INDEX

Abstracted Service:
  Amazon Cloud crashes and,
  220–221
  managing of, 58–59
  SOA and, 31–33
ACID (atomic, consistent, isolated, and durable) transactionality,
  191, 192
ADM (Architecture Development Method), 14, 243–244, 249
“Agile,” deconstructed, 4–8
Agile Architecture, 3–19
  “agile” deconstructed, 4–8
  “architecture” defined, 12–13
Complex Systems and, 16–19
dangers of checklist architecture,
  253–254
difficulties of, 31
Enterprise Architecture not being done, 13–15
future of, 257–260
how to buy, 250–251
key to successful, 17
meta thinking and, 10–12
paradigm shifts in thinking about,
  257–258
RFP mistakes, 251
RFP pointers, 252–253
software/human systems and, 8–10
as true revolution, xv–xvi, 3
Agile Architecture, in practice,
  89–102
  agility meta-requirement and,
  90–91, 100–102
aligning with BPM, 96–98
business process definitions, 89
composition-centric approach to automation, 89–93
legacy applications and, 92, 95, 101
modeling and, 98–100
SOA and integration challenges, 93–96
Agile Architecture Quality Star,
  78–79
Agile Manifesto:
  changing software requirements and, 7–8
  checklist architecture contrasted, 255
  four core principles of, 4–5
  paradox of, 10
Agility Model:
  key capabilities of, 84
  Richardson Maturity Model (RMM) and, 158–160
Agility Model, governance and,
  60–63
diagram of, 62
measuring of, 61–62
Amazon:
  Cloud Computing and, 204, 207
  lessons of Cloud crashes, 219–221
  shopping cart and Cloud Computing, 232
Ambiguity, allowing for, 81
APIs (Application Programming Interfaces (APIs), 27, 150–152, 219
Application, meaning of, 149
Application architecture, 246
Application portability, 189
Application Programming Interfaces,
  27, 150–152, 219
Application state, Cloud Computing and, 226, 229–235
Application tier, Cloud Computing and, 231, 233–235

“Architecting the enterprise,” 9

Architecture, defined, 12–13

Architecture Development Method (ADM), 14, 243–244, 249

Architecture-driven governance, 53, 54–57

Complex Systems Engineering and, 76

“Argument,” as collective term for architecture, 12

Attractors, in Complex Systems theory, 18–19

Automated configurations, Cloud Computing and, 219, 224–225

Automation, in architecture-driven governance, 55

Babbage, Charles, 217

BASE (basic availability soft state, eventual consistency), 192–193

Best-effort quality, 76–80

in action, 80–82

Agile Architecture Quality Star, 78–79

Best-Effort Triangle, 78–79

Big brother effect, 76

Big Data Explosion Crisis point, 115, 116–119

BPM (Business Process Management), 29

Cloud Computing and states, 228–229, 232–236

SOA, dynamic changes in Services, and Agile Architecture in practice, 90–93, 96–98, 100

Business agility:

Complex Systems Engineering and, 110–111

defined, 6

proactive, strategic part, 6–7

reactive, tactical part, 6

Business(es):

business architecture, 246

business context, of Service, 42–43

business/IT alignment, 41–42

defining of, 4

definitions of business processes, 89

gap between needs and software functions, 6

Business Process Execution Language (BPEL), 172

Business Process Management (BPM), 29, 228, 232

Cloud Computing and states, 228–229, 232–236

SOA, dynamic changes in Services, and Agile Architecture in practice, 90–93, 96–98, 100

