be identified and chosen in order to establish a suitable SOA governance infrastructure.
- *SOA Governance Roadmap* – Also referred to as the SOA Governance Program Project Plan, this document establishes the timeline, resources, budget, and other real-world considerations required to actually realize the goals of the SGPO and, more specifically, a specific SOA governance program.

There can be many more parts and extensions to an SOA governance program specific to the needs of a given IT department and its SOA project goals.

Step 3: Running the SOA Governance Program (Best Practices and Common Pitfalls)

The SOA governance program is a living entity that requires continuous maintenance. Over time, and in response to real-world issues and challenges, the SOA governance program will naturally evolve as precepts, roles, and processes are refined or added to the overall SOA governance system.

This section contains a series of best practices that provide guidance for successfully running an SOA governance program, as well as a set of common pitfalls that warn against factors and circumstances that can inhibit the adoption and evolution of the program.

Collect the Right Metrics and Have the Right People Use Them

Metrics, the fourth primary building block of a governance system, represent a vital element in the on-going operation of the SOA governance program. Having the tools and processes to consistently collect and disseminate key metrics is just as important as having the right individuals and groups assigned the responsibility to interpret and make decisions based on the reported metrics.

Provide Transparency and Foster Collaboration

Depending on its scope, an SOA governance program can affect a wide range of departments, groups, and individuals. Instead of creating the program in isolation, its development should be an open process, accessible for review and involvement to others within the IT department. Not only will this generate goodwill among those less enthusiastic about upcoming SOA adoption initiatives, but it will also allow people to voice concerns and provide suggestions. This type of feedback can help improve the SOA governance system, while also easing its eventual implementation.

Ensure Consistency and Reliability

SOA governance precepts need to be consistently enforced and SOA governance processes need to be consistently carried out. Providing a reliable means of managing and maintaining the SOA governance system is the foremost responsibility of the SGPO and depends heavily on the quality and detail with which the SOA governance program has been developed.

Besides human incompetence and poor SOA governance program definition, another reason this best practice may not be followed is an unexpected withdrawal of funding allocated to the SGPO. Should this occur, it is preferable to downsize the scope of the SOA governance program instead of trying to continue carrying out SOA governance activities without the necessary resources to ensure consistency and reliability.

Compliance and Incentives

An SOA governance system will introduce precepts that will sometimes restrict certain tasks that IT project team members have traditionally been free to complete by using their own judgment. At the same time, precepts also help make critical decisions for IT professionals that can ease their responsibilities while also guaranteeing consistency across services and service-oriented solutions. It is important that project teams embrace SOA governance precepts and processes and that they clearly understand how and why new types of compliance are required, while also fully acknowledging that their judgment and freedom in other areas are still required and relied upon.

Furthermore, offering formal incentives for regularly supporting precepts can go a long way to fostering consistent adherence. Because people will generally do that for which they are most rewarded, an absence of incentives can encourage them to violate or ignore SOA governance precepts. When this happens, something generally needs to change: the incentive, the precept, or the people.

Education and Communication

SOA governance systems can impose precepts more restrictive than traditional IT governance systems. Furthermore, some organizations can find it difficult to fully mandate the adoption of and compliance to SOA governance precepts. Even when compliance is required, in some IT cultures, groups or individuals can still choose to “rebel” by intentionally disregarding precepts because they are considered too burdensome.

Regardless of whether compliance to SOA governance precepts is voluntary or mandatory, it is critical that everyone affected fully understand why these precepts exist and how their compliance ultimately results in tangible benefits. Furthermore, it can be helpful to specifically address the common question: “What’s in it for me?” Fostering a true understanding of how support for the SOA governance system can result in personal benefit will further help unify IT project teams and personnel.

For this purpose, the SGPO must put together an education and communications program. This program must begin by establishing SOA terminology, concepts, and practices using a common vocabulary that all project team members can understand. It must then introduce the SOA governance system and impress its virtues.

Common Pitfalls

From the many failed and successful SOA adoption initiatives has emerged a set of common pitfalls that pertain directly to establishing and running an SOA governance program:

- *Lack of Recognized Authority* – The SGPO must be endowed with the responsibility and authority to develop and execute the SOA governance program. For this to happen, other IT departments and project teams must accept that authority. When the SGPO’s authority is ignored or not recognized, there needs to be recourse. If the lack of recognition persists, there need to be consequences for those who refuse to provide support.
- *Misalignment with IT Governance* – An SOA governance system must be consistent with and supportive of existing corporate and IT governance systems. If other IT governance precepts and processes are not taken into consideration, the SOA governance system can become inadvertently misaligned. This will result in conflicts and can further introduce risks to the IT department as a whole.
- *Overestimating or Underestimating Cloud Computing Factors* – There are various ways that cloud platforms and technologies can be made part of the planned SOA project. An organization may have or may plan to establish a private cloud comprised

of standardized IT resources that require distinct administration processes, or it may be moving IT resources to a public cloud that imposes non-compliant requirements that may require even more distinct administration approaches. Either way, it is important for the SGPO to be open and flexible regarding these possibilities and—if cloud deployment is a possibility—to fully understand the consequences of having some or all services or IT resources of a given project deployed in cloud environments.

