

Index

4 + 1 view architecture model, 87–89

A

Acceptance into testing, 210

Acceptance test, 202–203, 244

Actions, 98–99, 101

Activities (in the RUP). *See* disciplines.

Activities (of roles)

definition, 38–39

examples of, 39

identifying in RUP, 39

input/output. *See* artifacts.

organizing. *See* disciplines.

performing steps, 39–40

project management iteration plan, 138–139

reviewing steps, 39–40

steps, 39–40

thinking steps, 39–40

Activity: prefix, 39

Activity diagrams. *See* workflow.

Actors

business modeling discipline, 150–151

definition, 98

examples, 99–100

particular, 99

use-case-driven process, 98

Analysis and design discipline

analysis model, 177–178

analysis vs. design, 173–174

Architect Reviewer role, 176

architectural definition, 180

architectural refinement, 180–182

artifacts, 175–176, 178–179, 278

behavior analysis, 182

Capsule Designer role, 176

capsules, 178

classes, 177

component-based design, 179

component design, 182–183

data models, 177

database design, 183–184

Database Designer role, 176

design limits, 174

Design Model, 176–177

Design Reviewer role, 176

Designer role, 175–176

events, 179

interfaces, 178

introduction, 44–45

packages, 177

protocols, 179

purpose, 173

real-time systems, 178–179

roles, 175–176

round-trip engineering, 174

signals, 179

Software Architect role, 175

Software Architecture Documents, 176

subsystems, 177

tool support, 184

use-case-driven process, 107

user-interface design, 176–177

user-interface prototyping, 177

workflow, 180–184

Analysis model, 153, 177–178

Analyst roles, 273–274

Any role, 276

Architect Reviewer role, 176

Architectural mechanism, 95

Architecture. *See also* modeling software.

in analysis and design, 180–182

definition, 84–85

vs. design, 89–90

importance of, 82–84

intellectual control, 92

interested parties, 85–86

patterns, 95–96

project management, 93

Architecture (*continued*)

- prototype, iteration plan example, 250–254
- purpose of, 92–93
- requirements, 83–84
- reusability, 92–93
- significance, 89–90
- style, 94–95

Architecture, perspectives

- 4 + 1 view model, 87–89
- deployment, 88–89
- implementation view, 87–88
- interested parties, 85–86
- logical view, 87
- models, 89–90
- process view, 88
- use-case view, 89
- views, 86–87, 89–90

Architecture-centric process, 91

Artifact: prefix, 42

Artifacts. *See also entries for specific artifacts.*

- analysis and design discipline, 175–176, 178–179, 278
- within artifacts, 41
- business modeling discipline, 146–148, 277
- configuration and change management discipline, 279
- definition, 40
- delivering, 42
- deployment discipline, 43, 239–241, 279
- design discipline, 43
- documents as, 41
- environment discipline, 232–233, 280
- forms of, 40
- growing information sets, 43–44
- guidelines, 48
- implementation discipline, 43, 278–279
- information disciplines, 42–44
- management discipline, 43
- models of. *See templates.*
- project management discipline, 122–123, 279–280
- project management iteration plan, 138
- prototypes of. *See templates.*
- reports, 42
- requirements discipline, 43, 168–170, 277–278
- responsibility for, 42
- test discipline, 204–207, 279
- version control, 41

Automating business processes, 152

B

- Baseline, 70, 225–226
- Behavior analysis, 182
- Behavioral prototypes, 190, 191
- Benchmark tests, 203
- Bergstrom, Stefan, 260
- Best practices
 - continuously verify software quality, 13–14
 - control changes to software, 14–15, 29
 - develop software iteratively, 6–8, 23–25
 - manage requirements, 8–9, 25–26
 - use component-based architecture, 9–11, 26–28
 - visually model software, 11–13, 28
- Beta testing, 242, 244
- Brooks, Fred, 128
- Builds, 188, 210
- Business Analysis Models, 147
- Business Architecture Document, 148
- Business case definition, example, 248–250
- Business components, 94
- Business Designer role, 147, 273
- Business Glossary, 148
- Business modeling discipline
 - actors, 150–151
 - analysis model, 153
 - artifacts, 146–148, 277
 - automating business processes, 152
 - Business Analysis Models, 147
 - Business Architecture Document, 148
 - Business Designer role, 147
 - Business Glossary, 148
 - Business-Process Analyst role, 147
 - Business Reviewer role, 147
 - Business Rules, 147
 - Business Use-Case Models, 147
 - Business Vision Documents, 147
 - domain modeling, 145
 - e-business, 142–143
 - entity classes, 153
 - introduction, 44–45
 - organization charts, 145
 - requirements sources, 153
 - resource planning, 153
 - roles, 146–148
 - scenarios, 145–146
 - software development business, 155
 - software engineering techniques, 143–145
 - Stakeholders, 147
 - Supplementary Business Specifications, 148