Business Process Modeling and Notation (BPMN), 172

Business Service abstraction, 21–23, 41, 46, 47–48

Business value, Cloud Computing and Service Level Agreements and, 211

Butterfly Effect, 70–72

Capital expense, Private Clouds and, 203

CAP (consistency, availability, partition tolerance) Theorem, 191

CSE (Complex Systems Engineering), 73–87, 121, 255

best-effort quality, 76–80

best-effort quality, in action, 80–82

characteristics of, 75

demise of Enterprise IT and, 136

as Enterprise IT supertrend, 110

flash mobs and self-organization, 86–87
node-centric perspective and, 75–76
resilience and, 83–86
simplicity of, 74
traditional systems engineering
contrasted, 73–74
Change:
governance and meta-policies,
58–60
quality and change-time
requirements, 56, 59–60, 63, 255
responding to, in Agile Manifesto,
4
Chaos theory. See Butterfly Effect
Chargebacks, Cloud Computing and,
207
Checklist architecture, dangers of,
253–254
Chief Information Officers (CIOs),
SOA’s lack of success and, 29
Churn, in Cloud, 207
“Client,” as used in REST, 161
Clinger–Cohen Act, 254
Cloud Computing, 177–241
Amazon crash lessons, 219–221
architecture and challenges of,
179–182
BPM and, 228–229, 232–236
challenges of, 30–31
characteristics of, 179
Cloud as the computer, 241
Cloud-Oriented Architecture and,
236–238
configuration management, 223–225
data consistency and, 190–193
data garbage in, garbage out and,
214–215
demise of Enterprise IT and,
135–136, 141, 145
deployment models, 177–178
deployment scenarios, 187–188, 189
disadvantages of Private Clouds,
202–205
economic pitfalls and, 205–208
Enterprise Public Cloud and,
197–200
failure expectations and plans for,
221–222
fake clouds, 217–219
HATEOAS and, 231–232,
233–234
location independence and Next
Big Thing, 238–241
multitenancy and, 193–197
NIST definition of, 177
REST, SOA and states, 225–232
security of Public and Private
Clouds, 200–202
Service Level Agreements and,
208–211
service models, 178
SOA and, 24, 27
software licenses and, 212–214
vendor-centric model, 179–184
Cloud-Oriented Architecture
(COA), 236–238
location independence and Next
Big Thing, 238–239
Cloudwashing, 212
Clustered Shared Schema Approach
(second degree multitenancy),
advantages and disadvantages of,
195–196
COA (Cloud-Oriented
Architecture), 236–238
location independence and Next
Big Thing, 238–239
Collaboration, composition-driven
business processes and, 91
Commercial off the shelf (COTS)
software, lack of agility in, 7,
145–146
Communication, in architecture-
driven governance, 54
Community Cloud, 178
Complex Systems:
emergence, as defining characteristic of, 15
fixed requirements discouraging emergence, 16–17
inherently collaborative nature of, 19
self-adapting nature of, 18
simplicity of, 16
successful innovation examples of, 17–18
as systems of systems, 16
Complex Systems Engineering (CSE), 73–87, 121, 255
best-effort quality, 76–80
best-effort quality, in action, 80–82
characteristics of, 75
demise of Enterprise IT and, 136
as Enterprise IT supertrend, 110
flash mobs and self-organization, 86–87
node-centric perspective and, 75–76
resilience and, 83–86
simplicity of, 74
traditional systems engineering contrasted, 73–74
Composition-centric approach to automation, 89–93
Configuration management, in Cloud Computing, 223–225
Consulting firms, SOA’s lack of success and, 29–30
Continuous Business Transformation, 110–112
Contracts. See also Service Level Agreements (SLAs)
contract variability, Agility Model and governance, 61
SOA and contracts, 34–38
Cookies, 227
CORBA (common object request broker architecture), 22, 38
Corporate governance, defined, 52
COTS (commercial off the shelf) software, lack of agility in, 7, 145–146
Cost efficiencies, Private Clouds and, 204
Create Employee Service, example of document-style interactions, 153–155
Crisis Points, of ZapThink 2020 Vision, 107, 113–116
Big Data explosion, 115, 116–119
collapse of Enterprise IT, 114
cyberwar, 115, 119–125
enterprise application crash, 115