- *Impractical or Overly Formal Processes* – SOA governance processes are intended to help enforce and organize the application of precepts. Sometimes it can be tempting to create highly structured and detailed processes that cover all possible bases. Although such processes may be thorough, they can be too burdensome, onerous, or time consuming to carry out consistently in the real world. When designing SOA governance processes, consider the impact of the process on the project lifecycle and timeline and investigate any opportunity to streamline and automate parts of the process. Tools that integrate the governance process directly with development or administration platforms may further be helpful in allowing developers and administrators to efficiently identify and fix compliance issues.
- *Poor Documentation* – SOA governance precepts should be well-documented and disseminated. Many precepts require human interpretation, which means that people in the trenches will need to clearly understand how and when to apply them. Sometimes members of the SGPO take the formality of an SOA governance system too seriously. As a result, precepts and processes can be documented using overly academic or technical language. This can make the documents difficult to fully understand and, at times, inaccessible to some project team members.
- *Overspending on SOA Governance Tools* – SOA vendors have developed highly sophisticated administration and management tools (commonly labeled as “governance” products) with various design and runtime features. While powerful, these tools sometimes provide functionality that is not needed or not suitable for an organization’s specific governance requirements. Further, these tools can be very expensive, especially in larger IT enterprises. Therefore, it is often worth waiting to invest in a full-blown SOA governance infrastructure until an SOA governance program has matured to the extent that the actual design and runtime automation requirements can be identified and well defined. Otherwise, over-spending or mis-spending on governance tools and technology can put a significant dent in an SOA initiative’s overall ROI and further limit funds that may have been better allocated to supporting the SGPO in other areas.

SUMMARY OF KEY POINTS

- An SOA governance program encompasses the models that comprise an SOA governance system and further provides actionable artifacts that determine how the system will be established and maintained.
 - A basic framework for an SOA governance program consists of three primary parts that address the assessment of the current organizational state, the planning and building of the program, as well as its evolutionary operation.
-

Index

A

acquisition strategies for governance technology, 444-447
active governance tasks, 429
activities in RUP, 620. *See also* vitality activities
administrative governance technology, 429
Adoption Impact Analysis process, 176-178
adoption planning
 case study, 182-186
 people for, 179-182
 precepts for, 169-173
 processes for, 173-178
Adoption Risk Assessment process, 178
Agnostic Capability design pattern, 491
Agnostic Context design pattern, 492
agnostic logic, defined, 39-40
Agnostic Sub-Controller design pattern, 493
analysis stages in MSOAM, mapping to RUP disciplines, 626-627. *See also* Service Inventory Analysis lifecycle; Service-Oriented Analysis stage
Annotated SOA Manifesto, 34, 53, 578-590

approve activity (vitality activities), 423
artifacts in RUP, 620
assess activity (vitality activities), 422
assessing governance technology, 448-449
Asynchronous Queuing design pattern, 494
Atomic Service Transaction design pattern, 495
attachments (SOAP), 483
attacks, types of, 162
authentication. *See* Brokered Authentication design pattern; Direct Authentication design pattern
automated governance tasks, 427

B

backwards compatibility, 596-599
balanced scope, 51, 53-55. *See also* pillars of service-orientation
best practices
 assessing governance technology, 448-449
 in SOA governance program implementation, 146-150
books related to this book, 5-6
breadth roles in RUP, 623-624

Brokered Authentication design pattern,
 162, 496
Business Aligned maturity level, 58
Business Analyst role, 113, 397-399
business changes versus technology
 changes in vitality triggers, 415-416
business dictionaries, 375-376
Business Driven maturity level, 58
business heat map, 195
business information, assigning
 value to, 409
Business Policy Standards precept,
 382-384
Business Requirements Prioritization
 process, 195-197
business shifts, as vitality triggers, 417

C

CA (certificate authority), 161
Canonical Data Model design
 pattern, 224
Canonical Expression design pattern,
 206, 225, 497
Canonical Protocol design pattern,
 225, 498
Canonical Resources design pattern, 499
Canonical Schema Bus compound
 pattern, 501
Canonical Schema design pattern,
 224, 500
Canonical Versioning design pattern,
 610, 502
Canonical XML, 161
Capability Composition design pattern,
 207, 503
capability granularity, defined, 45
capability profile structure, 118-119
Capability Recomposition design
 pattern, 504
capitalization usage, 14

case studies (Raysmoore Corporation)
 background, 18-20
 conclusion, 454-455
 precepts (governance controls),
 167-168
 Service Deployment and
 Maintenance, 312-313
 Service Development, 276
 Service Discovery, 350-351
 Service Inventory Analysis, 201-205
 Service Logic Design, 265-266
 Service-Oriented Analysis, 217-220
 Service Testing, 294-297
 Service Usage and Monitoring,
 333-334
 SOA adoption planning, 182-186
central funding model
 in platform funding, 61, 64-66
 in service funding, 69, 71-72
centralized domain SOA Governance
 Program Office, 134
centralized enterprise SOA Governance
 Program Office, 133
Centralized Service Registry precept,
 335-337
certificate authority (CA), 161
Certified Cloud Computing, 15-16
cloud, defined, 36
cloud-based security groups, 161
Cloud Burst Threshold, 318
cloud computing, defined, 35
Cloud Computing Governance
 Specialist role, 114
Cloud Computing Security Specialist
 role, 114
cloud consumers, defined, 38
cloud delivery models, list of, 38
cloud deployment models, list of, 37-38
cloud governance tasks, 428
Cloud Integration Testing Standards
 precept, 283-284

- cloud providers, defined, 38**
- Cloud Resource Administrator role, 100-102**
 - reference diagram, 462
 - for Service Deployment and Maintenance, 305-306
 - in Service Testing, 288
 - in Service Usage and Monitoring, 328
 - in Service Versioning and Retirement, 363
- Cloud Service Owner role, 98-99**
- cloud services, defined, 36**
- cloud vendor leasing acquisition strategy for governance technology, 447**
- coarse-grained constraints, 595**
- code examples**
 - backwards compatibility
 - for methods, 598*
 - for REST services, 597*
 - for Web services, 596-597*
 - in XML Schemas, 598-599*
 - complexType construct containing fine and coarse-grained constraints, 595
 - default value of minOccurs attribute, 602
 - forwards compatibility with wildcards, 601
 - incrementing minOccurs attribute value, 604
- communicate activity (vitality activities), 423**
- Communications Quality Review process, 391**
- community cloud deployment model, 37**
- compatibility, versioning and, 596-605**
 - backwards compatibility, 596-599
 - compatible changes, 602-604
 - forwards compatibility, 599-602
 - incompatible changes, 604-605
 - REST services compatibility, 605-608
- compatibility guarantee, 609**
- Compatible Change design pattern, 353, 505**
- compatible changes, 602-604**
- Compensating Service Transaction design pattern, 506**
- complexity of versioning strategies, 615**
- compliance metrics, 166, 419**
- components, services as, 32**
- Composition Autonomy design pattern, 507**
- composition controller capabilities, design characteristics, 487**
- composition member capabilities, design characteristics, 486**
- compound patterns**
 - Canonical Schema Bus, 501
 - Enterprise Service Bus, 523
 - Federated Endpoint Layer, 527
 - Official Endpoint, 539
 - Orchestration, 540
 - Service Broker, 553
 - Three-Layer Inventory, 570
- computing professionals, Certified Cloud Computing, 15-16**
- Concurrent Contracts design pattern, 225, 508**
- configuration management tools, 443**
- constraint granularity, 45, 595**
- content sharing and publishing tools, 442-443**
- Contract Centralization design pattern, 225, 509**
- Contract Denormalization design pattern, 510**
- “contract first” approach, 476**
- cost metrics, 164**

