

Index

SYMBOLS

* (asterisks)

- as mandatory field indicator, 145
- as standard indicator, 178

A

acceptance testing

- defined, 230–232
- in production environments, 239
- technical, 233

access

- design considerations, 154
- page specifications, 180

access, data. *See* data access

accessibility

- considerations, 138
- design considerations, 149–150
- design overview, 145–149
- in quality landscape, 38

Account Manager service, 415

accounts

- application modeling. *See* application modeling
- event types, 573–574
- HFRs (High-Level Functional Requirements).
 See HFRs (High-Level Functional Requirements)
- My Account page, 336–337
- online application walkthrough, 331–335
- partitioning based on, 520
- planning architecture. *See* architecture planning
- Welcome page, 335–336

accounts, storyboard

- accounts, 339–342, 344–345
- activation, 339–342
- changing passwords, 346–348
- closing, 349–350
- creating new, 337–339
- logging in, 344–345

logging out, 345

re-requesting activation key, 342–344

re-requesting key, 342–344

resetting passwords, 345–346

updates, 348–349

accounts, use case

activation, 375–377

changing passwords, 384–387

closing, 390–392

creating new, 372–374

logging in, 379–381

logging out, 381–382

re-requesting activation key, 377–379

resetting passwords, 382–383

updating, 388–390

acquiring objects, 526–527

actions

page specifications, 180

providing audit trail, 510

in use cases, 166–173

Activate Account page

functional requirements, 359–361

modeling flow, 450–452

modeling inputs and outputs, 444

storyboard, 339–342

use case, 375–377

activation

account, 332

Activation service, 415

component responsibility modeling, 435–439

new key processing, 460

reporting components, 463–465

re-requesting activation key. *See* Re-Request

 Activation Key page

 sending emails, 424–425

 Welcome page, 335–336

active nodes, 209–210

active-active configurations, 226–227, 238–239

active-passive configurations, 223–226, 238–239

activities

- defect contents, 123
- defining core, 98–100
- developer testing, 622–628
- production preparation, 82–83
- severity classifications, 122
- tracing based on, 588–590

activities and environments, 229–242

- acceptance testing, 231–232
- functional testing, 235–236
- overview, 229–230
- regression testing, 236–238
- summary, 241–242
- technical testing, 232–235
- testing in production environment, 238–241
- testing types, 230

actors, 166–173

actual results

- comparison tools, 106
- testing, 625

actual time, 124

Adams, Douglas, 142

adaptability, 39

adapters

- in architecture, 183
- system framework, 403–405

ad-hoc batch, 293

administration

- batch jobs, 294
- in Operations Manual, 254
- Service Delivery and Operations. *See* Service Delivery and Operations

administration servers

- configuration, 205
- defined, 202
- Installation and Configuration Guide, 259–260
- performance counters, 565
- separating components, 640

Administrator's Guide, 141

aged accounts

- batch jobs, 461–463
- functional requirements, 367–369
- reporting components, 463–465

agents, 219–220

aggregation, counter, 567

alerts

- defined, 273–274
- incident investigation, 310
- incident verification and resolution, 312–313

- monitoring and. *See* monitoring and alerts
- in Operations Manual, 254
- role of Service Delivery team, 250–251

algorithms for load balancing, 206–207

allowable actions

- functional requirements, 351–352
- in use case diagram, 168

allowable characters, 181

allowable values, 182

ALT tags, 150

analytics

- defined, 284–285
- in Developer's Guide, 324
- fitting for purpose, 13
- impact analysis. *See* impact analysis
- in Operations Manual, 254
- outputs. *See* outputs, analysis and design

antivirus (AV). *See* AV (antivirus)

application maintenance, 317–326

- defined, 650–655
- designing for. *See* designing for application maintenance
- Developer's Guide, 324–325
- functions, 319–324
- incident investigation, 309
- overview, 317–318
- role of Service Delivery team, 248–252
- scope of support, 245–246
- summary, 326
- team, 318–319
- three levels of live service support, 245–246

application modeling, 433–466

- Activate Account flow, 450–452
- adding inputs and outputs, 439–444
- batch jobs, 455–463
- Change Password flow, 453–454
- Close Account flow, 455
- component responsibility, 434–439
- Login flow, 445–449
- Logout flow, 449–450
- overview, 433–434
- repeating patterns, 445
- reporting components, 463–465
- Re-Request Activation Key flow, 452–453
- Reset Password flow, 453
- Update Account flow, 454–455

application servers

- in batch architecture, 470–472
- configuration, 204–205

- defined, 202
 - Installation and Configuration Guide, 259
 - performance counters, 565
 - separating components, 640
 - applications**
 - component specifications, 182–184
 - in development scope, 92
 - exceptions, 548–549
 - fit for purpose, 10–14
 - framework, 399–402
 - guides, 22
 - maintenance environments, 18
 - maintenance processes and tools, 20
 - software topology, 213–215
 - tuning, 523–529
 - approved defects, 115**
 - architecture**
 - analysis and design, 182–184
 - application modeling. *See* application modeling
 - blueprints and deployment diagrams, 96
 - build, release and regular integration
 - mobilization, 103
 - component lists and specifications, 184–185
 - conceptual. *See* conceptual architecture
 - designing for batch, 470–472
 - interactions and exceptions, 189–192
 - logging and diagnostics, 187–189
 - master-agent, 219–220
 - message and interface mappings, 185–186
 - Operations Manual, 254–259
 - production environment, 198–200
 - reporting, 281–282
 - test mobilization, 106
 - architecture planning, 395–431**
 - batch processes, 424–425
 - business framework components, 422–423
 - business processing components, 416–422
 - common application framework, 399–402
 - conceptual, 396–397
 - high-level transaction flow, 408–411
 - logical component model, 428–430
 - logical data models and data access, 426–428
 - logical layers, 397–399
 - overview, 395–396
 - presentation tier, 411–414
 - production deployment concept, 407–408
 - summary, 430–431
 - system framework, 403–405
 - tooling framework, 406–407
 - web services facade, 414–416
 - archiving**
 - batch jobs, 291
 - generic batch jobs, 498
 - assembly testing, 6, 52–53**
 - atomic component tests, 232**
 - audio content accessibility, 150**
 - auditing**
 - design considerations, 154
 - event types, 573
 - providing trail, 509–515
 - authentication**
 - Activate Account flow, 450–452
 - defined, 415
 - event types, 573
 - Login page, 447–449
 - password. *See* passwords
 - Reset Password flow, 453
 - authorization, 115. *See also* passwords**
 - auto-generated values, 182**
 - automated tests**
 - impact of change, 628
 - pros vs. cons, 71
 - automatic failure and recovery, 258–259**
 - automatic restarting, 557**
 - automatic updates, 216–217**
 - automation**
 - build, 16
 - code profiling, 49
 - AV (antivirus)**
 - batch jobs, 293
 - role of Service Delivery team, 250
 - software topology, 216
 - availability**
 - in quality landscape, 38
 - requirements, 165
 - AverageTimer, 560**
- ## B
- Back button, 533–537**
 - back of the box, 139–140**
 - back-end firewalls, 201**
 - backup strategies. *See also* recovery strategies**
 - fitting for purpose, 13–14
 - servers, 202
 - balancing, load. *See* load balancing**

batch, 289–303

- administration, 249
- defining specifications, 95
- design for. *See* designing for batch
- in development scope, 92
- event types, 574
- fitting for purpose, 12–13
- functional requirements, 367–369
- modeling jobs, 455–463
- in Operations Manual, 254
- performance counters, 564
- processes, 398, 424–425
- processing, 289–295
- providing audit trail, 512–515
- role of Service Delivery team, 251
- run date, 302
- scheduler, 295–301
- servers, 203
- summary, 303
- testing requirements, 627

`BatchJobFactory` **class, 474–476**

best practices

- analysis of, 66–71
- code quality, 46–47
- cost-benefit analysis, 65–66
- defining, 97–98

black-box testing, 52

blackout windows, 269–270

blueprints, architecture, 96

bottom-up approach, 52

branching

- application maintenance, 323
- configuration, 131–132

browsers

- design considerations, 157
- user behavior, 533–537

budgets, 27–31. *See also* costs

bugs. *See* defects

build testing

- code, 46–48
- code coverage, 50
- code profiling and peer reviews, 48–49
- documentation generation, 51–52
- test processes, 43–46
- unit testing, 50–51

builds

- configuration management, 127
- environment, 16
- improving quality, 59–60
- mobilization, 80, 102–105

- regression testing and, 237–238

bulk operations

- optimizing queries, 522
- tuning application layer, 528

business

- framework components, 422–423
- intelligence, 275
- processing components, 398, 416–422
- reporting, 428
- reports, 276–281

By column, 509

C

c tag, 602

C# testing requirements, 625–627

CA (Certificate Authority), 150

caching

- application, 527
- Change Password flow, 454
- tuning database layer with, 522–523

calculated values, 182

calendars. *See also* dates

- in application framework, 399
- batch run date, 302
- custom, 158–159

case sensitivity, 181

case study

- application modeling. *See* application modeling
- conceptual design. *See* conceptual design
- planning architecture. *See* architecture planning

Certificate Authority (CA), 150

change control

- defined, 7
- mobilization, 80, 132–133
- processes and tools, 20
- role of Service Delivery team, 250