- Target-Organization Assessment, 147
 - tool support, 156
 - use-case-driven process, 109
 - workflow, 148–150
 - Business Process Analyst role, 233
 - Business-Process Analyst role, 147, 273
 - Business Reviewer role, 147
 - Business Rules, 147
 - Business Use-Case Models, 147
 - Business Vision Documents, 147
- C**
- Capsule Designer role, 176, 274
 - Capsules, 178
 - Causes of software development problems, 4–5
 - CBD (component-based development), 10, 93–94
 - CCM cube, 216–221
 - Change Control Board, 223
 - Change Control Manager role, 221, 275
 - Change-control plan, 224–225
 - Change delivery, 225
 - Change management. *See also* configuration and change management discipline.
 - benefits of the iterative process, 77
 - best practices, 14–15, 29
 - environment, 225
 - overview, 14–15
 - request management, 219–220, 226
 - requirements discipline, 166
 - and software development problems, 15
 - Classes, 177
 - Clemm, Geoffrey, 227
 - Close-out phase, 127
 - Close-out project, 127–128
 - Coarse-grained plans, 115–117
 - Code Reviewer role, 193
 - Collaboration diagrams. *See* workflow.
 - COM (component object model), 10
 - Common Object Request Broker Architecture (CORBA), 10
 - Component-based architecture, 9–11, 26–28
 - Component-based design, 179
 - Component-based development (CBD), 10, 93–94
 - Component design, 182–183
 - Component object model (COM), 10
 - Components
 - business, 94
 - definition, 27, 93–94
 - development, 94
 - runtime, 94
 - Concurrency, use cases, 102–103
 - Configuration and change management discipline. *See also* change management.
 - artifacts, 279
 - baselines, 225–226
 - CCM cube, 216–221
 - Change Control Board, 223
 - Change Control Manager role, 221
 - change-control plan, 224–225
 - change delivery, 225
 - change-management environment, 225
 - change request management, 219–220, 226
 - change requests, 223
 - Configuration Audit Findings, 223
 - configuration management, 217–219
 - configuration management plan, 223
 - Configuration Manager role, 221
 - configuration status, 226
 - implementation model, 223
 - Implementer role, 222
 - Integrator role, 222
 - introduction, 44–45
 - project configuration plan, 224–225
 - purpose, 215–216
 - releases, 225–226
 - roles, 221–223
 - in the RUP, 29
 - Software Architect role, 222
 - Software Configuration Strategies...*, 227
 - status and measurement, 220–221
 - tool support, 226–227
 - trend reports, 221
 - UCM (Unified Change Management), 227
 - workflow, 223–226
 - Configuration Audit Findings, 223
 - Configuration management, 217–219
 - Configuration management plan, 223
 - Configuration Manager role, 221, 275
 - Configuration status, 226
 - Configuration tests, 203
 - Configuring the RUP, 30–31, 230–231
 - Construction phase
 - activities, 73
 - artifacts, 73
 - definition, 63
 - evaluation criteria, 73
 - importance of, 66
 - iteration plan example, 255–258

- Construction phase (*continued*)
 - milestones, 73
 - objectives, 72–73
 - project management iteration plan, 137
 - purpose of, 72
- CORBA (Common Object Request Broker Architecture), 10
- Core disciplines, 44–45
- Core workflow, 47
- Course Developer role, 276
- Coverage, 134–136
- Criticality, 134–136
- Customizing the RUP, 231–232

