
Index

A

- “A Day in the Life,” 25–29
- Accelerated builds, 250
- Acceptance tests. *See* Functional tests
- AccuRev, 234
- Afferent Coupling, 170–172, 240
- Agitator, 236
- Agitator Agitar One, 236
- Alienbrain, 234
- Amazon, 190
- Ambient Devices, 214, 241–242
- Ambient Orb, 214–215, 241–242
- Ambler, Scott, 109*n*
- Analysis tools, 37
- Ant, 68, 256
 - and ambientorb, 215
 - build difference report, 198
 - build scripts, 6–7, 10, 34, 53–54, 219
 - and Cargo, 19
 - and Checkstyle, 17–18, 169
 - and CPD, 177–180
 - database integration, 112
 - and JDepend, 172
 - and JUnit, 16, 140
 - and PMD, 175
 - scripts, 113–116
 - and Simian, 178–181
 - sql, 14, 113–116
- ant db:prepare, 112–113, 116
- ant deploy, 191
- AnthillPro, 229, 264–266
- Apache, 233
 - Continuum, 85, 229, 266–267
 - Gump, 232
 - Maven, 229, 233, 258
 - Maven 1, 256–258
 - Maven 2, 258–260
 - Tomcat server, 18–19
 - XML scripts, 232
 - See also* Ant
- Apache Java Enterprise Mail Server (Apache James), 210*n*, 242
- Appleton, Brad, 8, 74–75, 120
- Architects, and feedback, 208
- Architectural adherence, 59–60
- Artifact publication, 252
- assert, 132–138, 146, 157
- Asserts, in test cases, 156–157
- Asset labeling, 191–194
- Assumptions, 23–25, 30, 191
- Atlassian, 229
- Authentication, 252
- Automated
 - builds, 6, 66–69, 224, 255–263
 - code documentation tool, 57
 - code inspections, 17–18, 163
 - inspection resources, 60, 239–241
 - inspectors, 228
 - process, 27
 - queued integration builds, 223–224
 - regression testing, 37, 53–54, 237
 - testing, 15–16, 41–42, 44, 197
- Automation for the people, 227–228

B

- Bamboo, 229
- batchtest, 16, 140, 185
- Beck, Kent, 88, 250*n*
- Begel, Andrew, 177
- Berczuk, Stephen, 8, 74–75, 120
- “Big Ball of Mud,” 59
- Bitten, 232
- Booch, Grady, 36
- Borland, 85, 223*n*, 231, 234
- Branch coverage, 181

- Branching, 100–101
- Broken builds, 41, 44, 86
- Broken code, 39–44, 86
- Browser-based testing, 238
- Browser simulation, 136, 236
- Bug detection, 53, 239
- Build (CI step), 34–35
- build-database.xml, 112, 115–116
- .build extension, 261
- BuildBeat, 232
- BuildBot, 232
- BuildForge, 96, 230
- Build(s), 4, 27
 - automated, 6, 66–69, 224
 - broken, 41, 44, 86
 - delegating, 9
 - difference report, 198
 - execution, 250–251
 - failed, 32–33, 98, 213
 - feedback reports, 196–198
 - full builds, 67
 - incremental, 94
 - labels, 195–196, 251
 - life, 265
 - management tool, 230
 - mechanisms, 80–81
 - metrics, 88–89
 - performance, 87
 - private, 6–7, 10, 26–28, 41–44, 79, 99
 - scalability, 87
 - schedulers, 8–9, 250–252, 263–272
 - scripts, 10, 52, 70, 73–74, 228, 232–233
 - single command, 69–73
 - smell, 228
 - speed, 87–96
 - status, 43, 126, 206–207
 - success/failure trends, 31
 - tool integration, 251
 - tools, 10, 68, 248–250
 - triggering, 81
 - types, 78–81
- BVCs (big visual charts), 220