cost of capital, 164

Cross-Domain Utility Layer design pattern, 511

Custom Development Technology Standards precept, 268-270

custom SOA governance solutions, 443-444

D

data, defined, 372

Data Architect role, 113, 399

Data Confidentiality design pattern, 162, 512

Data Format Transformation design pattern, 513

data granularity, defined, 45

Data Model Transformation design pattern, 514

Data Origin Authentication design pattern, 162, 515

Data Quality Review process, 389-391

deactivation. *See* Service Versioning and Retirement stage

Decomposed Capability design pattern, 516

Decoupled Contract design pattern, 225, 517

Decryption Transform for XML Signature, 161

Define Enterprise Business Models in Service Inventory Analysis lifecycle, 408-409, 626

Define Technology Architecture process, 626

Definition of the Service Inventory Blueprint process, 627

delivery models, 38

deployment. *See* cloud deployment models; Service Deployment and Maintenance stage

depth roles in RUP, 623-624

design patterns, 13-14

Agnostic Capability, 491

Agnostic Context, 492

Agnostic Sub-Controller, 493

Asynchronous Queuing, 494

Atomic Service Transaction, 495

Brokered Authentication, 162, 496

Canonical Data Model, 224

Canonical Expression, 206, 225, 497

Canonical Protocol, 225, 498

Canonical Resources, 499

Canonical Schema, 224, 500

Canonical Versioning, 502, 610

Capability Composition, 207, 503

Capability Recomposition, 504

Compatible Change, 353, 505

Compensating Service Transaction, 506

Composition Autonomy, 507

Concurrent Contracts, 508

Contract Centralization, 225, 509

Contract Denormalization, 510

Cross-Domain Utility Layer, 511

Data Confidentiality, 162, 512

Data Format Transformation, 513

Data Model Transformation, 514

Data Origin Authentication, 162, 515

Decomposed Capability, 516

Decoupled Contract, 225, 517

defined, 46-47

Direct Authentication, 162, 518

Distributed Capability, 519

Domain Inventory, 54, 193, 520

Dual Protocols, 225, 521

Enterprise Inventory, 193, 522

Entity Abstraction, 54, 524

Event-Driven Messaging, 525

Exception Shielding, 162, 526

File Gateway, 528

Functional Decomposition, 529

- Intermediate Routing, 530
- Inventory Endpoint, 531
- Legacy Wrapper, 249, 532
- Logic Centralization, 533
- Message Screening, 162, 534
- Messaging Metadata, 378, 535
- Metadata Centralization, 234, 536
- Multi-Channel Endpoint, 537
- Non-Agnostic Context, 538
- Partial State Deferral, 541
- Partial Validation, 542
- Policy Centralization, 388, 543
- Process Abstraction, 54, 544
- Process Centralization, 545
- Protocol Bridging, 546
- Proxy Capability, 353, 547
- Redundant Implementation, 548
- Reliable Messaging, 549
- Rules Centralization, 550
- Schema Centralization, , 224, 551
- Service Agent, 552
- Service Callback, 554
- Service Data Replication, 555
- Service Decomposition, 353, 556
- Service Encapsulation, 557
- Service Façade, 249, 558
- Service Grid, 559
- Service Instance Routing, 560
- Service Layers, 54, 561
- Service Messaging, 562
- Service Normalization, 207, 344, 563
- Service Perimeter Guard, 162, 564
- Service Refactoring, 353, 565
- State Messaging, 566
- State Repository, 567
- Stateful Services, 568
- Termination Notification, 356, 569
- Trusted Subsystem, 162, 571
- UI Mediator, 572
- Utility Abstraction, 54, 573
- Validation Abstraction, , 225, 574
- Version Identification, 353, 575, 608
- design principles, 13-14**
 - list of, 27
 - Service Abstraction, 228, 225, 374, 478
 - Service Autonomy, 481
 - Service Composability, 88, 486-487
 - Service Discoverability, 91, 225, 228, 234, 239, 335, 391, 484-485
 - Service Loose Coupling, 225, 226, 228, 477
 - Service Reusability, 479-480
 - Service Statelessness, 482-483
 - Standardized Service Contract, 87, 225, 228, 237, 475-476
- design stages in MSOAM, mapping to RUP disciplines, 626-627**
- design-time governance tasks, 428**
- digital certificates, 161**
- digital signatures, 160**
- Direct Authentication design pattern, 162, 518**
- disciplines, 51-52. See also pillars of service-orientation**
 - in RUP, 620
 - mapping to MSOAM analysis and design stages, 626-627*
 - mapping to MSOAM service delivery project stages, 625-626*
- Distributed Capability design pattern, 519**
- Domain Business Dictionary precept, 375-376**
- Domain Inventory design pattern, 54, 193, 520**
- Domain Ontology precept, 380-382**
- domain service inventory, defined, 41**
- domains, assessing, 137-139**
- Dual Protocols design pattern, 225, 521**