- D**
- Data models, 177
- Database design, 183–184
- Database Designer role, 176, 274
- Deliverables. *See* artifacts.
- Delivering software to users. *See* deployment.
- Deployment discipline
 - acceptance test, 244
 - access to download site, 245
 - architecture, 88–89
 - artifacts, 239–241, 279
 - beta testing, 242, 244
 - introduction, 44–45
 - modes, 238–239
 - packaging, 244
 - planning, 242
 - product deployment unit, 242
 - purpose, 237–239
 - roles, 239–241
 - support material, 242
 - timing, 239
 - workflow, 242–245
- Deployment Manager role, 275
- Deployment modes, 238–239
- Deployment set, 43
- Design confirmation, and the waterfall process, 57
- Design Model, 176–177
- Design Reviewer role, 176
- Design set, 43
- Design vs. architecture, 89–90
- Designer role, 175–176, 274
- Developer roles, 274
- Developing software. *See* software development.
- Development Case, 231, 233
- Development components, 94
- Development cycle, waterfall process, 63
- Development processes
 - See* architecture-centric process
 - See* CBD
 - See* iterative process
 - See* use-case-driven process
 - See* waterfall process
- Dialogs. *See* scenarios.
- Disciplines, 44–45
 - See also* activities
 - See also* analysis and design
 - See also* business modeling
 - See also* configuration and change management
 - See also* deployment
 - See also* environment
 - See also* implementation
 - See also* project management
 - See also* requirements
 - See also* test
- Distributing software to users. *See* deployment.
- Documents. *See also* reports.
 - as artifacts, 41
 - in the RUP, 41–42
 - waterfall process, 59
- Domain modeling, 145

- E**
- E-business, 142–143
- EJB (Enterprise JavaBeans), 10
- Elaboration phase
 - activities, 70–71
 - artifacts, 71
 - definition, 63
 - evaluation criteria, 71–72
 - importance of, 65
 - iteration plan example, 250–254
 - milestones, 71–72
 - objectives, 70
 - project management iteration plan, 134–136
 - purpose, 69
- Enterprise JavaBeans (EJB), 10
- Entity classes, 153
- Environment discipline
 - artifacts, 232–233, 280
 - Business Process Analyst role, 233
 - configuring the RUP, 230–231
 - customizing the RUP, 231–232
 - Development Case, 231, 233

- implementing the RUP, 232
 - instantiating the RUP, 231
 - introduction, 44–45
 - iteration preparation, 235
 - Process Views, 231
 - project preparation, 234
 - Project Web Site, 231
 - purpose, 229–232
 - roles, 232–233
 - RUP Base, 230
 - RUP Library, 230
 - RUP Plug-In, 230–231
 - RUP Process Component, 230
 - RUP Process Configuration, 230
 - Software Architect role, 233
 - Structural Plug-In, 232
 - System Administrator role, 233
 - System Analyst role, 233
 - Technical Writer role, 233
 - Thin Plug-In, 231–232
 - Tool Specialist role, 233
 - tool support, 236
 - User-Interface Designer role, 233
- Evaluating
- construction phase, 73
 - elaboration phase, 71–72
 - inception phase, 69
 - RIP implementation, 268
 - test discipline mission, 208
 - transition phase, 75
- Events, 179
- Evolution cycles, waterfall process, 63
- Evolutionary prototypes, 190, 192
- Evolving use cases, 104–105
- Exploratory prototypes, 190, 191
- Extending use cases, 106
- F**
- Fine-grained plans, 115–117
 - Flow charts. *See* workflow.
 - Flow of events, 100–101, 106.
 - See also* workflow.
 - 4 + 1 view architecture model, 87–89
 - Functional requirements, 158
 - Functional tests, 203
 - FURPS quality attributes, 158
- G**
- Gilb, Tom, 7
 - Glossary artifact, 169
- Grady, Robert, 158
- Graphic Artist role, 276
- Groups in the RUP model. *See* actors; roles.
- Growing information sets, 43–44
- Guidelines, 48–50. *See also* tool mentors.
- H**
- Heuristics. *See* guidelines.
- I**
- IBM, history of RUP, 32–33
 - Implementation discipline
 - artifacts, 278–279
 - builds, 188
 - Code Reviewer role, 193
 - Implementation Elements, 193
 - Implementation Subsystem, 193
 - Implementer role, 192
 - incremental integration, 189
 - integration, 188–189
 - Integration Build Plan, 193
 - introduction, 44–45
 - phased integration, 189
 - prototypes, 190–192
 - purpose, 187–188
 - roles, 192–193
 - round-trip engineering, 195
 - Software Architect role, 193
 - System Integrator role, 192
 - tool support, 196
 - use-case-driven process, 107
 - workflow, 193–195
 - Implementation Elements, 193
 - Implementation model, 223
 - Implementation set, 43
 - Implementation Subsystem, 193
 - Implementation view, 87–88
 - Implementer role
 - configuration and change management, 222
 - definition, 274
 - implementation, 192
 - Implementing a system, example, 255–258
 - Implementing the Rational Unified Process...*, 260
 - Implementing the RUP
 - current state assessment, 262–263
 - effects of, 260–262
 - environment discipline, 232
 - evaluating the process, 268
 - executing the process, 268