Change Password page

- functional requirements, 362–363
- Login flow, 449
- modeling flow, 453–454
- storyboard, 346–348
- use case, 384–387

changes

- application maintenance, 322–323
- commenting standards, 606

characters

- allowable, 181
- generating enumerators, 504–505

- password requirements, 353
- validating and encoding user input, 537–538
- check-pointing**
 - batch job execution flow, 480
 - multi-threaded jobs, 491
- classes**
 - architecture, 184
 - BatchJobFactory, 474–476
 - coding standards, 609–610
 - coupling, 655
 - Exception, 548–550
 - model, 427
 - testing requirements, 625–627
 - tracing names, 400
 - transaction flow, 408–411
- classifications**
 - defining defects, 119–122
 - releases, 641–644
- cleanup**
 - batch jobs, 461–463
 - design considerations, 154
 - exception handling, 555–556
- client-side**
 - software topology, 219
 - state management, 176–177
- Close Account page**
 - email batch processing, 461
 - functional requirements, 365–366
 - modeling flow, 455
 - storyboard, 349–350
 - use case, 390–392
- closed accounts**
 - batch jobs, 463
 - functional requirements, 367–369
 - reporting components, 463–465
- closed defects, 118–119**
- clustering**
 - defined, 209–210
 - exception handling, 552
 - generic batch jobs, 498
 - monitoring, 271
- code**
 - audit records, 514
 - best practices analysis, 69
 - coverage, 50
 - improving quality, 60
 - project settings, 599–600
 - quality, 46–48
 - standards and practices, 609–610
 - tracing. *See* tracing
 - updates, 320
 - using enumerators in, 503–505
- code profiling**
 - code metrics profilers, 654–655
 - in Developer's Guide, 324
 - peer reviews and, 48–49
- code **tag, 602**
- collision domains, 213**
- colors and accessibility, 150**
- commands. *See* methods**
- comments**
 - defining templates and exemplars, 101
 - documentation generation, 51–52
 - pros vs. cons, 69
 - standards, 603–608
- committed data, 156**
- common application framework, 399–402**
- communication**
 - constraints, 88
 - in Deployment Guide, 257–258
 - impact analysis, 115–116
 - in web services facade, 414–416
- comparison, data**
 - generic batch jobs, 498
 - tools, 627–628
- completeness in quality landscape, 38**
- completion reports**
 - build and unit test process, 45–46
 - defining core activities, 99
 - test mobilization, 108
 - user training, 22
 - wrapping up development, 101
- components**
 - analysis and design specifications, 182–184
 - atomic tests, 232
 - based logging, 189
 - business framework, 422–423
 - business processing, 416–422
 - defining core activities, 98–100
 - defining inputs, 94
 - developer testing, 622–623
 - development and review, 44–45
 - in development scope, 91–92
 - estimating time and costs, 71–75
 - lists and specifications, 184–185
 - logical models, 428–430
 - production deployment, 407–408
 - in production environment, 201–203
 - responsibility modeling, 434–439
 - reusability, 47

components (*continued*)

- separating, 639–640
- software topology, 213–215
- transaction flow, 408–411
- use case flow, 172–173

ComposeAndSend **method**, 459

composite fields, 182

conceptual architecture

- common application framework, 399–402
- defined, 396–397
- developer testing, 622
- high-level transaction flow, 408–411
- logical layers, 397–399
- production deployment concept, 407–408
- system framework, 403–405
- tooling framework, 406–407

conceptual design, 329–393

- batch requirements, 367–369
- data element requirements, 352–356
- defining functional requirements, 350–352
- email processing requirements, 366
- guiding principles, 331
- My Account page, 336–337
- online application walkthrough, 331–335
- online user interface and processing requirements, 356–366
- overview, 329–330
- reporting requirements, 369–370
- SB-001 Create New Account, 337–339
- SB-002 Activate Account, 339–342
- SB-003 Re-Request Activation Key, 342–344
- SB-004 Login, 344–345
- SB-005 Logout, 345
- SB-006 Request Password, 345–346
- SB-007 Change Password, 346–348
- SB-008 Update Account, 348–349
- SB-009 Close Account, 349–350
- summary, 392–393
- technical requirements, 370–372
- UC-001 Create New Account, 372–374
- UC-002 Activate Account, 375–377
- UC-002 Re-Request Activation Key, 377–379
- UC-004 Login, 379–381
- UC-005 Logout, 381–382
- UC-006 Reset Password, 382–383
- UC-007 Change Password, 384–387
- UC-008 Update Account, 388–390
- UC-009 Close Account, 390–392
- Welcome page, 335–336

concurrency, 155

conditions

- code coverage, 50
- coding standards, 609–610
- unit testing, 51

configurability

- application-specific settings, 611–615
- messages, 617–618
- requirements, 166
- static page elements, 615–616
- support for, 610–611
- validation rules, 616–617

configuration

- application-specific settings, 611–615
- commenting standards, 607–608
- event types, 574
- factory pattern, 632
- files and settings, 593
- guides, 97
- individual server, 203–205
- monitoring, 265–268
- monitoring design, 576–578
- in Operations Manual, 254
- in system framework, 405
- tracing, 591–592
- updates, 320–321
- values, 92

configuration management

- application maintenance, 323–324
- application maintenance functions, 319
- changes, 532–533
- code quality, 48
- in Developer's Guide, 325
- fit for purpose environments, 16
- mobilization, 80, 125–132
- processes and tools, 19
- production readiness, 7
- in quality landscape, 39

connections

- counters, 563
- custom performance counters, 272
- load balancing, 206–207
- in web services facade, 414–416

constraints

- analysis and tracking, 18
- build, release and regular integration mobilization, 102
- change control mobilization, 133
- configuration management, 125–132
- defect management, 110–111

- production, 83–90
- project management, 31
- test mobilization, 105
- construction, 137–194**
 - accessibility considerations, 149–150
 - architecture and application component specifications, 182–184
 - component lists and specifications, 184–185
 - custom built vs. off-the-shelf, 152–153
 - data dictionary, 180–182
 - defining inputs, 93–96
 - design considerations, 141–148, 154–159
 - implementation considerations, 150–152
 - interactions and exceptions, 189–192
 - logging and diagnostics, 187–189
 - message and interface mappings, 185–186
 - navigation maps, 173–174
 - overview, 137–138
 - page layouts and specifications, 178–180
 - quality landscape process, 41–43
 - requirements output, 161–166
 - scope outputs, 160–161
 - storyboards, 174–177
 - summary, 192–194
 - support and maintenance considerations, 152
 - thinking out-of-the-box, 139–141
 - use case diagrams and use cases, 166–173
- contents**
 - configuration management, 126–127
 - defect management, 123–124
 - thinking out-of-the-box, 139–141
- context, transaction, 411**
- contextual elements, 618**
- continuous integration, 6**
- control**
 - batch job, 472–474
 - batch job execution flow, 476–482
 - batch processes, 486–488
 - BatchJobFactory class, 474–476
 - change. See change control
 - generic batch jobs, 498
 - operations, 247
 - progress reporting, 482–486
 - user behavior, 533–537
- cookies, 176–177**
- COPQ (cost of poor quality), 64–66**
- copy text, 151**
- copyright statements, 144**
- core activities, 98–100**
- correctness in quality landscape, 38**
- cost of poor quality (COPQ), 64–66**
- costs**
 - conceptual design, 331
 - cost-benefit analysis of quality landscape, 65–71
 - custom built vs. off-the-shelf, 152–153
 - of defects, 61–64
 - effectives and quality, 37
 - financial analysis of quality landscape, 64–65
 - maintenance considerations, 152
 - quality and estimates, 71–75
 - of quality landscape, 61
- counters, performance**
 - configuration, 576–577
 - good, bad and ugly, 562–563
 - incident investigation, 310
 - intermittent counter updates, 568–569
 - monitoring, 270–272
 - monitoring events, 265–268
 - overview, 560–562
 - in system framework, 405
 - thresholds and tolerances, 569–570
 - what, when and why, 563–568
- counts, transaction, 507–508**
- covered builds, 59**
- Create New Account page**
 - functional requirements, 357–359
 - modeling inputs and outputs, 439–444
 - storyboard, 337–339
 - use case, 372–374
- criteria**
 - code coverage, 50
 - multiple batch jobs, 493–495
 - production readiness. See production readiness reporting components, 463–465
- Critical classification, 119–122**
- cryptography**
 - event types, 575
 - stubs and simulators, 634
 - testing requirements, 626
- custom built**
 - conceptual design, 331
 - design considerations, 138, 148
 - vs. off-the-shelf, 152–153
- custom calendars**
 - in common application framework, 402
 - design considerations, 158–159
- custom performance counters, 272**
- custom test tools, 106**

custom timeout values, 547–548

customer service, 152

cyclomatic complexity, 49, 655

D

daily batch, 291

daily routine, 99–100

data

- centers, 220–227
- defects and testing, 54
- defining master sets, 509
- elements, 352–356
- generic batch jobs, 498
- improving quality, 57–58
- in logical layers, 399
- meaningful, 501–509
- monitoring, 265–268
- partitioning tables and, 520–521
- protecting against resubmission of, 540–543
- stores, 12
- updates, 320