E

education, 51-52. *See also* pillars of service-orientation

Educator, 112

embedded policy logic, 374

encryption, 160

enforcement governance technology, 430

enterprise, assessing, 137-139

Enterprise Architect role, 106

- reference diagram, 467
- in Service Deployment and Maintenance, 308
- in Service Development, 274
- in Service Inventory Analysis, 199
- in Service Logic Design, 261
- in Service-Oriented Analysis, 215
- in Service-Oriented Design, 242
- in Service Testing, 289
- in Service Usage and Monitoring, 325-326
- in SOA adoption planning, 179-180

Enterprise Business Dictionary precept, 375-376

enterprise business models, establishing, 408-409

Enterprise Design Standards Custodian role, 107-108

- reference diagram, 468
- in Service Development, 273-274
- in Service Inventory Analysis, 198-199
- in Service Logic Design, 260
- in Service-Oriented Analysis, 214
- in Service-Oriented Design, 241-242
- in Service Versioning and Retirement, 360-361

Enterprise Inventory design pattern, 193, 522

Enterprise Ontology precept, 380-382

enterprise roles in MSOAM, 624-626

Enterprise Service Bus compound pattern, 523

Enterprise Unified Process (EUP), 628

Entity Abstraction design pattern, 54, 524

entity services, defined, 39

entry fees in usage funding model, 66

EUP (Enterprise Unified Process), 628

Event-Driven Messaging design pattern, 525

Exception Shielding design pattern, 162, 526

F

federated domain SOA Governance Program Offices, 135

Federated Endpoint Layer compound pattern, 527

fees in usage funding model, 66

File Gateway design pattern, 528

fine-grained constraints, 595

flexible versioning strategy, 611-613

forwards compatibility, 599-602

Functional Decomposition design pattern, 529

functional metadata, 378

functional tests, 278

funding models, 60-77

- platform funding, 60-69
- service funding, 69-74
- Standardized Funding Model precept, 172-173

G

gaps in RUP and service-orientation, 628

glossary Web site, 5, 15-16, 632

governance

- defined, 122-123
- management and, 124-126

- MDM (master data management)
 - and, 409
 - methodology and, 124-126
 - scope of, 123-126
 - selecting style of, 126-127
 - SOA and, 130
 - SOA Governance Program Office (SGPO), 131-136
 - vitality. *See* vitality
 - governance controls, 127-129**
 - metrics, 129, 146, 164-165
 - people. *See* organizational roles
 - precepts. *See* precepts (governance controls)
 - processes. *See* processes (governance controls)
 - governance impact of versioning strategies, 614**
 - governance roles in MSOAM, 624-626**
 - governance systems, defined, 426-427**
 - governance task types, 427-429**
 - governance technology**
 - acquisition strategies, 444-447
 - assessing, 448-449
 - categories of, 429-431
 - product types
 - configuration management tools, 443*
 - content sharing and publishing tools, 442-443*
 - custom SOA governance solutions, 443-444*
 - policy systems, 437-439*
 - quality assurance tools, 439-441*
 - repositories, 433-435*
 - service agents, 435-437*
 - service registries, 431-433*
 - SOA management suites, 441-442*
 - technical editors and graphic tools, 442*
 - granularity, service-related granularity, defined, 44-45
 - guidelines, defined, 128
- ## H
- hardened virtual server images, 161
 - hardware accelerators, 483
 - hashing, 160
 - human-readable policies, 373
 - hybrid cloud deployment model, 38
 - hybrid funding model in service funding, 69, 72-74
- ## I
- IaaS (Infrastructure-as-a-Service)
 - delivery model, 38
 - identify activity (vitality activities), 421
 - identity and access management (IAM), 160
 - implementation requirements, service contracts, 475
 - incompatible changes, 604-605
 - independent domain SOA Governance Program Offices, 136
 - industry shifts, as vitality triggers, 417-418
 - information
 - business information, assigning value to, 409
 - defined, 372
 - Information Alignment Audit process, 393-395
 - Infrastructure-as-a-Service (IaaS)
 - delivery model, 38
 - integration costs, 164
 - integration tests, 279
 - Intermediate Routing design
 - pattern, 530
 - Inventory Endpoint design pattern, 531

IT Manager role, 115
 IT resources, defined, 35-36
 IT roles, 112-115

J—K—L

jurisdiction models in SOA Governance
 Program Office (SGPO), 133-136
 knowledge, defined, 372
 leasing from cloud vendor
 acquisition strategy for
 governance technology, 447
 Legacy Wrapper design pattern, 249, 532
 Legal Data Audit process, 257-258
 lifecycle stages. *See* service project
 lifecycle stages
 locked-in costs, 164
 Logic Centralization design pattern, 533
 logical domain precepts, 159
 loose versioning strategy, 611, 613-614

M

Mainstream SOA Methodology.
 See MSOAM (Mainstream SOA
 Methodology)
 maintenance, 298
 management
 governance and, 124-126
 methodology and, 125-126
 manual governance tasks, 427
 mapping diagrams, 12
 master data management (MDM),
 governance and, 409
 maturity levels in SOA planning, 56-59
 message-layer security, 160
 Message Screening design pattern,
 162, 534
 Messaging Metadata design pattern,
 378, 535

metadata, 377-380
 Metadata Centralization design pattern,
 234, 536
 methodology
 governance and, 124-126
 management and, 125-126
 metrics (governance controls), 127,
 129, 146
 cost metrics, 164
 standards-related precept
 metrics, 165
 threshold metrics, 165
 as vitality triggers, 418-419
 milestone triggers, 420
 monitoring governance technology, 429
 MSOAM (Mainstream SOA
 Methodology)
 analysis and design stages, mapping
 to RUP disciplines, 626-627
 roles in
 enterprise and governance roles,
 624-626
 mapping to RUP roles, 623-624
 service delivery project stages,
 mapping to RUP disciplines,
 625-626
 Multi-Channel Endpoint design
 pattern, 537
 multiple vendor acquisition strategy for
 governance technology, 445-446