Implementing the RUP (*continued*)
 identifying risks, 264–265
 introduction, 259–260
 planning the process, 266–268
 process as project, 269–270
 setting or revising goals, 263–264

Inception phase
 activities, 67
 artifacts, 67–68
 definition, 62–63
 evaluation criteria, 69
 importance of, 65
 iteration plan example, 248–250
 milestones, 69
 objectives, 67
 purpose, 67

Incremental development. *See* iterative process.

Incremental integration, 189

Individuals in the RUP model. *See* actors; roles; Stakeholders.

Information sets, 42–44

Initial development cycle, waterfall process, 63

Inputs to the RUP. *See* artifacts.

Installation tests, 203

Instantiating the RUP, 231

Integration, 188–189

Integration Build Plan, 193

Integration testing, 202–203

Integrator role, 222, 274

Integrity tests, 203

Interface, user. *See* user interface.

Interfaces, analysis and design, 178

Iterations

 duration, 130–132

 management, 127

 number of, 132–133

 plans. *See* project management discipline, iteration plans.

 preparation, environment discipline, 235

Iterative lifecycle, 199–201

Iterative process

 applied to waterfall process, 60–61, 65–66

 benefits of, 76–78

 best practices, 6–8, 23–25

 diagram, 7

 and software development problems, 8

Iterative projects, 114–117

J

Jacobson, Ivar, 33, 260

L

Load tests, 203

Logical view, 87

M

Management Reviewer role, 275

Management set, 43

Manager roles, 274–275

Measurement

 goals, 120–121

 measures, 121–122

 primitive measures, 121–122

 project management discipline, 119–122

 purpose, 119

Measures, 121–122

Milestones, waterfall process, 62–65

Modeling

 4 + 1 view architecture model, 87–89

 analysis model, 177–178

 architecture, 89–90

 business practices. *See* business modeling discipline.

 data model, 177

 Design Model, 176–177

Real-Time Object-Oriented Modeling, 179

 requirements discipline, 170

Modeling software. *See also* architecture.

 best practices, 11–13, 28

 diagram, 12

 importance of models, 81

 overview, 11–13

 and software development problems, 13

Monitoring the project, 126

N

Nonfunctional requirements, 159–160

O

Object Management Group (OMG), 10

ObjecTime, 32–33

Objectory AB, 32–33

Objectory Process, 33

Observable results of value, 99, 103–104

OMG (Object Management Group), 10

Organization charts, 145

Organizing use cases, 105–106

Output from the RUP. *See* artifacts.

