INDEX

A
acceptance testing, 32
account management services, 104, 106, 110–112, 117
account reporting services, 104, 106, 108–110, 116, 117
activities
  in design substantiation, 184
  as pressure points, 214
  vs. processes, 166, 168
  in system-level behavioral decomposition, 170–172
Air Force system failure example, 14–15
application(s)
  changes to, 156, 158
  definition of, 55
  physical distribution of, 68
application capacity planning, 61, 235, 236
application dependencies
  cataloging, 77–81
  mapping, 70–71
  patterns of, 76–77
  in sub-architectures, 110
  and system failure, 24
  types of, 72–76
application discovery
  architecture, 58–61
  asset management, 64–66
  and capacity planning, 237–238
  continuous, 56
  function of, 55, 67
  identity, 57–58, 78
  network infrastructure, 63–64
  performance/resource utilization, 61–63
  and system failure detection, 24
  tools for, 56–57
application-level behavioral decomposition, 163–165, 167–169, 172, 173
application-level design, 86, 87–88, 90–91
application-level pressure points, 215–218, 224
application life cycle management (ALM), 65
application mapping
  of dependencies, 24, 70–77
  function of, 67, 68
  of topology, 67–70, 238
application performance management (APM), 239
architecture
  application, 58–61
  capabilities of, 120, 122–128
elasticity of, 71, 248
enterprise (see enterprise architecture)
  grand-scale (see grand-scale architecture)
architecture decomposition
  business-driven, 9, 145–160
  and capacity planning, 237, 238
  classification in, 91–102, 203
  defined, 5
  functional (behavioral), 161–178, 214
  grand-scale, 116–117, 138–144
  sub-architectures in, 103–114, 120
  technical analysis in (see technical analysis)
architecture decoupling scale, 91–94, 101, 109–114, 133, 137
architecture design
  application-level, 86, 87–88, 90–91
  vs. development, 3, 6
  design flaws, 189, 190, 191, 194, 199, 232
  end-state, 2–3
  enterprise-level, 86–87, 88–91
  to reduce pressure risk, 216
  unfulfilled requirements of (debt), 184
  verification of, 176–178, 194–195
  See also design substantiation
architecture distribution scale, 91–94, 101, 109–114, 133, 137
architecture expansion
  and chained message route fabric pattern, 50
  and compound fabric patterns, 52
  and federated architecture type, 89
  and system failure, 26–27, 28
architecture heterogeneity, 85
architecture patterns, 24, 28
architecture segmentation
  analysis/refinement, 189, 191–192
  construction/integration, 189, 192–193
  design loop-back, 189, 191, 194–195
  testing, 192–193
  validation/certification, 189, 193–194
architecture structure, 202–203
architecture verification
  authentication tasks in, 9
  and capacity planning, 236, 238
  and decomposition process, 149, 176
  design substantiation in, 181–196
  function of, 5–6, 179–180
  in incremental approach, 8–9, 83
  and software modeling/simulation, 7
  and stress testing, 224–225
architecture verification principal, 188–189, 194
asset(s)
crosscutting, 163–165, 170, 172, 173, 215, 217
distribution of, 24, 28, 200
integration of, 203, 215, 224
reuse of, 125, 126, 166
volatility of, 159
asset management, 64–66, 77–81
authentication server, 127, 142
autonomy, 61
B
balance scorecard, 70
bandwidth
and capacity planning, 235, 239
and network congestion, 198
in network infrastructure discovery, 63
and pressure points, 224
behavioral (functional) decomposition
application-level, 167–169
cascading behaviors in, 164
component-level, 165–167
diagram for, 173–176
model for, 163–165
purpose of, 161–162, 172–173
results of, 176–178
system-level, 170–172
behavioral pressure points, 200, 214–218, 225–226, 228, 244
best practices
and change initiatives, 7–8
to drive design substantiation, 186
bill pay application, 104, 106, 112–114, 116
blocking, 73
bottom-up architecture evolution, 8
breaking points, 227–229, 238–239, 241, 244–247, 249
breaking points discovery stress testing, 231–232
bridge fabric patterns, 50–51, 52, 53, 77, 206, 207, 211
broker-based architecture, 90, 123
budgets
and architecture verification, 179
and capacity planning, 32, 62, 236, 248, 250
and design substantiation, 188
and enterprise architecture design, 2, 3, 181
and system failures, 14, 19, 20, 29, 30
and system reuse, 26
bus architecture, 89
business analysis
for sub-architecture discovery, 103–117
for tracing architecture evolution, 117–118
business continuity
and application dependency discovery, 71
and asset management, 64
and circular message route fabric pattern, 48
ensuring, 33
and pressure points, 214, 218, 219, 232
business discovery, 102
business documentation, 237
business domains (lines)
contextual, 146
definition of, 146
as pressure points, 212–214
structural, 146–156
business federation, 44
business granularity, 149, 151, 154
business impact analysis, 20
business layer
in application architecture, 59
in application mapping, 69
structural decomposition of, 151
business-level system failures, 17, 19–21
business logic layer, 60, 61, 76, 159
business model
and architecture discovery/analysis process, 37
and business-driven architecture decomposition, 145, 154
perpetual transaction cycles in, 47
and system failure, 21
business ownership
in architecture decomposition, 145
and business change, 219
business process models, 37
business services
sub-architectures supporting, 103, 104–105
volatility of, 158, 159, 160
business solutions
and application architecture type, 60
and application discovery, 55
and system fabric discovery, 39
business strategy
and architecture discovery/analysis, 37, 58
and architecture evolution, 118, 219
and business domain distribution, 154
sub-architecture execution of, 104
and technical analysis, 141
business tiers, 149–151
business transactions
linear, 49
discovery process, 24
mapping routes of, 67, 74
stability of, 214
tracing, to discover sub-architectures, 103–104, 107
business view, 145
business volatility, 218–222
bus technologies, 43, 44
C
capacity planning
architecture, 61–63, 235, 236
enterprise, 9, 32, 235, 236, 248–251
function of, 199, 235–236, 248
methodology for, 236–242
modeling in, 241, 242–247
in verification process, 180
capacity predictive modeling, 62
capacity utilization testing, 63
centralized business domain, 151–153
central processing unit (CPU), 32, 62, 80, 216, 235, 239, 250
change
in architecture segments (volatility), 200
in business segments (volatility), 156–160
organizational, 7–9
in production environments, 218–222
change management, 24
circular business transactions, 46–47, 49
circular message route fabric pattern, 46–48, 77, 206, 207
client-server application architecture, 61
cloud computing, 125–126, 173, 222, 227
clustered business domain, 154–155, 212
combined segment testing, 193
common utilities, 60
communication platforms, volatility of, 157
communication protocols, 58
community center fabric pattern, 41–43, 77, 97, 131, 135–136, 206, 207, 210
community fabric pattern, 40–41, 52, 77, 96, 206, 207
component(s)
definition of, 165
reuse of, 169, 176, 216
component-level behavioral decomposition, 163–167, 172, 173–174
component-level pressure points, 215–217, 224
computing environment, size of, 235
configuration management
and application discovery, 65
and system failure detection, 24, 33
consumer(s)
in application architecture diagram, 59, 60
enterprise connection to, 58
consumer demand
and architecture pressure, 199
and business volatility pressure, 221
performance data collected on, 242
consumer segmentation analysis, 20, 37
continuous application discovery, 56, 68
conversation, 72, 73
crosscutting layer, 59
cross-functional contracts, 185–195
crossroad fabric patterns, 50–51, 52, 53, 77, 206–207, 211
CRUD (create, read, update, delete), 75, 79, 124, 227
customer data integration (CDI), 127