N

naming standards, Service and
 Capability Candidate Naming
 Standards precept, 206
 Non-Agnostic Context design
 pattern, 538
 non-agnostic logic, defined, 39-40
 notification service for this book series,
 16, 632

O

objectives, defined, 128

Official Endpoint compound

pattern, 539

on-going costs, 164

on-premise, defined, 37

on-premise governance tasks, 428

ontologies, 380-382

open source acquisition strategy for

governance technology, 446-447

Operational Policy Standards precept, 384-386

Orchestration compound pattern, 540

Organizational Governance Maturity

Assessment process, 173-175

organizational maturity, levels of, 56-59

Organizational Maturity Criteria

Definition precept, 171

organizational roles, 92-115,

127-128, 156

Business Analysts, 397-399

Cloud Resource Administrator,
100-102, 462

Cloud Service Owner, 98-99

Data Architects, 399

Educator, 112

Enterprise Architect, 106, 467

Enterprise Design Standards

Custodian, 107-108, 468

IT roles, 112-115

planning and building SOA

governance programs, 143

Policy Custodian, 104, 401, 464

Schema Custodian, 102-103,
399-400, 463

Service Administrator, 100, 461

Service Analyst, 96, 458

Service Architect, 96, 459

Service Custodian, 98, 460

for Service Deployment and
Maintenance, 304-311

Service Developer, 97, 460

for Service Development, 272-275

for Service Discovery, 345-348

for Service Inventory Analysis,
197-200

for Service Logic Design, 259-264

for Service-Oriented Analysis,
212-217

for Service-Oriented Design,
236-246

Service Registry Custodian, 105,
402-403, 465

for Service Testing, 287-293

for Service Usage and Monitoring,
325-332

for Service Versioning and
Retirement, 360-366

for SOA adoption planning, 179-182

SOA Governance Specialist, 111,
406-407, 471-472

SOA Quality Assurance Specialist,
109, 405-406, 469

SOA Security Specialist, 110, 470

Technical Communications
Specialist, 105, 403, 466

organizational shifts as vitality triggers, 419-420

P

PaaS (Platform-as-a-Service) delivery model, 38

Partial State Deferral design pattern, 541

Partial Validation design pattern, 542

passive governance tasks, 428

patterns. See design patterns

people (governance controls). See organizational roles

performance, state management and, 483

performance metrics, 166, 419

performance tests, 279

- periodic vitality triggers, 420**
- per use fees in usage funding model, 66**
- pillars of service-orientation, 51-55**
 - balanced scope, 53-55
 - discipline, 52
 - education, 52
 - mapping to RUP principles, 620-622
 - teamwork, 52
- PKI (Public Key Infrastructure), 161**
- planning. *See also* SOA planning**
 - SOA adoptions
 - case study, 182-186*
 - people for, 179-182*
 - precepts for, 169-173*
 - processes for, 173-178*
 - SOA governance programs, 139-146
- Platform-as-a-Service (PaaS) delivery model, 38**
- platform funding models, 60-69**
 - central funding model, 64-66
 - project funding model, 61
 - usage funding model, 66-69
- policies**
 - defined, 128
 - explained, 373-374
 - WS-Policy assertions, 355
- Policy Centralization design pattern, 386-388, 543**
- Policy Conflict Audit process, 395-397**
- Policy Custodian role, 104, 401**
 - reference diagram, 464
 - in Service Deployment and Maintenance, 311
 - in Service-Oriented Design, 238
 - in Service Versioning and Retirement, 364
- policy systems, 437-439**
- policy tests, 278**
- precepts (governance controls), 127-128, 156**
 - Business Policy Standards, 382-384
 - case study, 167-168
 - Enterprise Business Dictionary/
Domain Business Dictionary,
375-376
 - Enterprise Ontology/Domain
Ontology, 380-382
 - logical domain precepts, 159
 - Operational Policy Standards,
384-386
 - planning and building SOA
 - governance programs, 139-141
 - Policy Centralization, 386-388
 - security control precepts, 160-163
 - for Service Deployment and
Maintenance, 298-300
 - for Service Development, 267-270
 - for Service Discovery, 335-337
 - service information precepts, 158
 - for Service Inventory Analysis,
193-195
 - for Service Logic Design, 249-253
 - Service Metadata Standards,
377-380
 - for Service-Oriented Analysis,
206-210
 - for Service-Oriented Design,
223-231
 - service policy precepts, 158
 - service profile standards, 157
 - for Service Testing, 279-286
 - for Service Usage and Monitoring,
317-322
 - for Service Versioning and
Retirement, 352-356
 - for SOA adoption planning, 169-173
 - SOA governance technology
standards, 163

Preferred Adoption Scope Definition precept, 169-170

Prentice Hall Service-Oriented Computing Series from Thomas Erl, 632

principle profiles

- Service Abstraction, 478
- Service Autonomy, 481
- Service Composability, 486-487
- Service Discoverability, 484-485
- Service Loose Coupling, 477
- Service Reusability, 479-480
- Service Statelessness, 482-483
- Standardized Service Contract, 475-476

principles of RUP (Rational Unified Processing), mapping to pillars of service-orientation, 620-622

private cloud deployment model, 38

private service registries, 432

Process Abstraction design pattern, 54, 544

Process Centralization design pattern, 545

processes (governance controls), 127, 129, 156

- Communications Quality Review, 391
- Data Quality Review, 389-391
- Information Alignment Audit, 393-395
- planning and building SOA governance programs, 141-142
- Policy Conflict Audit, 395-397
- for Service Deployment and Maintenance, 301-304
- for Service Discovery, 337-344
- for Service Inventory Analysis, 195-197
- for Service Logic Design, 253-258
- for Service-Oriented Analysis, 210-211

for Service-Oriented Design, 231-235

for Service Testing, 286

for Service Usage and Monitoring, 323-324

for Service Versioning and Retirement, 357-360

for SOA adoption planning, 173-178

Production Deployment and Maintenance Standards precept, 298-300

profiles. See service profiles

programming logic metadata, 378

project funding model

in platform funding, 61

in service funding, 69-70

project lifecycle stages. See service project lifecycle stages

Protocol Bridging design pattern, 546

Proxy Capability design pattern, 353, 547

public cloud deployment model, 37

Public Key Infrastructure (PKI), 161

Q—R

quality assurance, SOA Quality Assurance Specialist role, 109

quality assurance tools, 439-441

quality of service metadata, 378

Rational Unified Process. See RUP (Rational Unified Process)