data access

- event types, 575
- logical models and, 426–428
- testing, 623

data dictionaries

- analysis and design, 180–182
- entity definitions, 178

data entry forms, 356–357

data models

- logical, 426–428
- specifications, 95

database scripts, 127–131

database servers

- in batch architecture, 470–472
- configuration, 205
- defined, 202–203
- Installation and Configuration Guide, 260

databases

- administration, 248–249
- batch jobs, 290–294
- components and scope, 91
- design considerations, 156
- improving quality, 57–58
- incident investigation, 311
- lookup tables, 502–505
- objects, 624–625
- providing audit trail, 509–515
- reporting architecture, 281–282

storing configuration values in, 614–615

storing state, 177

tuning, 523–529

updates, 320

dates

- batch jobs, 292
- batch run, 302
- component responsibility modeling, 437–439
- entries and validation, 157–159
- indexing tables, 519
- monitoring events, 267
- multiple batch jobs, 493
- in production environment, 200
- providing audit trail, 510
- recording, 505–506

deadlocks, 156

debug traces, 583

default behaviors

- exception handling, 556
- system framework, 405

default selections, 179

default values, 182

defect management

- application maintenance functions, 319
- constraints, 110–111
- contents, 123–124
- defined, 7
- fixes and quality, 54–56
- importance of quality, 40–41
- incident investigation. *See* incident investigation
- mobilization, 80
- processes and tools, 19–20
- reports, 124
- testing and, 53–54
- workflow, 111–119

defects

- costs of, 61–64
- severity and priority, 119–122
- test mobilization, 108
- testing. *See also* testing

deferred defects

- contents, 123
- defined, 117

defragmentation, 293

deletion

- audit records, 515
- generic batch jobs, 498

delivery

- mobilization, 81–82
- production readiness, 26–27

Demilitarized Zone (DMZ), 200–201**dependencies**

- batch jobs, 297
- batch scheduling, 298–299

deployment

- architecture, 96
- Deployment Guide, 254–259
- design considerations, 151
- designing for, 637–645
- in development scope, 92
- fitting for purpose, 14
- processes and tools, 19
- production architecture, 407–408
- production readiness, 6
- in quality landscape, 39
- role of Service Delivery team, 249
- servers, 203
- software topology, 219

dequeuing, 456**descriptions**

- defect contents, 123
- incident investigation, 309
- monitoring events, 267

design

- application maintenance, 653
- application modeling. *See* application modeling
- batch jobs, 294
- conceptual. *See* conceptual design
- fit for purpose, 15
- functional testing, 235–236
- outputs. *See* outputs, analysis and design
- planning architecture. *See* architecture planning
- processes and tools, 18
- production line activities, 82–83
- production readiness, 6
- reports, 280
- technical, 91
- validation, 43

design considerations

- accessibility, 149–150
- construction, 141–148
- custom built vs. off-the-shelf, 152–153
- defined, 138
- general, 154–159
- implementation, 150–152
- support and maintenance, 152

designing for application maintenance, 595–620

- application-specific configuration settings, 611–615
- coding standards and practices, 609–610

- commenting standards, 603–608
- defining standards, 600–601
- messages, 617–618
- overview, 595
- preparing solution, 596–600
- static page elements, 615–616
- summary, 618–620
- supporting configurability, 610–611
- validation rules, 616–617
- XML documentation comments, 601–602

designing for batch, 469–500

- architecture, 470–472
- batch job execution flow, 476–482
- batch job parameters, 495–497
- `BatchJobFactory` class, 474–476
- controlling processes, 486–488
- execution and control framework, 472–474
- generic jobs and functionality, 497–499
- multiple jobs, 492–495
- multiple threads and jobs, 488–489
- multi-threaded jobs, 489–492
- overview, 469
- progress reporting, 482–486
- summary, 499–500

designing for deployment, 637–645**designing for incident investigation, 579–594**

- activity-based tracing, 588–590
- investigative tooling, 593
- multiple trace files, 590–591
- overview, 579
- summary, 594
- tracing, 580–588
- tracing configuration, 591–592

designing for monitoring, 559–578

- configuration, 576–578
- counters, good, bad and ugly, 562–563
- counters, what, when and why, 563–568
- events and event logs, 570–576
- intermittent counter updates, 568–569
- overview, 559–560
- performance counters, 560–562
- summary, 578
- thresholds and tolerances, 569–570

designing for performance, 517–530

- overview, 517–518
- summary, 529–530
- tuning application layer, 523–529
- tuning database layer, 518–523

designing for reporting, 501–516

- audit trail, 509–515
- meaningful data, 501–509
- overview, 501
- summary, 515–516

designing for resilience, 531–558

- exception handling, adding value, 554–556
- exception handling flow, 550–554
- exception handling planning, 544–548
- exception handling strategy, 548–550
- exceptions lead to operational procedures, 556–557
- handling system failure, 543–544
- managing configuration changes, 532–533
- non-stop processing, 557
- overview, 531
- protecting against resubmission of data, 540–543
- protecting against URL hacking, 538–540
- summary, 558
- supporting and controlling user behavior, 533–537
- validating and encoding user input, 537–538

designing for testing, 621–636

- developer activities, 622–628
- factory pattern, 628–633
- overview, 621–622
- stubs and simulators, 633–635
- summary, 635–636

destination IP addresses, 207

developers

- Developer's Guide, 324–325
- testing activities, 622–628

development

- build and unit test process, 44–45
- code standards, 47
- configuration management, 127
- costs, 64–65
- defects effect on, 53–54
- design considerations, 150–151
- environments, 15–16
- estimating time and costs, 71–75
- mobilization, 80
- processes and tools, 18–19
- production readiness. *See* production readiness

development mobilization

- inputs, 93–96
- process, 96–100
- scope, 90–93
- templates and exemplars, 100–101

- wrapping up, 101–102

diagnostics. *See also* monitoring

- analysis and design, 187–189
- code quality, 48
- in common application framework, 402
- designing for incident investigation. *See* designing for incident investigation
- in system framework, 403–405
- use case flow, 172

diagrams. *See also* models

- interactions and exceptions, 190
- online application walkthrough, 333
- page layouts and specifications, 178–180
- production environment, 199

directories, batch job, 297

disabilities, 149–150

disaster recovery (DR). *See* DR (disaster recovery)

disks

- defragmentation, 293
- monitoring availability, 271

DMZ (Demilitarized Zone), 200–201

DNS (domain name system) load balancing, 209

documentation

- account use case. *See* accounts, use case
- batch jobs, 293
- best practices, 97–98
- class coupling, 655
- commenting standards, 603–608
- configuration management, 127
- core activities, 99–100
- in development scope, 92
- exceptions lead to procedures, 556–557
- Operations Manual. *See* Operations Manual
- role of Service Delivery team, 250
- software topology, 216
- user training, 22–23
- wrapping up development, 101–102
- XML comments, 601–602

documentation generation

- cost-benefit analysis of quality landscape, 65–66
- improving quality with, 60
- pros vs. cons, 71
- quality, 51–52
- regression testing and, 238

domain name system (DNS) load balancing, 209

downstream filtering, 268

DR (disaster recovery)

- batch jobs, 293

maintenance and administration, 260
 primary and secondary data centers, 220–227
 sites, 211

dual-site environments, 211

dynamic code profiling

defined, 49
 regression testing and, 238

E

education. See training

efficiency

conceptual design, 331
 in quality landscape, 37–38
 requirements, 165

email

batch processes, 424–425
 component responsibility modeling, 435–439
 configurable messages, 617–618
 constraints, 88
 event types, 575
 functional requirements, 353–354
 logical data models, 426–428
 lookup tables, 502–505
 modeling inputs and outputs, 439–444
 processing requirements, 366
 servers, 201
 solicited, batch jobs, 455–461

encoding user input, 537–538

encryption

design considerations, 156
 email address, 354
 fitting for purpose, 12

end-points, 633

end-to-end performance

counters, 562
 multiple batch jobs, 493
 reports, 277–278

end-to-end transaction flow, 408–411

enhancements, application maintenance, 322–323

entities

data, 399, 427
 data dictionary, 181
 defining, 178
 factory pattern, 631
 testing requirements, 625

entries, log. See logging

entry criteria, 107–108

entry permutations

code coverage, 50
 integration testing, 53
 unit testing, 51

enumerators

application table, 615
 using in lookup tables, 503–505

environments

activities and. *See* activities and environments
 cleanup/teardown, 555–556
 configuration management structure, 127–131
 constraints, 88–89
 defects and testing, 54
 fit for purpose, 14–18
 improving quality, 57–58
 production. *See* production environments
 updates, 321
 user training, 22

errors

activation, 341–342
 architecture, 184
 designing for resilience. *See* designing for resilience
 event properties, 267
 events, 572–573
 logging and diagnostics, 188
 messages, 617–618
 project settings, 598–599
 system error messages, 179
 tracing, 583
 use case flow, 172
 validation, 339
 validation rules, 616–617

estimated time to impact

defect contents, 124
 defined, 115

estimates

design considerations, 147
 quality landscape, 71–75

event logs

investigative tooling, 593
 overview, 403–405

events

best practices analysis, 69
 configuration, 577–578
 custom, 272
 incident investigation, 310
 monitoring, 265–268, 271, 570–576
 performance counters thresholds, 568–569
 processing based on, 290
 tracing and, 580