P

Packages, 177

Packaging for deployment, 244

- People in the RUP model. *See* actors; roles; Stakeholders.
- Performance tests, 204
- PerformanceStudio, 213
- Performing steps, 39–40
- Phase (project) plans, 116, 128–133
- Phased integration, 189
- Phases of the RUP. *See* disciplines.
- Phases of the waterfall process
 - See* construction phase
 - See* elaboration phase
 - See* inception phase
 - See* transition phase
- Primitive measures, 121–122
- Problems in software development, solutions
 - change management, 15
 - common solutions to common problems, 95–96
 - component-based architecture, 10–11
 - iterative development, 8
 - modeling software, 13
 - quality verification, 14
 - requirements management, 8–9
- Problems in software development, symptoms and causes, 4–5. *See also* waterfall process, reasons for failure.
- Process Engineer role, 275
- Process framework, 17–18, 51
- Process product, 17
- Process view, 88
- Process Views, 231
- Processes, development
 - See* architecture-centric process
 - See* CBD
 - See* iterative process
 - See* use-case-driven process
 - See* waterfall process
- Processes involved in the RUP. *See* disciplines.
- Product deployment unit, 242
- Production roles, 275–276
- Programming guidelines, 50
- Progress assessment, waterfall process, 62–65
- Project configuration plan, 224–225
- Project management discipline
 - architecture, 93
 - artifacts, 122–123, 279–280
 - coarse-grained plans, 115–117
 - fine-grained plans, 115–117
 - introduction, 44–45
 - iterative projects, 114–117
 - measurement, 119–122
 - phase plans, 116
 - project plans, 116
 - purpose, 113–114
 - risk assessment, 117–119
 - roles, 122–123
 - two-level plans, 115–117
 - use-case-driven process, 108–109
- Project management discipline, iteration plans
 - activities, 138–139
 - architectural prototype, example, 250–254
 - artifacts, 138
 - business case definition, example, 248–250
 - construction phase, 137, 255–258
 - coverage, 134–136
 - criticality, 134–136
 - definition, 116
 - elaboration phase, 134–136, 250–254
 - examples, 247–258
 - implementing a system, example, 255–258
 - inception phase, example, 248–250
 - project vision definition, example, 248–250
 - risks, 134–136
 - steps in building, 134
 - transition phase, 137–138
 - workflow, 47
 - in the workflow, 126
- Project management discipline, workflow
 - close-out phase, 127
 - close-out project, 127–128
 - controlling the project, 126
 - diagram, 124
 - iteration duration, 130–132
 - iteration management, 127
 - iteration plans, 47
 - iterations, number of, 132–133
 - monitoring the project, 126
 - phase (project) plan, 128–133
 - project conception, 124–125
 - risk assessment, 125
 - rubber profile, 128–130
 - scheduling, 128
 - scoping, 125, 128
 - software development plan, 125–126
 - staffing, 128
- Project Manager role, 274
- Project preparation, environment discipline, 234
- Project vision definition, example, 248–250
- Project Web Site, 231
- Protocols, 179
- Prototypes, 190–192. *See also* storyboards.

Pure-Atria, 32–33
 PureCoverage, 213
 Purify, 213
 PurifyPlus, 213

Q

Quality of software
 benefits of the iterative process, 78
 necessary attributes, 158
 verification, 13–14
 Quantify, 213

R

Råberg, Lotta, 260
 Rational Objectory Process (ROP), 32–33
 Rational PerformanceStudio, 213
 Rational PureCoverage, 213
 Rational Purify, 213
 Rational PurifyPlus, 213
 Rational Quantify, 213
 Rational Software Corporation, 32–33
 Rational Test Realtime, 212
 Rational TestStudio, 212
 Rational Unified Process (RUP). *See* RUP
 (Rational Unified Process).
 Rational XDE Tester, 212
 Real-Time Object-Oriented Method, 32–33
Real-Time Object-Oriented Modeling, 179
 Real-time systems, 178–179
 Recommendations. *See* guidelines.
 Regression tests, 204
 Releases, 225–226
 Reliability, requirements, 159–160
 Reports, 42, 221. *See also* documents.
 Requirements
 capturing, 163–164
 changes, and the waterfall process, 55–57
 definition, 8–9, 158–160
 management, 8–9, 25–26, 163–164
 nonfunctional, 159–160
 performance, 159–160
 problem analysis, 166
 reliability, 159–160
 software, 162–163
 sources, business modeling, 153
 supportability, 159–160
 traceability, 163, 170
 types of, 160–163
 usability, 159–160
 Requirements discipline