Raysmoore Corporation case study. See case studies (Raysmoore Corporation)

recommended reading, 5-6, 14-16, 47-48, 628

Redundant Implementation design pattern, 548

refresh activity (vitality activities), 422-423

regression tests, 278
 Reliable Messaging design pattern, 549
 reporting governance technology, 430
 repositories, 433-435
 resources, 35-36
 responsibilities. *See* organizational roles
 REST services

- compatibility considerations, 605-608
- defined, 34
- versioning, 594-595
 - backwards compatibility*, 597-599
 - forwards compatibility*, 600
 - strategy considerations*, 615-616

 retirement. *See* Service Versioning and Retirement stage
 RFPs (requests for proposal), creating, 449
 roles. *See also* organizational roles

- in MSOAM, enterprise and governance roles, 624-626
- in RUP, 619, 623-624

 Rules Centralization design pattern, 550
 runtime governance tasks, 428
 Runtime Service Usage Thresholds precept, 317-319
 RUP (Rational Unified Process), 618

- breadth and depth roles, 623-624
- compatibility with SOA, 618-619, 628
- content elements of, 619-620
- disciplines in
 - mapping to MSOAM analysis and design stages*, 626-627
 - mapping to MSOAM service delivery project stages*, 625-626
- principles of, mapping to pillars of service-orientation, 620-622

S

SaaS (Software-as-a-Service) delivery model, 38
 SAML (Security Assertion Markup Language), 161
 scalability, 482
 Schema Centralization design pattern, 224, 551
 Schema Custodian role, 102-103, 399-400

- reference diagram, 463
- in Service Deployment and Maintenance, 311
- in Service-Oriented Design, 237-238
- in Service Versioning and Retirement, 364

 Schema Design Standards precept, 223-225
 scope

- of governance, 123-126
- Service Inventory Scope Definition precept, 193-195

 Security Assertion Markup Language (SAML), 161
 security attacks, types of, 162
 security control precepts, 160-163
 security policies, 160
 security sessions, 160
 security tests, 278
 security token actions, 160
 selecting style of governance, 126-127
 Service Abstraction design principle, 27, 225, 228, 374, 478
 Service Access Control process, 253
 Service Administrator role, 100

- reference diagram, 461
- in Service Deployment and Maintenance, 304-305
- in Service Testing, 287

- in Service Usage and Monitoring, 327-328
- in Service Versioning and Retirement, 362
- Service Agent design pattern, 552**
- service agents, 435-437**
- Service Aggressive maturity level, 59**
- Service Analyst role, 96**
 - reference diagram, 458
 - in Service Inventory Analysis, 197
 - in Service-Oriented Analysis, 212-213
- Service and Capability Candidate Naming Standards precept, 206**
- Service Architect role, 96**
 - reference diagram, 459
 - in Service Logic Design, 259-260
 - in Service-Oriented Analysis, 213
 - in Service-Oriented Design, 236-237
 - in Service Usage and Monitoring, 326-327
- Service Autonomy design principle, 27, 481**
- Service Aware maturity level, 57**
- Service Billing Threshold, 318**
- Service Broker compound pattern, 553**
- Service Callback design pattern, 554**
- Service Candidate Review process, 210-211**
- Service Candidate Versioning Standards precept, 209**
- service candidates, defined, 42**
- Service Capable maturity level, 57**
- service catalogs, service profiles and, 119. See also service portfolio**
- Service Certification Review process, 301-302**
- Service Composability design principle, 27, 88, 486-487**
- Service Composition Membership Threshold, 317**
- service compositions, defined, 40-41**
- Service Contract Design Review process, 231-232**
- Service Contract Design Standards precept, 225-227**
- Service Contract Registration process, 234-235**
- service contracts, 486. See also Service-Oriented Design stage**
 - defined, 43-44
 - versioning. *See* versioning
- Service Custodian role, 98**
 - reference diagram, 460
 - in Service Deployment and Maintenance, 307
 - in Service Discovery, 345-346
 - in Service Usage and Monitoring, 329
- Service Data Replication design pattern, 555**
- Service Data Throughput Threshold, 318**
- Service Decomposition design pattern, 353, 556**
- service delivery project stages**
 - in MSOAM, mapping to RUP disciplines, 625-626
- Service Deployment and Maintenance stage, 298**
 - case study, 312-313
 - people for, 304-311
 - precepts for, 298-300
 - processes for, 301-304
 - in service project lifecycle stages, 89
- Service Developer role, 97**
 - reference diagram, 460
 - in Service Development, 272