fall of frameworks, 114–115
Generation Y, 109, 115, 128–131
IPv4 exhaustion, 114
CRM (customer relationship management), Cloud Computing and, 141, 212
Crowdsourcing:
Complex Systems and, 18
Cyberwar Crisis Point, 124
Customer collaboration, in Agile Manifesto, 4
Customer relationship management (CRM), Cloud Computing and, 141, 212
Cybersecurity, 125–128
scenarios, 125–127
software immune system and, 127–128
protecting against, 122–125
DoDAF (Department of Defense Architecture Framework), 114, 245, 248, 254
Data:
Big Data Explosion Crisis point, 115, 116–119
data architecture, 246
Private Clouds and data centers, 203
Data, Cloud Computing and:
  consistency of, 190–193
data centers and, 236–238
garbage in, 214–215
garbage out, 216–217
portability of, 189
Data structure variability, Agility
  Model and governance, 62
DCOM (Distributed Component
  Object Model), 38
Deadlocks, states and Cloud
  Computing, 230
Declarative programming, 151–152
Deep Interoperability:
  as Enterprise IT supertrend, 110,
  112–113
  REST and, 168–170
“DELETE,” as used in REST, 162,
  163, 164
Democratization of technology, 145
demise of Enterprise IT and, 136
  as Enterprise IT supertrend,
  109–110
Democratization of technology, as
  Enterprise IT supertrend, 109–110
Department of Defense Architecture
  Framework (DoDAF), 114, 245,
  248, 254
Deployment models, of Cloud
  Computing, 177–178
Design-time governance, 55, 56
DevOps, 225
Distributed Component Object
  Model (DCOM), 38
Distributed computing, 150
Document-style Web Services
  example of, 153–155
  RPC contrasted, 152–155
EA. See Enterprise Architecture (EA)
EAI (Enterprise Application
  Integration), 28
eBusiness, 44
Edison, Thomas, 80
Elasticity:
  Cloud Computing and, 179,
  188–190
  Cloud Computing and Service
    Level Agreements and, 210–211
  Cloud Computing and utility,
    205–206
  failure expectations and plans for,
    222
  fake clouds and, 217–218
  Private Clouds and, 203
Emergence, as defining characteristic
  of Complex Systems, 15
Emergent property, business agility
  as, 9
Empowerment, of users, 101–102
End-to-end, continuous Hybrid
  Cloud testing, required of
  Public Cloud, 199
End-user license agreements
  (EULAs), 209, 210
Enterprise application crash Crisis
  Points, of ZapThink 2020
  Vision, 115
Enterprise Application Integration
  (EAI), 28
Enterprise Architecture (EA), 9–10
Complex Systems and, 18
Continuous Business
  Transformation, 111
framework and methodologies of,
  245–250
not being done, 13–14
  professional credentialing and,
  243–244
role today, 15
SOA’s lack of success and, 29
Enterprise Architecture Center of Excellence (EACOE), 111
Enterprise IT, Supertrounds of, 108–110
Complex Systems Engineering, 110
deep interoperability, 110
democratization of technology, 109–110
global cubicle, 109
location independence, 108–109
Enterprise IT Crisis Point:
Agile Architecture approach to project management, 136–138
application crash, 138–144
Cloud Computing and demise of, 114, 135–136, 141, 145
costs and steps in using Enterprise IT, 133–134
difficulty of replacing, 144–146
Enterprise operations context, of SLAs, 209
Enterprise Public Cloud Provider marketplace, 204–205
Enterprise Public Clouds, 197–200
Enterprise resource planning (ERP), Cloud Computing and, 141, 212
Enterprises:
grown, not architected, 15
as systems of systems, 9
Enterprise Service Bus (ESB), 187
pitfalls of, 44–47
as service intermediary, 47–49
SOA with, 40
SOA without, 47–50
EULAs (end-user license agreements), 209, 210
Eventual consistency, Cloud Computing and, 191–193
Evolution, in definition of architecture, 13
Executives, SOA’s lack of success and, 30
eXtensible Markup Language (XML)-based interfaces, 152
Facebook, 219
Failure, expecting and planning for, in Cloud Computing, 221–222.
See also Best-effort quality
Fake clouds, 217–218
questions to ask regarding, 218–219
Fielding, Roy, 147, 156–157
definition of REST, 157
Financials, of Cloud providers, 208