- C**
- C, 241, 243
- C#, 71, 230, 241, 243
- C++, 241, 243
- Capistrano (formerly SwitchTower), 241
- Cargo, 19
- Categorizing tests, 132, 138–140
- CCTray, 217–218
- Centralized software, 74–75
- Checkin branches, 231
- Checkstyle, 17–18, 58*n*, 169, 175, 228, 239
- ClearCase (SCM/version control tool), 8, 42*n*, 233, 266
- Clover, 180, 239
- Clover.NET, 180
- CM Crossroads, 232
- Cobertura, 180, 184, 239
- Cockburn, Alistair, 220
- Code
 - analysis tools, 37, 58*n*
 - audits, 173–176
 - compilation, 12–13, 248
 - coverage tool, 239, 240
 - and documentation, 20
 - documentation tool, 57
 - duplication, 239, 241
 - inspections, automated, 17–18
 - listeners, 183
 - metrics, 166–167, 170–172
 - metrics tool, 58*n*
 - quality analysis, 249
 - reuse, 176–180
 - smell, 57–58
- Code coverage, 27, 42, 54–55, 180–182, 184
- Codehaus, 259
- Coding standard, 37, 173–176
 - adherence, 58–59, 239
- Collateral damage effect, 170
- Command line, 6–7, 69, 112
- Commit build, 80
- Committing code frequently, 39–40, 44
- Compatibility, tools, 253
- Compilation, source code, 12–13
- Complexity reporting, 167–170
- component directory, 139–140
- Component packaging, 248
- Component tests, 134, 141–143
 - dbUnit, 134–135
 - length/speed to run, 142
 - repeatable, 148–156
- Concurrent Versions System (CVS), 8, 192, 198, 233, 266
- Confidence, 32

- Configuration files, 77–78
 - Continuous, 27
 - Continuous compilation, 35
 - Continuous Database Integration (CDBI), 107, 121–123
 - automating, 110–117
 - DBA on development team, 124
 - developer changes, 123
 - fixing broken builds, 124
 - integrate button, 125–126
 - local database sandbox, 117–119
 - version control repository, 119–121
 - Continuous deployment, 126, 189–191
 - build feedback reports, 196–198
 - build labels, 195–196
 - clean environment, 194–195
 - release rollback, 199
 - repository labels, 191–195
 - testing, 196
 - Continuous feedback, 203–209
 - Ambient Orb, 214–215
 - devices (CFDs), 205
 - e-mail, 210–212, 251
 - SMS (text messages), 56, 212–213, 217
 - sounds, 218–219
 - wide-screen monitors, 220–221
 - Windows task bar, 217–218
 - X10 devices, 216–217
 - Continuous inspection, 161–165
 - code audits, 173–176
 - code complexity, 167–170
 - code coverage, 180–182
 - code metrics, 166–167, 170–172
 - compared with testing, 164–165
 - design reviews, 170–172
 - duplicated code, 176–181
 - inspectors, 165–166
 - quality, 182–185
 - Continuous Integration, defined, 27
 - Continuous Integration Server Matrix, 230
 - Continuous-prevention development, 148
 - Continuum, 85, 229, 266–267
 - Copy/Paste Detector (CPD), 61, 177–180, 228
 - Coupling metrics, 170–172
 - Coverage frequency, 183–184
 - cron, 8, 81, 264
 - CRUD, 7, 144
 - CruiseControl, 230, 266–268
 - EMMA coverage report, 182
 - polling for changes, 8–9, 26
 - sending e-mail, 11, 56, 210–212
 - sounds, 219
 - web updates, 4
 - X10 devices, 216
 - CruiseControl config.xml, 8–9, 11
 - CruiseControl.NET, 217, 230, 268–269
 - CruiseControl.rb, 232
 - csc, 71
 - Cusumano, Michael A., 36
 - CVS (SCM/version control tool), 8, 192, 198, 233, 266
 - Cyclomatic complexity, 163
 - Cyclomatic Complexity Number (CCN), 167–169
- ## D
- D, 78
 - D (programming language), 243
 - Daily builds, 36, 228
 - 2003 study results, 66
 - Data Access Object (DAO), 135, 144–150, 153
 - Data Definition Language (DDL), 14, 114, 116
 - data-definition.sql, 112, 114, 116
 - Data Manipulation Language (DML), 14, 116
 - data-manipulation.sql, 112, 116
 - Data sources, 109
 - Database(s)
 - administration, 50–51
 - creation, 112–115
 - integration, 14–15
 - manipulation, 115–116
 - orchestration script, 116–117
 - resources, 234–235
 - sandbox, 117–119
 - scripts, 51
 - seeding, 116, 134–135, 143, 149, 154
 - server, 117
 - shared, 117–119
 - source code, 14
 - testing, 125
 - and version control repository, 50–51
 - See also* Continuous Database Integration (CDBI)
 - DBA, 110–112, 120, 123–124
 - db:create, 14, 112–113, 116
 - db:insert, 112–116
 - db:refresh, 127–128
 - DbUnit, 115–116, 149, 152, 236
 - component tests, 134–135