- Service Development stage**
 - case study, 276
 - people for, 272-275
 - precepts for, 267-270
 - in service project lifecycle stages, 87
- Service Discoverability design principle,** 27, 91, 225, 228, 234, 239, 335, 391, 484-485
- Service Discovery stage, 340-341**
 - case study, 350-351
 - people for, 345-348
 - precepts for, 335-337
 - processes for, 337-344
 - in service project lifecycle stages, 90-91
- Service Elasticity Threshold, 318**
- Service Encapsulation design pattern, 557**
- Service Exception Threshold, 318**
- Service Façade design pattern, 249, 558**
- service funding, 60, 69-74**
 - central funding model, 71-72
 - hybrid funding model, 72-74
 - project funding model, 69-70
 - usage funding model, 74
- service granularity, defined, 44**
- Service Grid design pattern, 559**
- Service Ineffectual maturity level, 58**
- Service Information Governance Council, establishing, 408**
- service information precepts, 158**
- Service Instance Routing design pattern, 560**
- Service Instance Threshold, 317**
- service inventory, defined, 41**
- Service Inventory Analysis lifecycle, 83, 626**
 - case study, 201-205
 - Define Enterprise Business Models step, 408-409
 - iterative cycles in, 192
 - people in, 197-200
 - precepts for, 193-195
 - processes for, 195-197
 - in service project lifecycle stages, 82-83
 - time allotted to, 189-190
- service inventory blueprints, defined, 41**
- service inventory funding models. See platform funding models**
- Service Inventory Scope Definition precept, 193-195**
- Service Layers design pattern, 54, 561**
- Service Logic Design Review precept, 255-257**
- Service Logic Design stage**
 - case study, 265-266
 - people for, 259-264
 - precepts for, 249-253
 - processes for, 253-258
 - in service project lifecycle stages, 87
- Service Logic Design Standards precept, 249-251**
- Service Logic Programming Standards precept, 267-268**
- Service Loose Coupling design principle, 27, 225-226, 228, 477**
- service maintenance, service versioning versus, 298**
- Service Maintenance Review process, 303-304**
- Service Messaging design pattern, 562**
- Service Metadata Standards precept, 377-380**
- service modeling process, 84-85. See also Service-Oriented Analysis stage**
- service models, defined, 38-40**
- Service Monitoring Footprint Threshold, 318**

- Service Neutral maturity level, 57**
- Service Normalization design pattern, 207-209, 344, 563**
- service-orientation**
 - defined, 26-27
 - pillars of, 51-55
 - mapping to RUP principles, 620-622*
 - RUP and, gaps in, 628
- Service-Orientation Architecture**
 - Design Standards precept, 252-253**
- Service-Orientation Contract Design**
 - Standards precept, 228**
- Service-Oriented Analysis stage**
 - case study, 217-220
 - people in, 212-217
 - precepts for, 206-210
 - processes for, 210-211
 - in service project lifecycle stages, 84-85
 - time allotted to, 189-190
- Service-Oriented Architecture: Concepts, Technology, and Design, 5, 80***
- service-oriented architecture (SOA), defined, 29**
- service-oriented computing, defined, 25-26**
- Service-Oriented Design stage**
 - people for, 236-246
 - precepts for, 223-231
 - processes for, 231-235
 - in service project lifecycle stages, 85-86
- Service Perimeter Guard design pattern, 162, 564**
- service policy precepts, 158**
- service portfolio, defined, 41-42**
- service profiles, 115-120**
 - capability profile structure, 118-119
 - service catalogs and, 119
 - service registries and, 119
 - structure of, 117
- service profile standards, 157**
- service project lifecycle stages, 81-91**
 - Service Deployment and Maintenance, 89
 - Service Development, 87
 - Service Discovery, 90-91
 - Service Inventory Analysis, 82-83
 - Service Logic Design, 87
 - Service-Oriented Analysis, 84-85
 - Service-Oriented Design, 85-86
 - Service Testing, 88-89
 - Service Usage and Monitoring, 90
 - Service Versioning and Retirement, 91
 - SOA Adoption Planning, 82
- Service Refactoring design pattern, 353, 565**
- service registries, 431-433. *See also* Service Discovery stage**
 - Centralized Service Registry precept, 335-337
 - service profiles and, 119
 - Service Registry Access Control process, 337-339
 - Service Registry Record Review process, 339
- Service Registry Access Control process, 337-339**
- Service Registry Custodian role, 105, 402-403**
 - reference diagram, 465
 - in Service Discovery, 346-347
- Service Registry Record Review process, 339**
- service-related granularity, defined, 44-45**
- Service Retirement Notification precept, 356**

- Service Retirement process, 359-360**
- Service Reusability design principle, 27, 479-480, 486**
- Service Statelessness design principle, 27, 482-483**
- Service Testing stage**
 - case study, 294-297
 - people for, 287-293
 - precepts for, 279-286
 - processes for, 286
 - in service project lifecycle stages, 88-89
 - types of tests, 278
- Service Testing Standards precept, 281-283**
- Service Test Results Review process, 286**
- Service Usage and Monitoring stage**
 - case study, 333-334
 - people for, 325-332
 - precepts for, 317-322
 - processes for, 323-324
 - in service project lifecycle stages, 90
- service versioning, service maintenance versus, 298**
- Service Versioning and Retirement stage**
 - people in, 360-366
 - precepts for, 352-356
 - processes for, 357-360
 - in service project lifecycle stages, 91
- Service Versioning process, 357-358**
- Service Versioning Strategy precept, 352-353**
- Service Vitality Review process, 323-324**
- Service Vitality Triggers precept, 320-322**
- services**
 - cloud services, defined, 36
 - as components, 32
 - defined, 31-34
 - as REST services, 34
 - scalability, 482
 - as Web services, 32-33
- SGPO. *See* SOA Governance Program Office (SGPO)**
- Shared Service Modification Request process, 343-344**
- Shared Service Usage Request process, 342-343**
- single sign-on, 161**
- single vendor acquisition strategy for governance technology, 444-445**
- SLA Template precept, 229-231**
- SLA Versioning Rules precept, 354-356**
- SOA (service-oriented architecture)**
 - defined, 29
 - governance and, 130
 - RUP (Rational Unified Process)
 - compatibility with, 618-619
 - scalability, 482
- SOA Adoption Planning stage**
 - case study, 182-186
 - people for, 179-182
 - precepts for, 169-173
 - processes for, 173-178
 - in service project lifecycle stages, 82
- SOA Certified Professional (SOACP), 15-16**
- SOA design patterns. *See* design patterns**
- SOA *Design Patterns*, 5**
- SOA governance program implementation, 137-150**
 - assessing the enterprise/domain, 137-139
 - best practices, 146-150
 - common pitfalls, 148-150
 - planning and building SOA
 - governance program, 139-146
- SOA Governance Program Office (SGPO), 131-132, 155**
 - jurisdiction models, 133-136