Fine-grained governance, required of Public Cloud, 199
Flash Crowd (Niven), 86
Flash mobs, self-organization and, 86–87
Flexibility:
designing for, 81–82
as key to profitability, longevity, success, 6
Frameworks:
Crisis Point of fall of, 114–115
EA and, 245–250
Functional programming, declarative nature of, 150–152
Garbage in, garbage out, Cloud Computing and, 214–217
Generation Y Crisis Point, 109, 115, 128–131
GET,” as used in REST, 162, 163, 164
Global Cubicle:
demise of Enterprise IT and, 136
as Enterprise IT supertrend, 109
Governance, 51–72
Agility Model of, 60–63
architecture-driven, 54–57, 76
Butterfly Effect and, 70–72
Cloud Computing and, 207–208, 217
Complex Systems Engineering
and, 110
composition-driven business
processes and, 90, 91, 93, 101
corporate, defined, 52
demise of Enterprise IT and, 135
distinct parts of, 55–56
as driver of Services, 9
fine-grained governance, required
of Public Cloud, 199
interrelationships of quality,
management and, 64–66
IT’s role in, 51–52
meta-policy and, 63–64
organization context, 52–54
quality and, 57–60
questions to ask regarding
framework for, 54
stages of Agile Architecture
governance, 67–69
user empowerment and, 101–102
Hall of mirrors problem, 11
HATEOAS (hypermedia as the
engine of application state):
Cloud Computing and, 231–232, 233–234
REST issues and, 158–159,
162–163, 166–167, 169
SPEAR and, 174
Heraclitus, 110
Heterogeneity, leveraging of, 50
HoneyMonkey project, of Microsoft, 127
Hosted Private Cloud, 177
Human-centric architecture-driven
governance, 54, 67, 69
Human-centric SOA governance,
67–68
Human/technology systems, 17
Hybrid Clouds:
defined, 178
ded-to-end continuous testing,
required of Public Cloud, 199
integration, required of Public
Cloud, 199
Hypermedia:
application state and, 231
as term used in REST, 161
Hypermedia as the engine of
application state (HATEOAS):
Cloud Computing and, 231–232, 233–234
REST issues and, 158–159,
162–163, 166–167, 169
SPEAR and, 174
Hypermedia-Oriented Architecture
(HOA), 233–234
location independence and Next
Big Thing, 238–239
Hypertext Transfer Protocol
(HTTP), 156
IA (Information Assurance), 126
IANA (Internet Assigned Numbers
Authority), 114
IaaS (Infrastructure-as-a-Service),
Cloud Computing and, 178,
212–213
IBM, Rational Unified Process of, 5
Immediate consistency, Cloud
Computing and, 191–193
Imperative programming, 149–150
Implementation variability,
Agility Model and governance,
61
Individual and interactions, in Agile
Manifesto, 4
Infinite capacity, Clouds and illusion of, 179
Information Assurance (IA), 126
Infrastructure-as-a-Service (IaaS),
Cloud Computing and, 178,
212–213
Infrastructure costs, 212
Private Clouds and, 203
Infrastructure variability, Agility
Model and governance, 61
Institute of Electrical and Electronics
Engineers (IEEE), definition for
architecture, 12
Integration-centric approach to
automation, 89, 94
Integration, organizational behavior and, 82
Iterative methodologies, 5
Internet Assigned Numbers
Authority (IANA), 114
IPv4 exhaustion, as Crisis Point in
ZapThink 2020 Vision, 114
Isolated Tenancy Approach (third
degree multitenancy),
advantages and disadvantages of, 196–197
IT context, of Service, 42–43
Iterative/agile software project,
5–6
IT governance. See Governance
IT infrastructure vendors, SOA’s lack of success and, 28–29
Java Message Service (JMS) interface,
175
Java Remote Method Invocation (RMI), 151
Knowledge management, in
architecture-driven governance,
55
Kuhn, Thomas, 103, 106, 113
Legacy applications:
options for Cloud Computing,
182–183, 187–189
SOA and composition-driven
business processes and, 92, 95,
101
Location independence:
as characteristic of Cloud
Computing, 179
as Enterprise IT super trend,
108–109
Next Big Thing in Cloud
Computing and, 238–241
LOIC (Low Orbit Ion Cannon), 124
Loose coupling:
complex systems and, 75
document-style and, 152–155
SOA and, 32–33, 37–38, 61
Lottery fallacy, 126–127
Low Orbit Ion Cannon (LOIC), 124
Managed hosting provider context,
of SLAs, 209