c
data abstractions, 124
data access layer (DAL), 27, 32, 51, 59, 61, 75, 79, 124, 142, 159, 172
data aggregator, 127–128, 142
data analysis, in capacity planning, 62, 240–242
data cataloging, 79–80
data collection, for capacity planning, 237, 238–240
data collection technologies, 239–240
data exchange format, 80
data generation technologies, 240
data integrity, 242
data link layer, 80
data mining engine, 128
data source providers, 60, 210, 225–227
data storage
caches, 75
in capacity planning, 235, 239, 250
warehouses, 69, 74, 79, 222
decentralized organizations, 1, 8
decomposition. See architecture decomposition
defect tracking, 65
delta analysis of applications, 56
design. See architecture design
design flaws, 189, 190, 191, 194, 199, 232
design patterns, 89
design substantiation
cross-functional contracts in, 185–186
function of, 181–185, 214
process for, 189–195
team members for, 186–189, 195–196
in verification process, 9, 180
disaster recovery (DR) mechanisms, 28, 214
disk space, 80, 235, 239
distributed architecture, 51, 61, 72, 89, 135–136, 165–166
durability discovery stress testing, 232–233

e
end-users
acceptance testing by, 32
as volatile business domain, 158
enterprise architecture
artifact studies for, 236–238
capacity planning for (see capacity planning)
decomposition of (see architecture decomposition)
design flaws in, 189, 190, 191, 194, 199, 232
design of (see architecture design)
design substantiation for (see design substantiation)
enterprise architecture (Continued)
definitions for, 3–4, 36
evolution of, 8, 118, 143–144
monolithic, 8, 58, 71, 76, 91–95, 165
patterns for, 89–90
perspectives of, 86–91, 145
pressure points in (see pressure points)
segmentation of, 176–178, 184–185, 189–195
technical analysis case study, 120–122
unified business categorization of, 114–117
verification of (see architecture verification)
voylarity in, 156–160
enterprise architecture discovery/analysis process
application discovery in, 55–66
function of, 35–36
prerequisites for, 36–37
scenarios for, 36
system fabric patterns in, 39–53
enterprise architecture-level system failures, 17, 23–29
enterprise-level design, 86–87, 88–91
enterprise-level separation of concerns, 124
text service bus (ESB)
and application dependencies, 75, 77
and architecture expansion, 27
in cloud computing, 125
in federated architecture, 99, 100, 203
function of, 3
as pressure point, 214
in system-level behavioral decomposition, 172
in technical analysis, 123–124
See also ESB chaining
environment
behavior of, 162
definition of, 84
pressure sources in, 199, 223
See also production environment
environment behavior decomposition model, 163–164, 175
environment volatility model, 158–160
ESB chaining, 124, 126, 131–132
external application dependencies, 74–76
F
fabric patterns. See system fabric
failure (of systems). See system failure
family fabric pattern, 44–46, 77, 206, 207, 210–211
features, in system-level behavioral decomposition, 170
federated business domain, 153–154
Ford Motor Co. system failure example, 15
forecast application performance, 63
functional decomposition. See behavioral decomposition
functionality externalization, 76
functions, in system-level behavioral decomposition, 170
G
gateways
and architecture expansion, 27
as architecture type, 90
in federated architecture, 99
function of, 75
as pressure point, 214
in system-level behavioral decomposition, 172
in technical analysis, 126–127
glue proxies, 216
grand-scale architecture
appropriateness of, 197
and capacity planning, 237, 244, 250
classification and decomposition, 116–117, 138–144, 162
design substantiation analysis, 191
pressure points in, 224, 232
See also enterprise architecture
granularity, 61
H
hardware
budgeting for, 62
capacity planning for, 235
cataloging of, 77
in technology analysis, 123
in topology map, 67
high-availability (HA) testing, 63
I
identity server, 125, 142
incremental software architecture approach
and architecture decomposition, 83, 148
and capacity planning, 237
and discovery process, 55