example tag, 602

exception handling

- adding value, 554–556
- flow, 550–554
- leading to operational procedures, 556–557
- planning for, 544–548
- pros vs. cons, 66–69
- strategy, 548–550
- system failure, 543–544

exception tag, 601

exceptions

- analysis and design, 189–192
- architecture, 184
- modeling application. See application modeling
- monitoring, 271
- system framework, 403–405

execution

- batch, 297–298
- batch job, 472–474
- BatchJobFactory class, 474–476
- exception handling, 548–550
- failures of, 25
- job flow, 476–482
- method processing, 528–529
- mobilization, 81–82
- production readiness, 26–27
- progress reporting, 482–486

exemplars

- defining test, 108–109
- development, 100–101

exit criteria, 107–108

exit permutations

- code coverage, 50
- integration testing, 53
- unit testing, 51

expected results

- comparison tools, 106
- testing, 625

eXtensible Markup Language (XML)

documentation

- commenting standards, 601–602
- project settings, 598–599

extensions

- fitting for purpose, 11–12
- in quality landscape, 39
- system framework, 403

extraction

- data, 292
- generic batch jobs, 498
- tools, 627–628

F

F5

- protecting against resubmission of data, 540–543
- supporting and controlling user behavior, 536–537

factory pattern, 628–633

failed defects, 118

failover clustering, 210

failure audits

- defined, 267
- events, 573

failure resilience. See designing for resilience

farms, server, 206

fault tolerance

- in Deployment Guide, 258–259
- high availability and, 205–211
- individual server configuration, 203–204
- shared storage, 212–213

features, tracking, 58

fiber cards, 204

fields

- adding inputs and outputs, 439–444
- defining, 181–182
- mandatory. See mandatory fields
- validation, 179

files

- data. See data
- generic batch jobs, 498
- using multiple trace, 589–590

filtering, downstream, 268

final releases, 101–102

financial analysis of quality landscape, 64–65

firewalls, 200–201

fit for purpose

- applications, 10–14
- environments, 14–18
- tools and processes, 18–20

fixed-width fields, 181

fixes. See also defect management

- costs of, 62
- implementation, testing and release, 321–322
- quality and, 54–56

fix-on-fail

- application maintenance, 319–320
- testing, 239–240

flagged accounts

- batch jobs, 461–463
- batch processes, 424
- indexing, 518–520

logical data models, 426
removing, 367–369

flexibility

conceptual design, 331
monitoring configuration, 576–578
in quality landscape, 37, 39

flow, architecture. See architecture

flow, modeling. See models

fluml, 189–190

folder structure

classifying releases, 643
configuration management, 127–131
defining, 596–598

formats

coding standards, 609
defining, 181

forms, requirements, 356–357

Forward button, 533–537

frameworks. See architecture

frequency, reporting, 282

front-end firewalls, 201

functional batch jobs, 290–294

functional gaps, 173

functional incidents, 314–315

functional requirements

batch, 367–369
data element, 352–356
defining, 350–352
email processing, 366
high-level. *See* HFRs (High-Level Functional Requirements)
online user interface and processing, 356–366
reporting, 369–370

functional testing, 230, 235–236

functionality

build and unit test process, 43
describing, 603–605
generic, 497–499
input requirements, 93
integration testing, 53
Login flow, 448

functions

application maintenance, 319–324
code coverage, 50
page specifications, 180

G

garbage collection, 525–527

generated code, 60

generic batch jobs and functionality, 497–499

generic defects, 114

generic services, 47

GET, 533–537

getter properties, 427

globalization

accessibility, 149
application maintenance, 653

governance, 97

graphics, 152

grouping

nodes, 300–301
requirements, 163

guides

cost-benefit analysis of quality landscape, 65
Developer's Guide, 324–325
development process, 97
Operations Manual. *See* Operations Manual
thinking out-of-the-box, 140–141
user training, 22–23

H

HA (high availability), 205–211

hacking, URL, 538–540

handling exceptions. See exception handling

handover documents, 22

hard parsing, 523

hardware

constraints, 87–88
development process, 96–97
high availability and fault tolerance, 205–211
individual server configuration, 203–205
network segmentation, 213–214
production environments, 201–203
scaling and sizing system, 255–257
shared storage area, 211–213
technical testing, 234

harness, test

development and review, 45
requirements, 626–627

hashing, URL, 207

headlines, 123

help pages, 180

HFRs (High-Level Functional Requirements)

batch, 367–369
data element, 352–356
defining, 350–352
email processing, 366
in logical data model, 426–427

HFRs (High-Level Functional Requirements)

(continued)

online user interface and processing, 356–366

reporting, 369–370

security answers, 443

hidden fields, 177

high availability (HA), 205–211

High classification, 119–122

High-Level Functional Requirements (HFRs).

See **HFRs (High-Level Functional Requirements)**

high-level technical requirements (HTRs), 370–372

high-level transaction flow, 408–411

history configuration, 125–132

horizontal partitioning, 520–521

hot-fixing, 55

hot-swappable components, 204

housekeeping

account cleanup jobs, 461–463

batch requirements, 367–369

design considerations, 154

in Operations Manual, 254

HTRs (high-level technical requirements), 370–372

human error

protecting against resubmission of data, 540–543

protecting against URL hacking, 538–540

supporting and controlling user behavior, 533–537

validating and encoding user input, 537–538

human resources, 86–87

I

IBatchManager

defined, 474–476

methods, 476–482

idempotent, 540–541

identification

requirement, 162

use case, 169

identifiers, 334–335

identity, batch, 298

if conditions, 609–610

impact analysis

branching and, 132

defect contents, 123

defect management workflow, 115–117

impacted defects, 117

implementation

batch jobs, 293

costs, 64–65

defect management, 117–118

design considerations, 138, 148, 150–152

fix, 321–322

incident resolution, 313

production readiness. See **production readiness**

importation, data, 498

incident investigation, 305–315

alerts, 273–274

application maintenance and, 319–320

commenting standards, 603

defined, 650

designing for. See **designing for incident investigation**

in Operations Manual, 254

overview, 305

in production environments, 239

providing audit trail, 509–515

roadmap, 306–313

role of Service Delivery team, 251

summary, 315

technical reporting, 283

typical, 313–315

indexing tables, 518–520

indicators, 178

individual server configuration, 203–205

information events

defined, 267, 572

monitoring, 271

information traces, 583

infrastructure

configuration management, 127

constraints, 87–88

role of Service Delivery team, 248

initialization

batch job execution flow, 476–482

batch job progress reporting, 484–485

performance counters, 562

Initialize method, 459

injectors, load, 234

inputs

adding to application, 439–444

architecture, 184

defining build and release, 103–104

defining test, 107

development, 93–96

message and interface mappings, 186

use case flow, 172

validating and encoding user, 537–538

installation

- developing guide, 97
- in development scope, 93
- Installation and Configuration Guide, 259–260
- in Operations Manual, 254

instrumentation

- code quality, 48
- incident investigation, 310
- pros vs. cons, 70

integration

- fit for purpose environments, 16
- improving quality with, 59
- production readiness, 6
- with third-party components, 91

integration testing

- conceptual architecture, 623–624
- cost-benefit analysis of quality landscape, 66
- costs, 65
- defining core activities, 98–99
- in development scope, 92
- vs. functional testing, 236
- message and interface mappings, 185–186
- processes, 52–53
- regression testing and, 237
- third-party, 233

integrity

- in quality landscape, 39
- requirements, 166
- running checks, 291–292

interactions, 189–192**interfaces**

- BatchJobFactory class, 474–476
- IBatchManager , 476–482
- mappings, 185–186
- recycling and pooling objects, 526

internalization, 149**intraday batch jobs, 471****invalid transaction counts/rate, 563****investigation, incident. See incident investigation****investigative tooling, 593****IP addresses, 207****isolation, transaction**

- defined, 155
- exception handling, 547–548

iterative development, 8–9**J****job schedulers. See scheduler, batch****job servers, 205****jobs, batch. See also batch**

- defined, 296–298
- processing, 290–294

joins, 519–520**L****laminates, 247****landscape, quality. See quality****languages**

- accessibility, 149
- code. See code
- commenting standards, 603
- static page elements, 615–616

late binding techniques, 475**layering**

- code quality, 47
- component, 182
- data access, 428
- data entities, 427
- logical. See logical layers

layout. See design**least number of connections, 206–207****levels of live service support, 244–246****libraries**

- separating components, 639–640
- splitting out test materials, 638–639

licensing

- custom built vs. off-the-shelf, 153
- software topology, 217
- tools and technology constraints, 86

life-critical systems, 84**lifecycle testing, 236****linked databases, 513****linking trace entries, 588–590****links, 144****lists**