Management, interrelationships of
quality and governance and,
64–66
Management-quality feedback loop,
66
Manes, Anne Thomas, 23
“Many-to-many-to-many,” 32, 33
Maps PlaceFinder Geocoding API,
of Yahoo!, 167–168
MasterCard, 122
MDA (Model Driven Architecture), 11
Measured service, as characteristic of
Cloud Computing, 179
Measurement, frequency of, 82
Meta-architecture, 12
Meta-best practices, 24
Metadata:
  metadata artifacts, 53
  role in SOA, 37
Meta-meta-models, 11
Meta-methodology, 249–250
Meta-policies, 57
  feedback loop, 118–119
  governance and, 63–64
Meta-requirements:
  agility and, 8, 90–91, 100–102
  agility and quality, 77
  business agility and SOA, 12
Methodologies, EA and, 245–250
Microsoft Azure, 204
Microsoft.NET, 151
Microsoft Research, HoneyMonkey project of, 127
Middleware, SOA’s lack of success and, 28–29
Millennials. See Generation Y Crisis Point
Model Driven Architecture (MDA), 11
Modeling, Agile Architecture and, 98–100
Moore’s Law, 116–117
Multitenancy, Cloud Computing and, 179, 193–197
  Clustered Shared Schema Approach (second degree), 195–196
  Isolated Tenancy Approach (third degree), 196–197
  Shared Schema Approach (first-degree), 193–195
National Institute for Standards and Technology (NIST):
  Cloud deployment scenarios, 187–189
definition of Cloud Computing, 177, 236
  interoperability and portability standards, 188–190
Negroponte, Nicholas, 257
.NET Remoting, 151
Network access, broad, as Cloud Computing characteristic, 179
Network effect, Complex Systems and, 18
Next-generation modularization, 145
NIST (National Institute for Standards and Technology):
  Cloud deployment scenarios, 187–189
  definition of Cloud Computing, 177, 236
  interoperability and portability standards, 188–190
Niven, Larry, 86
Node-centric perspective, complex systems engineering and, 75–76
Object-Oriented Analysis and Design (OOAD), 37
Offshoring, Enterprise IT and, 134–135
OOAD (Object-Oriented Analysis and Design), 37
Open source software, Cloud Computing and, 213–214
Open standards, deep interoperability, 112–113
Oracle Database Cloud, 218
Organization of systems, in definition of architecture, 13
Outsourcing, Enterprise IT and, 134–135
PaaS (Platform-as-a-Service), Cloud Computing and, 178, 183–184
Partition tolerance, Cloud Computing and, 191
PayPal, 122
Performance objectives (POs), 254
Platform-as-a-Service (PaaS), Cloud Computing and, 178, 183–184
Platform-level security measures, required of Public Cloud, 199
Policies. See also Meta-policies as driver of Services, 9
variability, Agility Model and governance, 62
Portfolio management, in IT, 137–138
“POST,” as used in REST, 162, 163, 164
Post Finance, 122
Pricing models, Cloud Computing and, 206–207
Private Clouds, 177
disadvantages of, 202–205
governance and, 217
less secure than Public Clouds, 200–202
Public Clouds contrasted, 197–198
Service Level Agreements and, 210
vendors and, 183–184
Proactive strategic part, of business agility, 6–7
Process variability, Agility Model and governance, 61
Project management, Agile Architecture approach to, 136–138
Public Clouds:
Amazon Cloud crashes, 219–221
capabilities required of, 198–199
Enterprise Public Cloud and, 197–200
governance and, 217
more secure than Private Clouds, 200–202
multitenancy and, 193
Private Clouds contrasted, 197–198
Public Cloud Providers, 177
“PUT,” as used in REST, 162, 163, 164
QoS (quality of service) metrics, 56
Quality:
best-effort quality, 76–80
best-effort quality, in action, 80–82
change and, 58–60
governance and, 57–60
interrelationships of management, governance, and, 64–66
traditional approach to, 58
Quality of service (QoS) metrics, 56
Race conditions, states and Cloud Computing, 230
Rackspace, 204
Rational Unified Process, of IBM, 5
Reactive part, of business agility, 6
Reliability, Cloud Computing and Service Level Agreements and, 210–211
Remote Method Invocation (RMI), of Java, 151
Remote Procedure Call (RPC) programming challenges of, 150–151
in Create Employee Service example, 154