Debugging, xxiii, 53, 117, 133, 239
 Dedicated machines, 80–84, 90, 99–100
 Defect-driven development, 144–146
 Defect testing, 143–148
 Defects, 29–31, 57–58
 Delegating builds, 9, 219
 delete, 71
 Delphi, 241
 Dependency analysis tools, 60
 Deployable software, 31
 Deployment, 18–19
 to an FTP server, 73
 functionality, 249
 resources, 241
 Design reviews, 170–172
 Design smell, 57
 Developer testing, 37, 132, 138–140
 Developers, 6–7, 39–43, 123
 and feedback, 208
 modifying database scripts, 123
 and sandboxes, 117–119
 Development environment, 28
 Development test execution, 249
 Directory structure, 74–76, 120–121, 139–140
 Distributed integration builds, 96
 Documentation, 20
 Documentation generation, 249
 Documentation resources, 243
 Don't repeat yourself (DRY), 117
 Doxygen (code documentation tool), 57, 243
 Draco.NET, 230, 269–271
 driver, 14, 113–115
 Duplicated code, 60–62, 176–181
 Dynamic languages, 12–13

E

E-mail, 10–11, 55–56, 210–212, 251, 266
 Early implementation, 35–36
 Early integration, 39–40
 eBay, 190
 Eclipse, 259
 Efferent Coupling, 170–172, 240
 EMMA, 180–182, 239
 Entity Relationship Diagram (ERD), 120, 126
 Eudora, 210
 Event driven, 251

Event-driven build mechanism, 81
 Evolution of integration, 36–37
 Evolutionary Database Design, 109*n*
 Exceptions, 144–153
 Extensibility, 249
 Extract method technique, 169
Extreme Programming Explained, 88, 250*n*
 eXtreme Programming (XP), 24

F

Fagan inspection process, 162
 Failed builds, 32–33, 76–77
 failonerror, 72
 Fan In, 170–172
 Fan Out, 170–172
 Fast builds, 87–96
 Features (of CI), 12–20
 Feedback, 20, 24, 203–209, 251
 Ambient Orb, 214–215
 e-mail, 210–212, 251
 reports, 196–198
 resources, 241–242
 SMS (text messages), 56, 212–213, 217
 sounds, 218–219
 wide-screen monitors, 220–221
 Windows task bar, 217–218
 X10 devices, 216–217
 Feedback and documentation
 Continuous Database Integration (CDBI),
 126
 Feedback mechanism, 10–11
 See also Continuous feedback
 File manipulation, 248
 FindBugs, 239
 Firefox plug-in, 221
 Fit, 236
 FitNesse, 236
 Flickr, 190
 Floyd, 236
 Fowler, Martin, 27*n*, 37, 38*n*, 61*n*, 69, 80, 109*n*,
 166*n*, 169*n*, 228
 Frederick, Jeffrey, 228
 FTP, 268
 Full build, 67
 Functional tests, 137–138, 182, 237, 238
 FxCop, 72–73, 175, 240

G

Gaim, 242
 Gauntlet, 85, 223*n*, 231
 Google, 190
 GoogleTalk, 242
 Graham, Susan L, 177
 Groovy, 232
 Gump, 232

H

Hibernate, 142–147, 150–155
 Hibernate configuration utility, 150–152
 Hibernate test case, 154–156
 HSQLDB, 234
 HTML reports, 167–168, 172
 HtmlUnit, 236
 HttpUnit, 153–154
 Hunt, Andrew, 117
 Hypersonic DB, 234