- counters, 563
- partitioning, 520–521

live messaging, 88**live service system**

- application maintenance. See application maintenance
- batch jobs, 294
- defined, 239

live support, 244–246**load balancing**

- defined, 206–209
- managing configuration changes, 533
- role of Service Delivery team, 250

load injectors, 234

load tests, 232–233

local processing, 172

localization

accessibility and, 149

static page elements, 615–616

location constraints, 89–90

locking

design considerations, 156

exception handling, 547

wait-retry pattern, 551

logging

analysis and design, 187–189

analytics, 284

avoiding unnecessary method calls, 523–525

code quality, 48

custom data/time, 159

events, 570–576

improving quality with, 60

incident investigation, 310

investigative tooling, 593

multi-threaded jobs, 491

pros vs. cons, 69

role of Service Delivery team, 250–251

in system framework, 403–405

transferring files, 291

logical component models, 428–430

logical data models

data access and, 426–428

defined, 178

logical layers

architecture planning, 397–399

common application framework, 399–402

system framework, 403–405

tooling framework, 406–407

Login page

functional requirements, 361

looking for repeating patterns, 445

modeling flow, 445–449

storyboard, 344–345

use case, 379–381

Logout page

functional requirements, 361–362

modeling flow, 449–450

storyboard, 345

use case, 381–382

lookup tables, 502–505

Low classification, 119–122

Lucas, George, 143

M

Main Body, 412

maintenance, 324–325

application. See application maintenance

audit records, 514

custom built vs. off-the-shelf, 153

defining structure, 97

design considerations, 138, 152

fit for purpose environments, 18

in Operations Manual, 254

processes and tools, 20

production readiness, 7

in quality landscape, 39

role of Service Delivery team, 249

screens, 178

Service Delivery and Operations, 260

testing, 232

management

configuration. See configuration management

defect. See defect management

defining test, 109

project management triangle, 27–31

managers

batch, 472–474

batch job progress reporting, 482–486

`BatchJobFactory` class, 474–476

business, 418–420

in conceptual architecture, 398

controlling batch processes, 486–488

generic batch jobs, 497–499

`IBatchManager`, 476–482

performance counters, 566

testing, 623

mandatory fields

design considerations, 151

indicators, 145–146

validation, 420, 617

manual testing

cons, 628

defined, 626

manuals. See guides

mappings

message and interface, 185–186

page specifications, 180

use case as, 172

maps, navigation. See navigation maps

Mark One Eyeball reviews, 48–49

master data sets, 509

master-agent architecture

- analytics, 284–285
- monitoring, 264–268
- software topology, 219–220

maximum development time, 71–75**maximum field length, 181****meaningful data, 501–509****media, 140****Medium classification, 119–122****memory**

- monitoring utilization, 271
- recycling and pooling objects, 525–527

merging approach

- application maintenance, 323
- configuration management, 131–132

messages

- alerts, 273
- defined, 611
- event types, 574
- implementation considerations, 152
- interface mappings and, 185–186
- supporting configurability, 617–618
- system error, 179

methodology constraints, 83–85**methods**

- avoiding unnecessary calls, 523–525
- batch job progress reporting, 482–486
- code coverage, 50
- component responsibility modeling, 435–439
- controlling batch processes, 486–488
- factory pattern, 628–633
- GET and POST, 533–537
- IBatchManager , 476–482
- logging, 567–568
- logical component model, 430
- Login, 445–449
- Now , 159
- optimizing processing, 528–529
- recycling and pooling objects, 525–527
- solicited email batch jobs, 455–461
- system framework, 404
- tracing. See tracing

middle-tier caches, 522**minimum development time, 71–75****minimum field length, 181****mission-critical systems, 84****mobile device accessibility, 150****mobilization, 81. See also production preparation****mock objects, 51–52****mockups**

- logging and diagnostics, 187–189
- message and interface, 185–186
- page layouts and specifications, 178–180

models

- application. See application modeling
- architecture, 182–183
- defining inputs, 94
- deployment, 257
- logical component, 428–430
- logical data, 426–428
- reporting, 280
- security and communications, 257–258

Model-View-Controller (MVC), 414**modularity**

- approach to architecture, 396
- in quality landscape, 39

monitoring

- build and unit test process, 43
 - defined, 651
 - designing for. See designing for monitoring
 - fitting for purpose, 13
 - integration testing, 53
 - load balancing, 208
 - with OAT, 231
 - in Operations Manual, 254
 - by Operations team, 247–248
 - servers, 202–203
 - by Service Delivery team, 248–252
 - technical reporting, 283
- monitoring and alerts, 263–274**
- blackout windows, 269–270
 - defining alerts, 273–274
 - overview, 263–268
 - requirements and rules, 269
 - summary, 274
 - what is monitored?, 270–272

multi-channel, 150**multiple jobs, 492–495****multiple releases, 323–324****multiple requests, 540–543****multiple synchronized processes, 557****multiple threaded processing**

- application-specific settings, 612–613
- batch job parameters, 495–497
- batch jobs, 489–492
- multiple jobs, 492–495
- overview, 488–489
- tracing, 589

multiple trace files, 590–591
multi-purpose forms, 180
multi-site environments, 211
multi-threaded jobs, 489–492
multi-tier conceptual architecture model, 396
must requirements, 161
MVC (Model-View-Controller), 414
My Account page, 336–337

N

names

- batch jobs, 296
- coding standards, 609
- entity, 181
- event log entries, 576
- tracing, 400
- use case, 169

namespaces, 403

navigation, 533–537

navigation maps

- analysis and design, 173–174
- defining inputs, 94
- online application walkthrough, 334

network interface cards (NICs), 204

Network Load Balancing (NLB), 209

networks

- monitoring utilization, 271
- role of Service Delivery team, 248
- SANs. See SANs (storage area networks)
- segmentation, 213–214
- shared storage area, 211–213

new accounts, creating. See **Create New Account page**

new defects, 113–115

NICs (network interface cards), 204

NLB (Network Load Balancing), 209

nodes

- batch jobs, 297
- batch scheduling, 300–301
- defined, 209

noise, 251

non-conformance, 607

non-critical systems, 84

non-functional requirements, 93

non-stop processing, 557

notes, release

- in development scope, 93
- user training, 22–23

Now **method, 159**

null values, 182

NumberOfItems, 560

O

OAT (operational acceptance testing), 230–232 **objects**

- creation/destruction, 480
- factory pattern, 628–633
- mock, 51
- recycling and pooling, 525–527
- testing requirements, 624–625

off-the-shelf

- vs. custom built, 152–153
- design considerations, 138, 148
- reporting components, 428

online application walkthrough, 331–335

online help, 180

online user interfaces, 356–366

operability

- build and unit test process, 43
- integration testing, 53
- in quality landscape, 39
- requirements, 166

operating systems

- fitting for purpose, 11
- in production environment, 200

operational acceptance testing (OAT), 230–232

operational procedures

- in development scope, 93
- exceptions lead to, 556–557

operations

- architecture, 202–203
- fit for purpose environments, 17
- role of, 247–248
- Service Delivery and Operations. See Service Delivery and Operations

Operations Manual, 141

- architecture planning, 254–259
- defined, 244
- incident investigation, 315
- Installation and Configuration Guide, 259–260
- maintenance and administration, 260
- overview, 252–254

optimizing method processing, 528–529

optimizing queries, 522

optional fields, 151

out-of-scope features, 161

outputs

- activity-based tracing, 588
- adding to application, 439–444
- architecture, 184
- batch, 297–298
- defining templates and exemplars, 100–101
- message and interface mappings, 186
- reports, 279–281
- tracing, 400–402

outputs, analysis and design

- architecture and application component specifications, 182–184
- component lists and specifications, 184–185
- data dictionary, 180–182
- defined, 138
- interactions and exceptions, 189–192
- logging and diagnostics, 187–189
- message and interface mappings, 185–186
- navigation maps, 173–174
- page layouts and specifications, 178–180
- requirements, 161–166
- scope, 160–161
- storyboards, 174–177
- use case diagrams and use cases, 166–173

Overall Total performance counter, 564

ownership, 162

P

packaging

- classifying releases, 641–644
- in development scope, 92
- thinking out-of-the-box, 139–141

Page Header, 412

page layouts

- analysis and design, 178–180
- defining inputs, 94
- implementation considerations, 152
- presentation tier, 411–414

pages

- accounts. See accounts
- My Account, 336–337
- testing requirements, 627
- Welcome, 335–336

para **tag, 602**

param **tag, 601**

parameters

- batch job, 495–497
- BatchJobFactory class, 474–475
- modeling inputs and outputs, 442
- protecting against URL hacking, 538–540

partitioning, 520–521

passive nodes

- defined, 209–210
- primary and secondary data centers, 223–226
- shared storage and, 211–212

passwords

- Activate Account page, 360
- changing. See Change Password page
- design considerations, 146
- functional requirements, 362–363
- in logging, 188
- Login page, 446–449
- modeling inputs and outputs, 439–444
- My Account page, 337
- online application walkthrough, 332
- requirements, 353
- Re-Request Activation Key flow, 452–453
- resetting. See Reset Password page
- use case diagram, 167

paths

- code coverage, 50
- unit testing, 51

patterns

- factory, 628–633
- finding repeating, 445
- matching, 495
- wait-retry. See wait-retry pattern

pausing batch jobs, 488

PDAs (personal digital assistants), 150

PDCs (primary data centers), 220–227

peer reviews, 48–49

performance

- conceptual design, 331
- counters. See counters, performance
- defined, 651
- designing for. See designing for performance
- factory pattern, 632
- improving with network segmentation, 213–214
- incident investigation, 310
- maintenance and administration, 260
- monitoring, 265–268, 270–272
- multiple batch jobs, 493
- in quality landscape, 37–38
- requirements, 164–165
- tracing configuration, 591–592
- use case flow, 172