Web Services and, 152–155
“Representation,” as used in REST, 161
Representational State Transfer (REST), 149–176. See also
Representational State Transfer (REST)-based SOA
as abstraction, 157
Agility Model and, 159–160
architectural constraints on, 162–163
as architectural style, 156, 157
best uses, 160
deep interoperability and, 168–170
definitions, 155–157
as distributed hypermedia application, 157–158
HATEOAS and, 158–159, 162–163, 166–167, 169
hypermedia-oriented architecture and, 233–234
precursors of, 149–155
Richardson Maturity Model and, 158–159
evocabulary of, 161–162
Web Services and, 163–168
Representational State Transfer (REST)-based SOA, 27, 40
Cloud Computing and states, 225–232
SOA in, 170–173
SPEAR initiative and, 173–176
“Representations of business capabilities,” 32
Reprovisioning, in Cloud Computing, 224
Request for Proposal (RFP), for Agile architecture:
mistakes in, 250–251
pointers for, 252–253
Requirements, of software:
business agility and, 6–7
changing, 7–8
fixed, and lack of emergence, 16–17
flexibility and, 6
myth of complete, well-defined, 5
Resilience, 83–86
at infrastructure level, 85
Service policies and, 85
ways to provide, 84
“Resource,” as used in REST, 161
Resource-Oriented Architecture (ROA), 237
Resource pooling, as characteristic of Cloud Computing, 179
Resources, REST architectural constraints and, 162
Resource state, Cloud Computing and, 226, 230–232, 235
Retrospectives, Scrum use and, 10
Revolutions, difficulty recognizing while in progress, 105–108
Richardson, Leonard, 158
Richardson Maturity Model (RMM), 158–159, 169
Agility Model and, 159–160
ROA (Resource-Oriented Architecture), 237
RPC (Remote Procedure Call) programming challenges of, 150–151
in Create Employee Service example, 154
Web Services and, 152–155
Run-time governance, 56
Run-time work flow, 235–236
Salesforce.com, 209–210, 215
SCM (supply chain management), Cloud Computing and, 141, 212
Scrum:
best practices, 10
meta-methodology, 11
Scrum Buts, 10
Complex Systems and, 18–19
schools of thought on, 10–11
Security measures:
platform-level, required of Public Cloud, 199
Public versus Private Clouds, 200–202
Self-descriptive messages, REST architectural constraints and, 162
Self-healing, composition-driven business processes and, 91
Self-organization, 86–87
Self-reference paradoxes, 11
Self service, as Cloud Computing characteristic, 179
Semantics tooling, Complex Systems and, 18
Semantic variability, Agility Model and governance, 62
“Server,” as used in REST, 161
Server utilization, Private Clouds and, 203
Service, defining of, 4
Service Level Agreements (SLAs), Cloud Computing and, 208–211
contexts of, 209–211
Service models, of Cloud Computing, 178
Service Orientation (SO) conversation, 42–44
Service Oriented Architecture (SOA), 9, 21–50
best practices and, 24–28
Centers of Excellence, 27
Complex Systems and, 16–17
composition-driven business processes and, 93–96
demise of Enterprise IT and, 145
document-style Service interactions and, 153
as evolutionary, not revolutionary, 105
framework and methodologies of, 245–250
freeing architecture from underlying infrastructure, 44–50
governance and Complex Systems, 17
hindrances to success of, 28–31
location independence and Next Big Thing, 238–241
loose coupling and, 32–33, 37–38
move to composition-centric approach to automation and, 89–93
multi-contracts and policy-driven behavior, 34–38
outside the box thinking about, 26–28
quality and, 77
REST and Cloud Computing, 227
in REST-based SOA, 170–176
REST contrasted, 163–165
Service as business abstraction, 21–23
Service as core of, 31–33
Service Orientation conversation and, 42–44
“SoA is dead” meme, 23–25
as style of Enterprise Architecture, 156
Web Services and, 27, 38–40
what’s not part of, 22
Service policies, resilience and, 85
Services:
  composition-centric approach to automation and, 90
  as core of SOA, 31–33
Shared Schema Approach (first-degree multitenancy), advantages and disadvantages of, 193–195