I

IBM developerWorks articles, 18, 84, 227–229
 IBSC acrostic, 34–35
 IDE (Integrated Development Environment), 7, 10,
 73–74, 165–166
 Identify (CI step), 34–35
 IDL, 243
 Implementation directory, 76, 120–121
 Improvements, 89–96
 Incremental build, 94
 Information overload, 207–208, 211
 Information radiators, 220
 Inspection, 28, 42
 automated, 17–18, 239–241
 compared with testing, 164–165
 database integration, 125
 for duplicate code, 61
 resources, 239–241
 tools, 60
 See also Continuous inspection
 Inspectors, 165–166, 228
 Instability, 170–172, 240
 Instant messaging, 221, 242, 266
 Integrate button, 13

IntegrateButton.com, 229
 Integrated Development Environment (IDE), 7
 Integration, term, 28
 Integration build, 6, 8–9, 26, 28, 79–80, 88
 automated, 223–224
 distributed, 96
 manual, 86
 as nonevent, 13
 Integration build machine, 12–13, 33, 81–84,
 90–91, 122
 Integration test, 136
 Interproject dependencies, 252
 interval, 9
 Iterative projects, 24

J

Jabber, 242
 Java
 build tools, 68, 71
 and Checkstyle, 17–18
 Cobertura, 180, 184, 239
 JavaNCSS, 167–169, 228, 240
 PMD, 163, 177
 test cases, 236
 Java Coding Conventions on One Page, 58*n*
 Javadoc, 20, 243
 javaranch.com, 3
 Javascript, 177, 237
 JDepend, 60, 172, 240
 JetBrains, 223*n*
 Jetty, 18–19
 JIRA, 229
 JUnit, 15–16, 37, 180, 237
 and Ant, 16
 batchtest, 140, 185
 JWebUnit, 237
 system tests, 136–137

L

Labels
 build, 195–196, 251
 repository, 191–194
 Large projects, 97
 Lava lamps, 216–217, 242
 Lee, Kevin, 229

Legacy applications, 97
 Line coverage, 180
 Linux, 235
 Listeners, 183
 Local database sandbox, 117–119
 Lookup tables, 111, 115
 Luntbuild, 85, 231, 252, 271–272

M

Mac OS X, 221, 235
 “Magic machines,” 84
 Mainline, 79–80, 100–101
 make, 10, 85, 255–256
 “Make it continuous” (CI step), 34–35
 Manual deployment of software, 52–53
 Manual integration build, 86
 Manual processes, 32
 Manual reviews, 161–163
 Manual testing, 197
 Maven, 20, 71, 167–168, 181, 233
 Maven 1, 256–258
 Maven 2, 258–260
 McConnell, Steve, 36, 228
 Mckoi, 235
 Merge (Cobertura), 184
 Mergere, 259
 Meszaros, Gerard, 238
 Metrics tool, 58*n*
 Mevenide, 259
 Microsoft, 210, 234, 243, 261–262, 268–269
 MSBuild, 262
 Team Foundation Server (TFS), 223*n*
 Microsoft Outlook, 210
Microsoft Secrets, 36
 MKS (SCM/version control tool), 8, 233
 Mocks, 92, 133, 135, 154–155
 Mojo, 259
 MSBuild, 10
 Multiplatform builds, 249–250
 MySQL, 14, 235
 MySQL database, 114, 116

N

NAnt, 10, 34, 69, 85, 233
 build file, 261–262

 delete, 71
 FTP, 73
 fxcop, 72
 nunit2, 72
 nant integrate, 69
 NCover, 180, 240
 NDbUnit, 116, 143, 149, 237
 NDepend, 60, 171, 240
 NDoc, 20, 243
 .NET, 34, 233, 237
 build tools, 68
 and FxCop, 72–73
 NDbUnit, 143*n*
 NDepend, 171
 Simian, 178
 .NET Framework Design Guidelines, 240
 NetBeans, 259
 Noncommenting source statements (NCSS), 168
 NUnit, 15, 37, 72, 237
 nunit2, 72, 73