function of, 2, 3, 8
process steps, 9
verification process in, 179
vs. software modeling/simulation, 7
information technology (IT)
executive accountability in, 18
risk assessment role of, 20
volatility of, 158
infrastructure. See network infrastructure
infrastructure as service (IaS), 126
integrated development environment (IDE), 30, 31
integration
of assets, 203, 215, 224
of systems, 24, 28, 33
integration testing, 31–32
integration verification, 185
interfaces, in application architecture, 60
intermediary installations
in broker-based architectures, 90
excessive, 27, 28, 123, 203
function of, 75
as pressure points, 211, 214, 218
intermediary nodes, 50–51
internal application dependencies, 76
International Standards Organization (ISO), 80
interoperability
and application dependencies, 71
and architecture expansion, 27
and ESBs, 123
and federated architecture, 100–101, 129–131, 202
and gateway architecture, 90, 99, 126
and message transport fabric pattern, 44
and pressure points, 214
and system failure, 24, 29
iterative software development, 182–183
K
key performance indicators (KPIs), 69
keystone architecture structure classification and
decomposition diagram, 91–102
in sub-architecture classification, 108–114, 129,
132–133, 136–137
L
legacy systems, 112, 189
linear business transactions, 49
load testing, 229
logical application dependencies, 70
loose coupling, 60, 68, 71, 75, 83, 89, 96, 99,
110, 165, 175
M
management-level system failures, 17, 18–19
market segmentation analysis, 19, 37
memory utilization, 62, 80, 216, 235,
239, 250
mesh architecture, 40, 77, 91–93, 95–97, 108,
110–112, 115, 117, 118, 142
message augmentation, 51
message injection, 229
message paths
and application dependencies, 75
and architecture structure, 202
and architecture type, 89–90
congestion of, 198, 199, 224
mapping routes of, 67
stress testing of, 197, 229, 231, 232–233
in sub-architecture discovery, 104
in system-level behavioral decomposition, 172
in technical profiling, 128
types of, 39–53
message transport fabric pattern, 43–44, 51,
52, 77, 206
messaging infrastructure, 60
Micro-Organizations, 1, 2
microservices, 8
middleware
application dependency on, 75–76
in application discovery process, 56
budgeting for, 62
cataloging of, 77, 80–81
as pressure point, 218
in system-level behavioral decomposition, 170,
172, 173
in technical analysis, 119, 123
in topology map, 67
mobile applications, 57
model-view-controller application architecture
(MVC), 61, 88
money transfer application, 105, 106, 112–114, 116
monolithic architecture, 8, 58, 71, 76, 91–95, 108,
112–116, 118, 142, 165
N
network infrastructure
application dependency on, 75–76
in application discovery process, 56, 63–64
budgeting for, 62
congestion in, 24, 25, 40, 165, 198, 199, 211,
223, 224, 229
in system-level behavioral decomposition, 170
in technical analysis, 119, 123
in topology map, 67
upgrades to, 33, 34
network layer, 80
network nodes
and application dependency patterns, 77
in bridge fabric pattern, 50–51, 52, 53
in chained message route fabric pattern, 48–50
in circular message route fabric pattern, 46–48
in community center fabric pattern, 41–43
in community fabric pattern, 40–41, 43, 52
in compound fabric patterns, 51–53
in crossroads fabric pattern, 50–51, 52, 53
in family fabric pattern, 44–46
in message transport fabric pattern, 43–44, 52
network performance/availability, 24
network topology
mesh in, 40
vs. network fabric, 39
and system failure detection, 24
nonfunctional requirements (NFR) document, 62,
237, 248
N-tier application architecture, 61, 68–69,
88, 202
O
observer patron, in design substantiation, 188, 194
off-the-shelf applications, 57, 167
one-way message exchange, 73
open-source products, 8
open systems interconnection, 80
operation-level system failures, 17, 33–34
operations, volatility of, 157
organizational change, 7–9
See also business volatility
organizational culture, 29
organizational structure, 1, 8, 154
See also business domains
orphan nodes, 41, 42, 51