peripheral monitoring, 251

persistence, 207

personal digital assistants (PDAs), 150

PG-004 Welcome Page, 335–336

- phase containment, 8**
- physical tiers, 407–408**
- picture-verification schemes, 147**
- planning architecture. See architecture planning**
- planning failures, 25**
- policy compliance, 85**
- pooling**
 - counters, 563
 - objects, 525–527
- POST, 533–537**
- postbacks, 534**
- power supplies, 204, 220–221**
- practices, best. See best practices**
- pre-conditions, 169**
- prefixes, tracing, 589**
- preparing for production. See production preparation**
- pre-production environments, 238–241**
- presentation tier, 397–398, 411–414**
- pricing. See costs**
- primary actors, 169**
- primary data centers (PDCs), 220–227**
- primary site environments, 238–239**
- priority, defect, 119–123**
- privacy statements, 144**
- procedures**
 - exceptions leading to, 556–557
 - role of Operations team, 247–248
 - testing, 231–232
- proceeding at risk, 86**
- processes**
 - batch, 424–425
 - batch control, 486–488
 - change control mobilization, 133
 - configuration constraints, 125–132
 - constraints, 83–85
 - custom, 272
 - defect management workflow, 111–119
 - defining, 104–105
 - defining defect contents, 119–122
 - defining defects priority/severity, 119–122
 - development, 96–100
 - fit for purpose, 18–20
 - monitoring, 265–268
 - of production readiness, 26–31
 - quality. See quality
 - updates, 321
 - user training, 22
- processing**
 - application modeling. See application modeling
 - batch, 289–295
 - business components, 416–422
 - designing for batch. See designing for batch
 - designing for performance. See designing for performance
 - email requirements, 366
 - functional requirements, 356–366
 - multiple threaded. See multiple threaded
 - processing
 - non-stop, 557
 - page layout, 179
 - tracing, 400–402
 - use case flow, 172
- processors**
 - business, 398, 417
 - monitoring performance, 270
 - performance counters, 566–567
 - testing, 623
- procurement constraints, 85**
- production**
 - deployment concept, 407–408
 - preparing solution, 597–598
 - separating test matter from, 59
 - vs. testing, 652
- production environments, 197–228**
 - big picture, 198–203
 - fitting for purpose, 17
 - high availability and fault tolerance, 205–211
 - individual server configuration, 203–205
 - network segmentation, 213–214
 - overview, 197
 - primary and secondary data centers, 220–227
 - shared storage area, 211–213
 - software topology, 214–220
 - summary, 227–228
- production preparation**
 - build, release and regular integration
 - mobilization, 102–105
 - change control mobilization, 132–133
 - configuration management mobilization, 125–132
 - constraints, 83–90
 - defect contents, 123–124
 - defect management constraints, 110–111
 - defect management reports, 124
 - defect management workflow, 111–119
 - defects severity and priority, 119–122
 - development inputs, 93–96
 - development process, 96–100
 - development scope, 90–93

- development templates and exemplars, 100–101
- importance of, 80–82
- key activities, 82–83
- overview, 79–80
- summary, 133–135
- test mobilization, 105–109
- wrapping up development, 101–102
- production readiness**
 - defined, 4–10
 - fit for purpose applications, 10–14
 - fit for purpose environments, 14–18
 - fit for purpose tools and processes, 18–20
 - importance of, 23–26
 - overview, 3–4
 - process of, 26–27
 - project management triangle, 27–31
 - summary, 31–34
 - training users, 20–23
- productivity**
 - guides, 22
 - quality and, 60
 - scripts, 127–131
- profiling**
 - application maintenance, 652–654
 - code, 48–49
 - code metrics profilers, 654–655
 - cost-benefit analysis of quality landscape, 66
 - pros vs. cons, 70
- progress reporting, 482–486**
- projects**
 - defining, 596–598
 - documentation. *See* documentation
 - management triangle, 27–31
 - portal, 23
 - settings, 598–600
- properties**
 - analytics, 284
 - architecture, 185
 - batch jobs, 296–297
 - data entity, 427
 - event, 267
 - factory pattern, 628–633
- prototyping**
 - design considerations, 147–148
 - stubs and simulators, 635
- proxy servers, 201**
- public interfaces, 52–53**
- purging**
 - audit records, 515

- batch jobs, 291
- generic batch jobs, 498
- purpose, fit for. *See* fit for purpose**

Q

- quality, 35–78**
 - build and unit test process, 43–46
 - characteristics, 36–40
 - code, 46–48
 - code coverage, 50
 - code profiling and peer reviews, 48–49
 - construction, 41–43
 - cost of defects, 61–64
 - cost-benefit analysis, 65–71
 - defect fixing, 54–56
 - defects and testing, 53–54
 - documentation generation, 51–52
 - estimates, 71–75
 - failures of, 25–26
 - financial analysis, 64–65
 - importance of, 40–41
 - integration test process, 52–53
 - overview, 35–36
 - price of, 61
 - project management triangle, 30–31
 - summary, 76–78
 - sweeping and, 56–57
 - tips for improving, 57–60
 - unit testing, 50–51
- quarantining data, 557**
- query string parameters, 177**
- query tuning, 518–523**
- questions, security, 355–356**
- queuing**
 - counters, 563
 - event types, 575
 - solicited emails, 455–461
- quickest response time, 207**

R

- rapid turnaround approach, 109**
- RateOfCountsPerSecond, 560**
- readability, 39**
- readiness, production. *See* production readiness**
- reading data, 156**
- ready-to-build processes, 104–105**
- ready-to-release processes, 104–105**

recording

- date/time, 505–506
- providing audit trail, 509–515

recovery strategies

- database backup, 292–293
- in Deployment Guide, 258–259
- fitting for purpose, 13–14
- maintenance and administration, 260
- role of Service Delivery team, 249
- shared storage and, 212
- testing, 232

recreating an incident

- defined, 306–308
- investigation, 312

recruitment constraints, 85

recycling

- batch jobs, 293
- objects, 525–527

re-factoring, 56

reference data

- defining inputs, 95
- page layout, 178
- validity, 155

refreshing

- resubmission of data, 540–543
- user behavior, 536–537

region accessibility, 149

registration

- component responsibility modeling, 435–439
- services, 415

registries, 613–614

regression testing

- cost-benefit analysis of quality landscape, 65
- defined, 6, 230, 236–238
- fit for purpose environments, 16–17
- improving quality with, 59
- test mobilization, 106

regular integration

- defined, 6
- fit for purpose environments, 16
- mobilization, 80, 102–105
- processes and tools, 19

rejected defects, 117

release notes

- in development scope, 93
- user training, 22–23

releases

- application maintenance, 323–324
- application maintenance functions, 319
- branching and, 131–132

- build and unit test process, 46
- classifying, 641–644
- configuration management, 127
- defect contents, 123
- defect management workflow, 118
- defining core activities, 99
- in Developer's Guide, 325
- in development scope, 92
- fix, 321–322
- improving quality, 58–60
- mobilization, 102–105
- separating components, 639–640
- strategy, 8–9
- wrapping up development, 101–102

releasing objects, 526–527

reliability

- in quality landscape, 38
- requirements, 163–164

remarks tag, 601, 605–607

remote processing, 172

remoting, 414

repeating patterns, 445

reporting, 275–287

- analytics, 284–285
- architecture, 281–282
- batch job progress, 482–486
- components, 463–465
- designing for. *See* designing for reporting
- fitting for purpose, 13
- functional requirements, 369–370
- multi-threaded jobs, 490
- in Operations Manual, 254
- overview, 275–281
- role of Service Delivery team, 249–252
- servers, 202
- summary, 285–287
- technical, 283
- testing, 625

reports

- completion. *See* completion reports
- defect management, 124
- in development scope, 92
- incident investigation, 310
- inputs, 95

repositories

- configuration management, 126–127
- email address, 354

requests

- modeling inputs and outputs, 439–444
- resubmission of data, 540–543

solicited emails, 456–457
 transaction flow, 408–411
 user behavior, 533–537

requirements

analysis and design, 161–166
 batch, 367–369
 batch processes, 424
 commenting standards, 605–606
 data element, 352–356
 defining, 5–6
 defining functional, 350–352
 defining inputs, 93
 defining process, 96–97
 vs. design, 147
 email processing, 366
 fit for purpose applications, 10–14
 fit for purpose environments, 14–18
 fit for purpose processes and tools, 18–20
 functional testing, 235–236
 high-level functional. *See* HFRs (High-Level Functional Requirements)
 monitoring, 269
 online user interface, 356–366
 page specifications, 178–180
 production environment, 198
 production line activities, 82–83
 project management triangle, 30–31
 reporting, 369–370, 428
 reporting components, 463–465
 technical, 370–372
 testing, 624–628
 tracking, 58
 use case, 171
 validation, 43