artifacts, 168–170, 277–278
 automatic document generation, 170
 capturing requirements, 163–164
 change management, 166
 functional requirements, 158
 Glossary, 169
 introduction, 44–45
 managing requirements, 163–164
 nonfunctional requirements, 159–160
 performance, 159–160
 problem analysis, 166
 purpose, 157–158
 reliability, 159–160
 Requirements Reviewer role, 168
 Requirements Specifier role, 167–168
 requirements traceability, 163, 170
 roles, 166–168
 scoping the project, 166
 Software Architect role, 168
 software requirements, 162–163
 SRS (Software Requirements Specification),
 169
 Stakeholders, requests vs. needs, 160–161,
 164, 166, 168
 Storyboards, 169–170
 Supplementary Specifications, 169
 supportability, 159–160
 System Analyst role, 166
 system definition, 166
 system features, 161–162
 tool support, 170
 types of requirements, 160–163
 usability, 159–160
 Use-Case Model, 169
 use cases, 162–163, 164
 User-Interface Prototypes, 169–170
 Vision Document, 163, 168–169
 visual modeling, 170
 workflow, 164–166
 Requirements Reviewer role, 168
 Requirements Specifier role, 167–168,
 273–274
 Requisites, Inc., 32–33
 Resource planning, business modeling, 153
 Reusability, 78, 92–93
 Review Coordinator role, 276
 Reviewer role, 276
 Reviewing steps, 39–40
 Risks
 acceptance, 119

- assessment, waterfall process, 57–58
 - avoidance, 119
 - contingency plans, 119
 - coping with, 119
 - direct vs. indirect, 118
 - impact, 118
 - implementing the RUP, 264–265
 - known vs. unknown, 117–118
 - magnitude indicator, 118
 - mitigation, 76–77, 119
 - probability, 118
 - project management, 117–119, 125, 134–136
 - transfer, 119
 - Role: prefix, 38
 - Roles. *See also* actors; *entries for specific roles.*
 - analysis and design discipline, 175–176
 - business modeling discipline, 146–148
 - categories of, 37–38
 - configuration and change management, 221–223
 - definition, 36
 - deployment, 239–241
 - environment discipline, 232–233
 - examples of, 36–37
 - identifying in RUP, 38
 - implementation discipline, 192–193
 - project management discipline, 122–123
 - requirements discipline, 166–168
 - summary of, 273–276
 - test discipline, 204–207
 - work performed by. *See* activities.
 - ROP (Rational Objectory Process), 32–33
 - Round-trip engineering, 174, 195
 - Rubber profile, 128–130
 - Rules. *See* guidelines.
 - Runtime components, 94
 - RUP Base, 230
 - RUP Library, 230
 - RUP Plug-In, 230–231
 - RUP Process Component, 230
 - RUP Process Configuration, 230
 - RUP (Rational Unified Process). *See also* implementing the RUP.
 - best practices. *See* best practices.
 - configuring, 30–31, 230–231
 - customizing, 231–232
 - definition, 17–18
 - history of, 32–33
 - inputs and outputs. *See* artifacts.
 - instantiating, 231
 - primary elements
 - See* activities
 - See* artifacts
 - See* disciplines
 - See* roles
 - See* workflow
 - as process framework, 17–18, 51
 - as process product, 17
 - processes involved in. *See* disciplines.
 - purpose of, 15–16
 - as software engineering process, 17
 - typical users, 31–32
- S**
- Sanity check, 210
 - Scenarios, 102, 145–146
 - Scheduling, 60, 128
 - Scoping the project, 125, 128, 166
 - Sequence diagrams. *See* workflow.
 - Sequential process. *See* waterfall process.
 - Signals, 179
 - Smoke test, 210
 - Software Architect role
 - analysis and design discipline, 175
 - configuration and change management discipline, 222
 - definition, 274
 - environment discipline, 233
 - implementation discipline, 193
 - requirements discipline, 168
 - Software Architecture Documents, 176
 - Software Configuration Strategies...*, 227
 - Software Corporation, 32–33
 - Software development
 - best practices. *See* best practices.
 - modeling. *See* architecture; modeling software.
 - plans, 125–126
 - problems. *See* problems in software development.
 - quality. *See* quality.
 - requirements. *See* requirements.
 - roles of, 15
 - Software development business, modeling, 155
 - Software development processes
 - See* architecture-centric process
 - See* CBD (component-based development)
 - See* iterative process
 - See* use-case-driven process
 - See* waterfall process