Shopping carts, REST state example of Cloud Computing and, 229–232
  See SOAP)
SOA. See Service Oriented Architecture
SOAP (Simple Object Access Protocol), 39, 152
Web Services and REST, 163–165
Social media, Cyberwar Crisis Point and, 123–124
Soft state, Cloud Computing and, 192
Software. See also Applications; Legacy applications
gap between functions and business needs, 6
not-so-agile updates to, 7
Software-as-a-Service (SaaS), Cloud Computing and, 178, 218–219
Software Development (SD) Triangle, 78
Software/human systems, 8–10
Software licenses, Cloud Computing and, 212–214
Software Security Assurance (SSA), 126
Software vendor context, of SLAs, 209
SPEAR architecture, of U.S. Coast Guard, 173–176, 229
Spiral methodology, 5
Sprints, Scrum use and, 10
States, Cloud Computing and concept of:
data and, 229–230
REST and, 225–232
ways to maintain, 227–229
Structure of Scientific Revolutions, The (Kuhn), 106
Stuxnet, 107, 119–121, 123
Sun Microsystems, 171
Supply chain management (SCM), Cloud Computing and, 141, 212
Systems of systems, 9
Complex Systems and, 16–17
Taylor, Frederick Winslow, 1
Technology, democratization of, 145
demise of Enterprise IT and, 136
as Enterprise IT supertrend, 110
Technology architecture, 246
Technology-centric architecture-driven governance, 67–69
Technology-centric SOA governance, 67–68
TSE (traditional systems engineering), contrasted to Complex Systems engineering, 73–74
The Open Group Architecture Framework (TOGAF), 14, 114, 243–244, 246, 248, 249
Tiered pricing, Cloud Computing and, 206–207
Tight coupling, RMI and, 151–152
Tim Woodman pattern, 49–50
TOGAF (The Open Group Architecture Framework), 14, 114, 243–244, 246, 248, 249
Traditional systems engineering (TSE), contrasted to Complex Systems Engineering, 73–74
Training, in architecture-driven governance, 54
trust.salesforce.com, 210
Twitter, 17
Ubiquitous Computing, 124
UDDI (Universal Description, Discovery, and Integration), 39, 164
“Uniform Resource Identifier (URI),” as used in REST, 161
Uniform Resource Names (URNs), 161
Universal Description, Discovery, and Integration (UDDI), 39, 164
URNs (Uniform Resource Names), 161
“URI (Uniform Resource Identifier),” as used in REST, 161
U.S. Coast Guard SPEAR architecture, 173–176, 229
Users:
  empowerment of, 101–102
  as part of agile system, 8
Variability, planning for, 84
Vendor-centric nature of Cloud Computing, 179–184
Versioning, of Service, 63–64
Viral marketing, Complex Systems and, 17
Virtual machine (VM) architecture, of Java, 151
Virtual Private Clouds (VPCs), 201–202
Virtual Private Networks (VPNs), 179
WADL (Web Application Description Language), 171
Waterfall software project, 5
Web Application Description Language (WADL), 171
Web-Oriented Architecture (WOA), location independence and Next Big Thing, 239–240
Web Services, 152–153
  REST contrasted, 163–168
  RPC programming and, 152–155
  RPC versus document-style, 153–155
  SOA and, 27, 38–40, 49
Web Services Description Language (WSDL), 31, 37–40, 152–153, 155
  REST and, 163–165, 166, 171
  WikiLeaks, 122, 123, 124
  Wikipedia, definition of REST, 156, 157
  WOA (Web-Oriented Architecture), location independence and Next Big Thing, 239–240
  Working software, in Agile Manifesto, 4
  WS-*, 39–40
  WSDL (Web Services Description Language), 31, 37–40, 152–153, 155
  REST and, 163–165, 166, 171
  Yahoo! Maps PlaceFinder
    Geocoding API, 167–168
  Zachman Framework, 14, 42, 114, 243, 248–249
  ZapThink, xvi, 4, 12, 23, 77, 94, 236
definition of business processes, 89
  ESB and, 47, 49–50
  Web Services and, 39
  ZapThink 2020 Vision, 105–131, 248, 249
  Big Data Crisis Point, 116–119
coming paradigm shift and revolution in progress, 105–108
  Continuous Business Transformation and, 110–112
  Crisis Points of, 107, 113–116
cybersecurity and, 125–128
  Cyberwar Crisis Point, 119–125
  Deep Interoperability, 110, 112–113, 168–170
  Enterprise IT Crisis Point, 114, 134–136
  Generation Y Crisis Point, 109, 115, 128–131
  poster, xvi, xvii