Re-Request Activation Key page

defined, 332
 email batch processing, 460
 functional requirements, 359–361
 modeling flow, 452–453
 storyboard, 342–344
 use case, 377–379

research, 147–148**Reset Password page**

functional requirements, 362
 modeling flow, 453
 online application walkthrough, 332
 storyboard, 345–346
 use case, 382–383

resilience, 651. *See also* designing for resilience**resolution**

implementation, 312–313
 incident investigation, 306–309

resolved defects, 123–124**resources**

constraints, 86–87
 multi-threaded jobs, 489–491
 static page elements, 615–616

responses

modeling, 439–444
 page specifications, 180
 quickest time, 207
 transaction flow, 408–411
 user behavior, 533–537

restarting, automatic, 557**resubmission of data**

protecting against, 540–543
 user behavior, 537

results, 430**retrying. *See* wait-retry pattern**

returns **tag, 601**

reusability

code quality, 47
 generic batch jobs and functionality, 497–499
 modeling application. *See* application modeling in quality landscape, 39

reviews

build and unit test process, 44–46
 defining core activities, 99
 impact analysis, 115–117
 peer, 48–49
 Service Delivery team, 250

risks, 86–87**roadmap, incident investigation, 306–313****roles in use cases, 166–173****root cause analysis, 123****round robin, 206****rules**

enabling, 599–600
 monitoring, 269

run dates. *See* dates**runtime. *See* time****S****sampling periods, 264****sandboxes, 232****SANs (storage area networks)**

defined, 203
 primary and secondary data centers, 223–225

SANs (storage area networks) (continued)

- reporting architecture, 281–282
- server configuration, 205
- shared storage area, 211–213
- SB-001 Create New Account, 337–339**
- SB-002 Activate Account, 339–342**
- SB-003 Re-Request Activation Key, 342–344**
- SB-004 Login, 344–345**
- SB-005 Logout, 345**
- SB-006 Request Password, 345–346**
- SB-007 Change Password, 346–348**
- SB-008 Update Account, 348–349**
- SB-009 Close Account, 349–350**

scalability

- in quality landscape, 39
- requirements, 165–166
- systems, 255–257

scavenging, 522

scenarios

- defining test, 107
- use case flow, 173

scheduled processing, 289

scheduled tasks, 290

scheduler, batch

- batch architecture, 470–472
- defined, 295–301
- multiple jobs, 492–495

scope

- analysis and design, 160–161
- defining build and release, 103–104
- defining test, 106–107
- development, 90–93
- failures of, 24–25
- importance of, 40–41
- project management triangle, 27–31
- of support, 246

scripts

- coding standards, 609–610
- configuration management, 127–131
- defects and testing, 54
- in development scope, 92
- improving quality, 58, 60
- test mobilization, 106

SDCs (secondary data centers)

- defined, 211
- production environments, 220–227

secondary site environments, 238–240

Secure Sockets Layer (SSL), 150

security

- in architecture, 184

- batch, 298
- conceptual design, 331
- constraints, 89–90
- data centers, 221
- in Deployment Guide, 257–258
- design considerations, 154
- with DMZ, 200–201
- fitting for purpose, 12
- page specifications, 180
- primary and secondary data centers, 222–223
- in quality landscape, 37, 39
- question inputs, 439–444
- question requirements, 355–356
- requirements, 166
- resubmission of data, 540–543
- role of Service Delivery team, 249
- software topology, 216, 219
- URL hacking, 538–540
- use case diagram, 168
- use case flow, 172

see **tag, 602**

see also **tag, 602**

segmentation, 213–214

selection criteria, 279–281

servers

- high availability and fault tolerance, 206–210
- individual configuration, 203–205
- Installation and Configuration Guide, 259–260
- monitoring requirements, 269
- multiple batch jobs, 492
- multi-threaded jobs, 489–491
- performance counters, 564–565
- primary and secondary data centers, 220–227
- in production environment, 200–203
- regression testing, 236–237
- scaling and sizing system, 255–257
- software topology, 213–220
- state management, 176–177
- supporting and controlling user behavior, 533–537
- in web services facade, 414–416

Service Delivery and Operations, 243–261

- architecture planning, 254–259
- incident investigation, 309
- Installation and Configuration Guide, 259–260
- maintenance and administration, 260
- Operations Manual, 252–254
- overview, 243–244
- providing audit trail, 509–515
- role of Operations team, 247–248

- role of Service Delivery team, 248–252
- scope of support, 246
- summary, 261
- three levels of live support, 244–246
- Service Level Agreements (SLAs), 319**
- Service Orientated Architecture (SOA), 414–416**
- service shell, 474**
- session-based logging, 189**
- sessions, sticky, 207**
- Set Continue URL, 414**
- sets, master data, 509**
- setter properties, 427**
- settings**
 - code quality, 48
 - coding standards, 610–615
 - project, 598–600
- severity, defect, 119–122, 123**
- shall requirements, 161**
- shared information servers, 205**
- shared storage area, 211–213**
- sharing constraints, 88–89**
- shells**
 - batch job execution flow, 476–482
 - batch job progress reporting, 482–486
 - execution and control framework, 472–474
- should requirements, 161**
- simulators**
 - designing for testing, 633–635
 - development considerations, 151
 - in development scope, 91
 - improving quality, 58
 - integration testing, 52
 - test mobilization, 106
 - unit testing, 51
- single points of failure (SPOF), 207–208**
- site maps, 94**
- sizing systems, 255–257**
- SLAs (Service Level Agreements), 319**
- smoke testing**
 - conceptual architecture, 623–624
 - defined, 6
 - mobilization, 108
 - regression testing and, 237
- SOA (Service Orientated Architecture), 414–416**
- soak tests, 233**
- software**
 - builds. See builds
 - monitoring. See monitoring and alerts
 - production line. See production preparation
 - production readiness. See production readiness topology, 214–220
- solicited emails**
 - batch jobs, 455–461
 - testing requirements, 626
- solution**
 - architecture, 201–203
 - overview in Developer's Guide, 324
 - overview in Operations Manual, 254
 - preparing, 596–600
 - Solution Architecture Design, 253–254
- source code files**
 - audit records, 514
 - configuration management, 127
- source IP addresses, 207**
- source property, 267**
- space**
 - constraints, 87–88
 - monitoring disk, 271
- special characters, 537–538**
- special considerations, 182**
- specifications**
 - component lists and, 184–185
 - requirement, 162
- SPOF (single points of failure), 207–208**
- SQL (Structured Query Language) execution, 498**
- SSL (Secure Sockets Layer), 150**
- stability**
 - in quality landscape, 38
 - requirements, 163–164
- stacking transactions, 533**
- standard usage indicators, 178**
- standards**
 - best practices analysis, 69–71
 - code quality, 47
 - coding, 609–610
 - commenting, 603–608
 - defining, 600–601
 - in Developer's Guide, 324
 - XML documentation comments, 601–602
- start time, 297**
- state**
 - accounts, 352
 - application caching, 527
 - Change Password flow, 453–454
 - change password functionality, 385–386
 - component responsibility modeling, 437–439
 - defect contents, 123
 - functional testing and, 236
 - lookup tables, 502–505

state (continued)

- management, 176–177
- page specifications, 180
- partitioning based on, 520–521
- providing audit trail, 510–511
- user, 351

statements

- code coverage, 50
- JOIN, 519–520
- TODO. See TODO statements
- unit testing, 51
- use case, 422–423

static code analysis

- defined, 49
- pros vs. cons, 70
- regression testing and, 238

static page elements

- configurability, 615–616
- defined, 611

status reporting

- batch job, 482–486
- multi-threaded jobs, 490

sticky sessions, 207

Stop method, 486–488

storage

- application-specific configuration settings, 613–615
- audit records, 514–515
- batch jobs, 292
- configuration management, 129
- primary and secondary data centers, 220–227
- shared areas, 211–213
- state in database, 177

storage area networks (SANs). See SANs (storage area networks)

storyboards

- account pages. See accounts, storyboard analysis and design, 174–177
- defining inputs, 94
- use case, 169

stress tests, 233

Structured Query Language (SQL) execution, 498

structures, 127–131

stubs

- designing for testing, 633–635
- development considerations, 151
- functional testing and, 236
- improving quality, 58
- integration testing, 52
- unit testing, 51

style

- coding standards, 609–610
- defining inputs, 94
- page layout, 178

success audits

- defined, 267
- Success Audit events, 573

summary tag, 601

support

- application maintenance. See application maintenance
- browser, 157
- configurability, 610–611
- custom built vs. off-the-shelf, 153
- design considerations, 138, 148, 152
- fit for purpose environments, 17
- processes and tools, 20
- production readiness, 7
- in quality landscape, 39
- requirements, 166
- scope of, 246
- three levels of live, 244–246
- tools and technology constraints, 86
- user behavior, 533–537

sweeping, 56–57

synchronizing processes, 557

sysadmins, 248

system alerts, 247. See also alerts

system exceptions, 548–550

system failure

- exception handling, adding value, 554–556
- exception handling flow, 550–554
- exception handling planning, 544–548
- exception handling strategy, 548–550
- exceptions lead to operational procedures, 556–557
- handling, 543–544

system registries, 613–614

systems

- critical categories, 84
- defect classifications, 120–121
- error messages, 179
- framework, 403–405
- scaling and sizing, 255–257
- testing, 6