O

Object Solutions: Managing the Object-Oriented Project, 36
 Objective-C, 243
 On-demand build mechanism, 80
 Oracle, 235
 Oracle Express Edition, 235
 Oracle PL/SQL, 238
 O’Reilly, Tim, 190

P

Pair programming, 161–162
 ParaBuild, 96, 231
 password, 14, 113, 115
 Path coverage, 181
 PDbSeed, 149
 Peer code reviews, 161–162
 PerfectBuild, 232
 Perforce (SCM/version control tool), 8, 234, 266
 PHP, 12–13, 243
 Plug-ins, 249, 259
 PMD, 58, 61, 169, 174–176, 240
 PMD-CPD, 61, 177–178
 PMD report, 176

PMEase QuickBuild, 231
 Poll for changes, 81, 250–251
 PostgreSQL, 235
 Practices, tables of, 44, 101–102, 127, 158, 186, 200
 Pragmatic Automation, 232
Pragmatic Programmer, 117
 Private builds, 6–7, 10, 26–28, 41–44, 79, 99
 Program execution, 248
 Project Object Model (POM), 257, 259, 260
 project.xml, 257–258
 Pulse, 85, 223, 232
 PVCS (SCM/version control tool), 8, 234
 Python, 12–13, 149, 236, 243

Q

Quality assurance, 28, 131, 182–185
 Quality control, 25
 Quality Labs, 242

R

Rake, 10, 233, 262–263
 Rational Unified Process (RUP), 24
 RDBMS, 109, 117
 Refactoring, 37–38, 61*n*, 157, 169
Refactoring: Improving the Design of Existing Code, 38*n*, 169*n*
 Refactoring databases, 109
Refactoring Databases, 109
 Regression tests, 37, 53–54
 Release build, 28, 80
 Reliability, 129–132, 254
 Remote users, 98
 Repeatable component tests, 148–156
 Repetitive processes, reducing, 30–31
 Repository labels, 191–195
 Repository pattern, 75
 Resources

- automated inspection, 239–241
- build scripting, 232–233
- databases, 234–235
- documentation, 243
- feedback, 241–242
- testing, 236–238
- tools and products, 229–232

- version control, 233–234
- web sites and articles, 227–229

Reusable scripts, 114
 Reverse engineering, 56
 Risk, defined, 29
 Risk management, 47
 Risk reduction, 29–30, 47–49

- defects, 53–55
- project visibility, 55–57
- software quality, 57–61
- software readiness, 49–53

 Rollbacks, 18, 43, 192, 199
 root directory, 139
 RSS, 10, 221
 Ruby, 12–13, 241, 262–263

- Rake, 233, 262
- unit testing, 133

 Ruby on Rails, 241

S

Sadalage, Pramod, 109*n*
 Sandbox, 117–119, 127, 235
 Sandboxing, 231
 Scheduled build mechanism, 80–81
 Scheduling builds, 8–9
 scm:update, 127
 Scripts

- Ant, 6–7, 10, 34, 53–54, 219
- build, 10, 52, 70, 73–74, 228, 232–233
- maintaining, 121
- reusable, 114
- SQL, 71–72, 112–116

 Secondary builds, 80
 Secure Copy (SCP), 268
 Security, 72, 81, 98, 252
 Seeding, 116, 134–135, 143, 149, 154
 Selby, Richard W, 36
 Selenium, 136–138, 237
 Server matrix, 230
 Servers, 5–9

- Continuum, 266
- CruiseControl, 50, 266–268
- CruiseControl.NET, 268
- Draco.NET, 269
- features of, 85
- lifespan, 254–255