- Software engineering process, 17
- Software engineering techniques, business modeling, 143–145
- Software generations, waterfall process, 63
- Software Process Engineering Metamodel (SPEM)*, 17
- Software Requirements Specification (SRS), 169
- Specializing use cases, 106
- SQA, Inc., 32–33
- SRS (Software Requirements Specification), 169
- Staffing, project management, 128
- Stages in the RUP. *See* disciplines.
- Stakeholder role, 276
- Stakeholders
 - business modeling discipline, 147
 - requests vs. needs, 160–161, 164, 166, 168
- Storyboards, 169–170. *See also* prototypes.
- Stress tests, 204
- Structural Plug-In, 232
- Structural prototypes, 190, 191
- Supplementary Business Specifications, 148
- Supplementary Specifications, 169
- Support roles, 275–276
- Supportability, requirements, 159–160
- Supporting disciplines, 44–45
- Symptoms of software development problems, 4–5
- System Administrator role, 233, 275–276
- System Analyst role, 166, 233, 273
- System definition, requirements discipline, 166
- System features, requirements discipline, 161–162
- System Integrator role, 192
- The system performs, use-case-driven process, 99
- System testing, 202–203
- T**
- Target-Organization Assessment, 147
- Technical disciplines, 44–45
- Technical Writer role, 233, 276
- Templates, definition, 50
- Test Analyst role, 204–205, 275
- Test approach verification, 208–210
- Test Cases, 205
- Test Configurations, 205
- Test Designer role, 205, 275
- Test discipline
 - acceptance into testing, 210
 - acceptance testing, 202–203
 - artifacts, 204–207, 279
 - avoiding extremes, 198
 - benchmark test, 203
 - build regression test, 210
 - build stability, 210
 - build verification test, 210
 - configuration test, 203
 - delivering results, 211
 - functional test, 203
 - improving test assets, 211–212
 - installation test, 203
 - integration testing, 202–203
 - integrity test, 203
 - introduction, 44–45
 - iterative lifecycle, 199–201
 - key issues, 198–199
 - levels of testing, 202–203
 - load test, 203
 - mission evaluation, 208
 - performance test, 204
 - progress evaluation, 210
 - purpose, 197–199
 - regression test, 204
 - roles, 204–207
 - sanity check, 210
 - smoke test, 210
 - stress test, 204
 - system testing, 202–203
 - Test Analyst role, 204–205
 - test approach verification, 208–210
 - Test Cases, 205
 - Test Configurations, 205
 - Test Designer role, 205
 - Test Evaluation Summary, 207
 - Test Ideas List, 205
 - Test Log, 207
 - Test Manager role, 204
 - Test Plan, 205
 - Test Results, 207
 - Test Scripts, 205
 - Test Suites, 205
 - Tester role, 205
 - tool support, 212–213
 - types of tests, 203–204
 - unit testing, 202–203
 - use-case-driven process, 108
 - workflow, 208
 - Workload Analysis Model, 205–207
- Test discipline, quality
 - aspects to be tested, 201–202