T

tab order, 179

tables

- indexing, 518–520
- partitioning, 520–521
- tags, 601–602**
- tape silos, 202**
- TCO (total cost of ownership), 64–66**
- teaming, 204**
- teardown, 555–556**
- technical batch jobs, 290–294**
- technical designs, 91**
- technical incidents, 313–314**
- technical reporting**
 - defined, 276
 - overview, 283
- technical requirements, 370–372**
 - build and unit test process, 43
 - design considerations, 156–157
 - monitoring, 269
 - in Operations Manual, 253
 - scope, 30–31
- technical testing**
 - defined, 230, 232–235
 - integration, 53
 - in production environments, 239
- technology**
 - conceptual design, 331
 - constraints, 85–86
 - guides, 22
 - in quality landscape, 37
- templates**
 - defining test, 108–109
 - development, 100–101
 - improving quality, 60
- Terminate method, 459**
- termination**
 - batch job execution flow, 476–482
 - batch job progress reporting, 484–485
 - controlling batch processes, 486–488
- Terms of Use (TOU) pages, 144**
- test stubs**
 - architecture, 106
 - in development scope, 91
- testing**
 - acceptance, 231–232
 - batch jobs, 293–294
 - build. See build testing
 - configuration management, 127–131
 - custom built vs. off-the-shelf, 153
 - defect classifications, 120–121
 - defined, 650
 - defining core activities, 98–99
 - defining inputs, 95
 - design considerations, 151
 - designing for. See designing for testing in Developer's Guide, 324
 - in development scope, 92
 - fit for purpose environments, 16–17
 - fixes, 321–322
 - functional, 235–236
 - incident investigation, 311–312
 - investigative tooling, 593
 - load balancing, 208
 - mobilization, 80, 105–109
 - monitoring and, 270
 - page layout, 178
 - preparing solution, 597–598
 - primary and secondary data centers, 226
 - processes and tools, 19
 - vs. production, 652
 - in production environment, 238–241
 - production readiness, 6, 8
 - quality. See quality
 - regression, 236–238
 - requirements and design, 163
 - review, 655–658
 - splitting out, 638–639
 - technical, 232–235
 - types of, 230
 - unit. See unit testing
 - use case flow, 173
- The Hitchhiker 's Guide to the Galaxy (Adams), 142***
- third-party components**
 - clustering support, 210
 - vs. custom built, 152–153
 - custom calendars and, 159
 - defining inputs, 96
 - in development scope, 91
 - master-agent architecture, 219–220
 - software topology, 213–215
 - testing, 233
- threaded processing, multiple. See multiple threaded processing**
- three levels of live service support**
 - overview, 244–246
 - role of Operations team, 247–248
 - role of Service Delivery team, 248–252
 - scope of, 246
- three-tier architecture, 183, 235**
- thresholds, 569–570**
- throw-away code, 47**

ticketing

- scope of support, 246
- three levels of live service support, 244–245

tiers

- presentation, 411–414
- three-tier architecture, 183, 235

time

- batch jobs, 297
- batch run date, 302
- component responsibility modeling, 437–439
- entries and validation, 157–159
- indexing tables, 519
- monitoring events, 267
- multiple batch jobs, 493–494
- in production environment, 200
- providing audit trail, 510
- recording, 505–506
- reporting architecture, 282
- runtime, 11–12

timeouts

- exception handling, 547–548
- wait-retry pattern, 551

timescales

- estimating costs and, 71–75
- project management triangle, 27–31

timestamps, 400–401

TODO statements

- commenting standards, 605, 608
- defining templates and exemplars, 100
- improving quality with, 60

token synchronization, 540

tolerance, fault. See fault tolerance

tolerances, 569–570

tooling

- framework, 406–407
- investigative, 593
- role of Service Delivery team, 252

tools

- best practices analysis, 66–71
- constraints, 85–86
- defects and testing, 54
- fit for purpose, 18–20
- improving quality, 58, 60
- incident investigation, 311
- technical testing, 234
- test mobilization, 106
- testing, 624–628
- user training, 22

total cost of ownership (TCO), 64–66

TOU (Terms of Use) pages, 144

tracing

- activity-based, 588–590
- avoiding unnecessary method calls, 523–525
- in common application framework, 183, 400–402
- configuration, 591–592
- events, 570–572
- method entry/exit, 567–568
- multiple trace files, 590–591
- overview, 580–588
- requirements, 161–162
- in system framework, 403–405

tracking

- defect management, 110–111
- releases and features, 58

training

- cost-benefit analysis of quality landscape, 65
- fit for purpose environments, 17
- testing, 232
- User Guides, 140–141
- users, 20–23

Transaction Retries reports, 277

Transaction Summary, 412

transactions

- based logging, 189
- counts and throughput, 563
- custom performance counters, 272
- design considerations, 154–155
- end-to-end performance reports, 277–278
- high-level flow, 408–411
- intermittent counter updates, 568–569
- maintaining counts, 507–508
- performance counters, 564–568
- in production environment, 200
- providing audit trail, 509–515
- specifying isolation, 547–548
- stacking, 533
- test mobilization, 108
- tracing. See tracing

Transactions per Second performance counter, 564

transferring files, 498–499

transformation, data access, 428

transportation, data access, 428

travel constraints, 90

trend analysis

- defined, 283
- reporting components, 463

triggers

- database, 515

optimizing queries, 522
use case, 169

troubleshooting, 254

trusted components, 417

tuning

application layer, 523–529
database layer, 518–523
databases, 625
fixes and quality, 56
role of Service Delivery team, 252

U

UAT (user acceptance testing), 230–231

UC-001 Create New Account, 372–374

UC-002 Activate Account, 375–377

UC-002 Re-Request Activation Key, 377–379

UC-004 Login, 379–381

UC-005 Logout, 381–382

UC-006 Reset Password, 382–383

UC-007 Change Password, 384–387

UC-008 Update Account, 388–390

UC-009 Close Account, 390–392

uncommitted data, 156

undocumented features, 14

Uniform Resource Locator (URL) hacking, 538–540

Uniform Resource Locator (URL) hashing, 207

unit testing

code coverage, 50
code profiling and peer reviews, 48–49
code quality, 46–48
conceptual architecture, 623–624
defined, 6, 50–51
defining core activities, 98–99
in development scope, 92
documentation generation, 51–52
regression testing and, 237
test processes, 43–46

Update Account page

functional requirements, 364
modeling flow, 454–455
storyboard, 348–349
use case, 388–390

updating

active-active configurations, 226–227
application maintenance functions, 319–321
automatic, 216–217
batch jobs, 292

configuration management, 129
custom performance counters, 272
defining core activities, 99
design considerations, 155
intermittent counter updates, 568–569
Login page, 447–448
online application walkthrough, 332–333
performance counters, 562
providing audit trail, 510

URL (Uniform Resource Locator) hacking, 538–540

URL (Uniform Resource Locator) hashing, 207

USAA (User State Allowable Action), 351

usability

defining inputs, 94
in quality landscape, 38
tools and technology constraints, 85–86

usage

counters, 563
guides, 97
patterns, 256–257

use case diagrams

analysis and design, 166–173
defined, 93

use cases

analysis and design, 166–173
business framework components, 422–423
conceptual design. See conceptual design
defined, 93
looking for repeating patterns, 445

user acceptance testing (UAT), 230–231

user behavior

managing configuration changes, 532–533
protecting against resubmission of data, 540–543
protecting against URL hacking, 538–540
supporting and controlling, 533–537
validating and encoding user input, 537–538

User Guides, 140–141

user interfaces

functional requirements, 356–366
presentation tier, 411–414

User State Allowable Action (USAA), 351

usernames

modeling inputs and outputs, 439–444
partitioning based on, 521
requirements, 353
Re-Request Activation Key flow, 452–453

users

creating positive experience, 37, 331
design for accessibility, 149–150

users (continued)

- Service Delivery and Operations. See Service Delivery and Operations
- testing and training, 232
- tools and technology constraints, 85–86
- training, 20–23
- Welcome page, 335–336

V

validation

- batch jobs, 292
- with business manager, 418–420
- date/time, 157–158
- event types, 575
- field-level, 179
- logging and diagnostics, 188
- passwords, 146
- requirements, 353–354
- requirements and design, 43
- rules, 611, 616–617
- use case flow, 172
- user input, 537–538

value list partitioning, 520–521

value range partitioning, 520–521

values

- application-specific configuration, 610–615
- code quality, 48
- defining, 182
- message and interface mappings, 186
- performance counters, 560–562

vendor documentation, 293

vendor selection, 152–153

verification

- defect management, 117–118
- incident, 312–313
- incident investigation, 306

vertical partitioning, 520

W

wait-retry pattern

- defined, 156
- exception handling, 550–554
- network segmentation, 213
- performance counters, 563

warning events

- defined, 267, 572
- monitoring, 271

warnings

- defining alerts, 273
- performance counters thresholds, 568–569
- project settings, 598–599
- traces, 583

waterfall approach to production, 8–9

web page accounts. See accounts

web servers

- in batch architecture, 470–472
- configuration, 204
- Installation and Configuration Guide, 259
- performance counters, 564–565
- in production environment, 201
- separating components, 640

web services

- architecture planning, 414–416
- in conceptual architecture, 398
- incorporating, 396
- performance counters, 565–566
- testing, 623
- testing requirements, 627

web.config files

- application-specific settings, 611–612
- configuration management, 128–129

weekly batch, 292–293

weighting, 207

Welcome page, 335–336

wheelchair icons, 145–147

white-box testing, 50

wikis, 23

window, batch, 294–295

Window Management Instrumentation (WMI), 220

Windows

- generic batch jobs, 498
- monitoring in, 265–268
- services in batch architecture, 470–472

WMI (Window Management Instrumentation), 220

wrappers

- in architecture, 183
- business components as, 421

X

XML (eXtensible Markup Language) documentation

- commenting standards, 601–602
- project settings, 598–599