Servers *continued*

- Luntbuild, 271
 - and Maven, 260
- set explain, 125
- Setup time, 38–39
- Share (CI step), 34–35
- Shared databases, 117–119
- Sierra, Kathy, 3
- Simian, 61, 178–181, 241
- Similarity Analyser, 61, 178–181, 241
- Sin (Continuous Integration for Subversion), 231
- Single command builds, 69–73
- SMS (text messages), 10, 56, 212–213, 217
- SMTP server, 213
- SnapshotCM, 234
- SOAP, 266
- Software
 - assets, 74–75, 83
 - build, 67–69
 - delivery, 49–52
 - inspection, 28, 95
 - manual deployment of, 52–53
- Software-build management server, 231
- Software Configuration Management Patterns*, 8, 74–75, 120
- Software Configuration Management (SCM)
 - tools, 8
- Software Project Survival Guide*, 36
- Sounds, 218–219
- SourceForge, 269–272
- SourceMonitor, 241
- SQL, 125, 235
- SQL scripts, 71–72, 112–116
- sql task, 113
- SQLUnit, 238
- src directory, 139
- Staged builds, 80, 88, 92
- StarTeam, 234, 266
- Statement coverage, 180
- Static analysis tool, 58*n*, 61, 162–163
- Static code analyzer, 240
- Status reports, 31
- Struts, 153
- Struts test case, 154–156
- StrutsTestCase, 135, 154–155
- Subsystem tests. *See* Component tests
- Subsystems, 94–95
- Subversion, 7–9, 26, 234, 266
- Surround SCM, 234

- Sybase, 235
- Synching with the database, 50–52
- Synergy, 234
- system directory, 139–140
- System tests, 136–137, 143

T

- Task branch, 120
- Team Foundation Server (TFS), 223*n*
- TeamCity, 223*n*, 232
- Ten-minute builds, 88
- Terms of the trade, 27–29
- Test coverage, 54–55
- Test-pass thresholds, 197
- TestEarly.com, 238
- Testing, 15–16, 91–92, 129–132
 - compared with inspection, 164–165
 - component tests, 134–136, 141
 - Continuous Database Integration (CDBI), 125
 - for defects, 143–148
 - developer tests, 138–140
 - functional tests, 137–138
 - repeatable component tests, 148–156
 - resources, 236–238
 - system tests, 136–137, 143
 - test cases, 156–157, 169, 236
 - unit tests, 132–133, 141
 - using NUnit and NAnt, 72
- Testing (term), 29
- TestNG, 132, 139, 238
- Text messages (SMS), 56, 212–213, 217
- Thomas, David, 117
- ThoughtWorks, 230, 232, 266–268
- Tinderbox, 232
- Tomcat server, 18–19
- Tools, evaluating, 245–248
 - automated build tools, 255–263
 - build schedulers, 250–252, 263–272
 - build tools, 248–250
 - compatibility, 253
 - longevity, 254–255
 - reliability, 254
 - usability, 255
- Tools and product resources, 229–232
- Toomim, Michael, 177
- Trends, build success/failure, 31
- Trunk, 79–80, 100–101

U

- unit directory, 139–140
- Unit testing, 53, 132, 237
 - and Ant build scripts, 54
 - length/speed of test, 141
 - Ruby, 133
- UNIX, 8, 233, 235, 243, 245–246
- Urbancode, 264–266
- User interface, 252
- userid, 113
- utPLSQL, 238

V

- Version control, 75–76
 - integration, 251
 - resources, 233–234
 - systems, 8, 85
 - tool integration, 249
 - See also* Subversion
- Version control repository, 6–8, 50
 - and CDBI, 119–121
 - checking for changes, 8–9
 - and databases, 14–15, 50–51
 - directory structure, 75–76
- Visual Basic, 241, 261
- Visual SourceSafe (SCM/version control tool), 8, 234

W

- Watir, 238
- Web site login, 136–137
- Web sites, and testing, 137
- Wide-screen monitors, 220–221
- Widgets, 221
- Windows, 235
- Windows task bar, 217–218
- Windows Task Scheduler, 8

X

- X10, 242
- X10 devices, 216–217
- XML, 134, 143, 177
- XML build file, 261
- .xml files, 77
- XML reports, 167, 172, 175, 178
- XML-RPC, 266
- XML seed files, 149
- XP, 36–37
- XSD, 177
- xslDIR, 213
- xslfile, 213
- XSLT, 179–180
- xUnit, 15, 37, 41, 54
- xUnit Test Patterns, 238