- definition, 200
 - functionality, 201
 - performance, 201–202
 - reliability, 201
 - responsibility for, 200–201
 - usability, 202
- Test Evaluation Summary, 207
- Test Ideas List, 205
- Test Log, 207
- Test Manager role, 204, 275
- Test Plan, 205
- Test Realtime, 212
- Test Results, 207
- Test Scripts, 205
- Test Suites, 205
- Tester role, 205, 275
- Tester roles, 275
- TestStudio, 212
- Thin Plug-In, 231–232
- Thinking steps, 39–40
- Throwaway prototypes, 190
- Time scale, and the waterfall process, 58–59
- Tool mentors, 50. *See also* guidelines.
- Tool Specialist role, 233, 276
- Tool support
- analysis and design discipline, 184
 - business modeling, 156
 - configuration and change management, 226–227
 - environment discipline, 236
 - implementation discipline, 196
 - requirements discipline, 170
 - software development, 212
 - test discipline, 212–213
- Traceability, requirements, 163, 170
- Transition phase
- activities, 75
 - definition, 63
 - evaluation criteria, 75
 - importance of, 66
 - milestones, 75
 - objectives, 75
 - project management iteration plan, 137–138
 - purpose of, 74
- Trend reports, 221
- Two-level plans, 115–117
- U**
- UCM (Unified Change Management), 227
- UML (Unified Modeling Language), 28
- Unified Change Management (UCM), 227
- Unified Modeling Language (UML), 28
- Unit testing, 202–203
- Usability, requirements, 159–160
- Use-case class, 102
- Use-case-driven process
- actions, 98
 - actors, 98
 - analysis and design, 107
 - business modeling, 109
 - flow of events, 100–101
 - implementation, 107
 - observable results of value, 99, 103–104
 - particular actors, 99
 - paths of action, 101
 - project management, 108–109
 - in the RUP, 29–30
 - scenarios, 102
 - sequence of actions, 98–99
 - the system performs, 99
 - testing, 108
- Use-Case Model, 102–103, 169
- Use-case package, 89
- Use-case view, 89
- Use cases
- concurrency, 102–103
 - definition, 98
 - diagram, 108
 - evolving, 104–105
 - examples, 99–100
 - extending, 106
 - identifying, 103–104
 - including common flow of events, 106
 - interactions between, 102–103
 - organizing, 105–106
 - requirements discipline, 162–163, 164
 - specializing, 106
- User interface
- design, 176–177
 - guidelines, 49
 - prototyping, 169–170, 177
- User-Interface Designer role, 233, 274
- User-Interface Prototypes, 169–170
- Users, 31–32. *See also* Stakeholders.
- V**
- Version control, artifacts, 41
- Views, architecture, 86–87, 89–90
- Vision definition, example, 248–250
- Vision Document, 163, 168–169
- Visually modeling. *See* modeling.

W**Waterfall process**

- breaking into manageable chunks, 60–61
- development cycle, 63
- evolution cycles, 63
- importance of activities, 65–66
- initial development cycle, 63
- iterative approach, 60–61, 65–66
- vs. iterative process, 6–8
- milestones, 62–65
- progress assessment, 62–65
- and project size, 58
- sequential process (diagram), 54
- software generations, 63
- successful use of, 57–58
- typical steps, 53–54

Waterfall process, construction phase

- activities, 73
- artifacts, 73
- definition, 63
- evaluation criteria, 73
- importance of, 66
- milestones, 73
- objectives, 72–73
- purpose of, 72

Waterfall process, elaboration phase

- activities, 70–71
- artifacts, 71
- definition, 63
- evaluation criteria, 71–72
- importance of, 65
- milestones, 71–72
- objectives, 70
- purpose, 69

Waterfall process, inception phase

- activities, 67
- artifacts, 67–68
- definition, 62–63
- evaluation criteria, 69
- importance of, 65
- milestones, 69
- objectives, 67
- purpose, 67

Waterfall process, reasons for failure. *See also* problems in software development.

- design confirmation, 57
- paper documentation, 59
- requirements changes, 55–57
- risk assessment, 57–58
- scheduling, 60
- time scale, 58–59

Waterfall process, transition phase

- activities, 75
- definition, 63
- evaluation criteria, 75
- importance of, 66
- milestones, 75
- objectives, 75
- purpose of, 74

White, Brian, 227**Work guidelines, 48****Work product. *See* artifacts.****Work units. *See* artifacts.****Workflow**

- analysis and design discipline, 180–184
- business modeling discipline, 148–150
- configuration and change management, 223–226
- core, 47
- definition, 45
- deployment, 242–245
- details, 47
- diagram, 46
- implementation discipline, 193–195
- iteration plans, 47
- project management discipline, 47
- requirements discipline, 164–166
- test discipline, 208
- types of, 47

Workflow, project management discipline, 124–133**Workload Analysis Model, 205–207****X****XDE Tester, 212**