P
pattern matching, 128
performance measurement
for applications, 61
and application topology mapping, 69
and capacity planning, 63
and stress testing, 197
technologies for, 239–240
physical application dependencies, 70, 238
physical layer, 80
platform as service (PaS), 126
point-to-point architecture, 25, 52, 77, 90, 96,
110–112, 136, 142
power blackout (2003), 16, 29
presentation layer, 59, 60, 61, 159
pressure points
behavioral, 200, 214–218, 225–226, 228
data collection from, 236, 238–240, 241,
244–250
environmental vs. architectural, 198–201
failure conditions for, 227–229
inserting, 223–227
joint, 203, 211–214
and stress testing, 197
structural, 200, 201–214, 224, 225, 228–229
volatility, 200, 218–222, 225, 227, 228
private application dependencies, 72–73
processes
vs. activities, 166, 168
in design substantiation, 184
as pressure points, 214
in system-level behavior decomposition,
170–172
product description, 37
product development, 19–20
production
and continuous application discovery, 56
and ensuring business continuity, 33
production environment
mapping application integration in, 67, 68
multiple systems in, 10, 11–12
pressure points in, 198–199, 202–203, 211, 214,
215–216
as profitability center, 219
product life cycle
and business volatility, 218
and system failure, 13, 23
product segmentation analysis, 20, 37
proxies, 75, 79, 90, 99, 214, 216
public application dependencies, 73–74

Q
quality of services (QoS), 70

R
RACI chart, 186, 195–196
recovery discovery stress testing, 230–231
release management (RM), 65–66
documents, 34
request-response, 72
response time
application, 70, 75
and excessive message mediators, 203
system, 25, 34
and system-level behavior, 164
return on investment (ROI), 145
reusability, 61
rich Internet application (RIA), 57, 76
risk management
and architecture discovery/analysis process,
37
and pressure points, 227–229, 244
and system failure, 20
routers
application dependency on, 75
configurations, 39, 44
and network congestion, 199
as network infrastructure, 170