- SOA Governance Program Project Plan, 146**
- SOA Governance Roadmap, 146**
- SOA Governance Specialist role, 111, 406-407**
 - reference diagrams, 471-472
 - in Service Deployment and Maintenance, 311
 - in Service Development, 275
 - in Service Discovery, 348
 - in Service Inventory Analysis, 199
 - in Service Logic Design, 263
 - in Service-Oriented Analysis, 216-217
 - in Service-Oriented Design, 245-246
 - in Service Testing, 292-293
 - in Service Usage and Monitoring, 332
 - in Service Versioning and Retirement, 365
 - in SOA adoption planning, 181
- SOA governance technology standards, 163**
- SOA Governance Tools, 146**
- SOA governance vitality. *See* vitality**
- SOA Magazine, The Web site, 15, 632**
- SOA management suites, 441-442**
- SOA Manifesto, 34, 53, 578-590**
- SOA mapping, RUP and, 628**
- SOA planning**
 - funding models, 60-77
 - platform funding, 60-69*
 - service funding, 69-74*
 - organizational maturity, levels of, 56-59
 - pillars of service-orientation, 51-55
- SOA Principles of Service Design, 5, 80**
- SOA Quality Assurance Specialist role, 109, 405-406**
 - reference diagram, 469
 - in Service Deployment and Maintenance, 309
 - in Service Testing, 290-291
- SOA Security Specialist role, 110**
 - reference diagram, 470
 - in Service Deployment and Maintenance, 310
 - in Service Discovery, 339
 - in Service Logic Design, 262
 - in Service-Oriented Design, 243
 - in Service Testing, 291
 - in Service Usage and Monitoring, 331
- SOA *with* REST, 5**
- SOAP**
 - attachments, 483
 - processors, 483
- Software-as-a-Service (SaaS) delivery model, 38**
- specifications, www.soaspecs.com Web site, 15-16**
- Standardized Funding Model precept, 172-173**
- Standardized Service Contract design principle, 27, 87, 225, 228, 237, 475-476**
- standards, defined, 128**
- standards compliance tests, 278**
- standards-related precept metrics, 165**
- state management**
 - performance and, 483
 - SOAP attachments and, 483
- State Messaging design pattern, 566**
- State Repository design pattern, 567**
- Stateful Services design pattern, 568**
- strategic adjustments, as vitality triggers, 416-417**

strictness of versioning strategies, 614
 strict versioning strategy, 611-612
 sunk costs, 164
 supplemental fees in usage funding model, 66
 symbols, legend, 12

T

task services, defined, 39
 tasks in RUP, 620
 teamwork, 51, 52. *See also* pillars of service-orientation
Technical Communications Specialist
 role, 105, 403
 reference diagram, 466
 in Service Discovery, 348
 in Service-Oriented Design, 239-240
 technical editors and graphic tools, 442
 technical policies, 373
 technology changes versus business changes, in vitality triggers, 415-416.
 See also governance technology
 technology metadata, 378
 technology shifts, as vitality triggers, 418
Termination Notification design pattern, 356, 569
Test Data Usage Guidelines precept, 285
 testing. *See* Service Testing stage
Testing Parameter Standards
 precept, 280
Testing Tool Standards precept, 279-280
Three-Layer Inventory compound
 pattern, 570
 threshold metrics, 165
 time triggers, 420
 tools, defined, 427
 transport-layer security, 160
 triggers. *See* vitality triggers
 trust brokering, 160
Trusted Subsystem design pattern, 162, 571

U

UI Mediator design pattern, 572
 unit tests, 278
 up-front costs, 164
usage funding model
 in platform funding, 61, 66-69
 in service funding, 69, 74
usage thresholds, Runtime Service Usage
 Thresholds precept, 317-319
Utility Abstraction design pattern, 54, 573
 utility services, defined, 39

V

Validation Abstraction design pattern, 225, 574
version control systems, 480
Version Identification design pattern, 353, 575, 608
version identifiers, 608-611
versioning. *See also* service versioning
 compatibility and, 596-608
 constraint granularity, 595
 questions concerning, 592-593
 REST services, 594-595
 Service Candidate Versioning
 Standards precept, 209
 strategies, 611-616
 version identifiers, 608-611
 Web services, 593-594
vitality
 defined, 412
 explained, 412
 framework for, 413
vitality activities, 412, 421-424
 approve activity, 423
 assess activity, 422
 communicate activity, 423
 identify activity, 421
 refresh activity, 422-423

vitality triggers, 412, 414-421

- business changes versus technology changes, 415-416
- industry shifts, 417-418
- metrics, 418-419
- organizational shifts, 419-420
- periodic triggers, 420
- Service Vitality Review process, 323-324
- Service Vitality Triggers precept, 320-322
- strategic adjustments, 416-417

W***Web Service Contract Design and Versioning for SOA, 80*****Web services**

- defined, 32-33
- versioning, 593-597

Web sites

- www.cloudschool.com, 15-16, 632
- www.cloudsymposium.com, 633
- www.serviceorientation.com, 633
- www.soabooks.com, 6, 14, 16, 48, 632
- www.soabooks.com/governance/, 431
- www.soaglossary.com, 5, 15-16, 48, 632
- www.soamag.com, 15, 632
- www.soa-manifesto.com, 34, 53, 578
- www.soa-manifesto.org, 34, 53, 578
- www.soapatterns.org, 633
- www.soaprinciples.com, 47, 633
- www.soaschool.com, 15, 632
- www.soaspecs.com, 15, 139, 632
- www.soasymposium.com, 633
- www.whatiscloud.com, 633
- www.whatissoa.com, 47, 633

wisdom, defined, 372

work products in RUP, 620

WS-Policy, 162, 476

WS-Policy assertions, 355

WS-PolicyAttachment, 162

WS-SecureConversation, 161

WS-Security, 161

WS-SecurityPolicy, 161

WS-Trust, 161

WSDL languages, 476

X–Z

XML-Encryption, 161

XML parsers, 483

XML schema languages, 476

XML-Signature, 161