S
security breaches, 34
security platform, 60, 63, 157, 170, 224
segmentation analysis, 19–20
service applications, 57
service layer, 60
service-level agreements (SLAs)
and application capacity, 61
and architecture capacity, 248
and end-state architecture design, 2
KPIs for, 70
and system testing, 32
service-oriented application architecture (SOA), 61,
89
single point of failure, 45, 48, 49, 98, 210
software
capacity planning for, 235
cataloging of, 77
construction of, 6, 83, 181–182
to detect technological system failures, 24
elasticity of, 71
establishing baseline for, 65
reuse of, 165, 169
in technical analysis, 123
upgrades to, 33
software architecture design
delays in, 184
vs. development, 3, 6
substantiation of (see design substantiation)
and system failures, 23–29
software architecture development
vs. design, 3, 6
duration of, 192
scope of, 191
stakeholders in, 188
system failures in, 17, 23, 29–31
testing in, 240
time/budget constraints in, 8
and verification, 83
waterfall vs. iterative approaches to, 182–183
software as service (SaS), 126
software asset distribution, 24, 28
software cataloging, 24
software configuration management (SCM), 65
software development life cycle (SDLC), 65
software engineer, in design substantiation, 187, 193, 194
software entropy, 183, 192
software modeling, 6–7, 30
software module testing, 192, 193
software simulation, 7, 240
source code
defects in, 13, 16, 29
design of, 3, 6, 65, 179, 194
in design substantiation, 186
in performance measurement technologies, 240
and system-level behavior, 164
testing of, 31, 192
star network topology, 42, 77, 90, 97, 108, 131, 135, 141
statelessness, 61
stress testing
function of, 6, 197, 199
methodology for, 227–233
and pressure points, 207, 210, 214, 218, 223–227
as verification process step, 9, 180
structural pressure points, 200, 201–214, 224, 225, 228–229, 244
structural signature, 202
sub-architectures
definition of, 102
discovery, 103–107, 120, 122–128, 140
overlapping, 138, 140
structural pressures of, 211–214
in technical analysis, 119, 120
in unified business categorization, 115–117, 138–144
system(s)
cascading behaviors in, 164, 212, 215–216
definition of, 9–10, 170
deployment of, 33
federated, 24, 28–29
maintenance of, 33, 34
scalability of, 24, 28, 235
system fabric
and application dependencies, 76–77
and capacity planning, 237
definition of, 39
pattern types, 40–53, 202
and structural pressures, 206–211
in technical analysis, 119, 120
tools for analyzing, 56
system failure
and circular message route fabric pattern, 48
classification of, 16–21
definition of, 13
and environment-level behaviors, 164–165, 199
elements of, 14–16
factors in, 11–12
operation-level, 33–34
software development-level, 29–31
technological, 23–29
testing-level, 31–32
system integration, 24, 28, 33
system-level behavioral decomposition, 170–172
system-level pressure points, 215–216, 224
system load testing, 32
system of systems, 23
system testing, 32
T
technical analysis
case study, 120–122
function of, 102, 119
profiling in, 123–128
sub-architecture discovery in, 120, 128–138
technical capability, 119, 120, 140–141, 221
technical profiling, 120, 122–128, 138
technical specifications document, 237
technical views, 120, 138
vs. business views, 145
technological system-level failures, 17
detecting, 24–25, 28–29
elements of, 26–27
types of, 23–24
technological volatility, 156–157, 218, 221–222
technology development, 7
testing
in capacity planning, 63
in design substantiation, 192–193
pre-production application readiness, 63–64
in software development, 240
types, 31–32
See also stress testing
testing-level system failures, 17, 31–32
thick client applications, 57
tight coupling, 71, 76, 89, 94, 96, 112–114, 165
top-down change initiatives, 7
topology mapping, 67–70, 238
transport layer, 80
two-way message exchange, 72–73

U
unit testing, 31, 192, 193
user interface
  in application architecture, 59
  role in system failure, 21
user provisioning (administration), 34, 60

V
volatility pressure points, 200, 218–222, 225, 227, 228, 244

W
waterfall software development, 182
Web applications, 57, 69
workload management, 98