## Index

Note to the Reader: Throughout this index, **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

### A
- A to Z Project Billing tool, 13, 14
- Acceptance testing, 71
- AccuRev software, 38, 39
- Active code generators, 145–146
- Add command, 34
- Aggregation application block, 134
- Agile methodologies, 16–17, 17
- AlachiSoft TierDeveloper tool, 149
- Albahari, Ben, 133
- Alexander, Christopher, 24
- Alhir, Sinan Si, 23
- AllNetic Working Time Tracker, 13
- Alpha builds, 29
- Antechinus C# Editor, 102
- Application blocks, 26, 134–135, 189
- Application code. See code; coding; source code
- Application delivery. See delivering application logs. See logging architecture, EIF, 186, 186 articles. See websites
- ASP.NET Web Matrix IDE, 102
- Assertions, See also coding defensively defined, 52
  - Guidelines for, 55–56
  - Mechanics of, 53–54, 54
  - In .NET, 52–56, 53–54
  - In other languages, 55
  - Overview, 52
  - Uses for, 56
- Asynchronous Invocation application block, 134
- Atozed Software A to Z Project Billing, 13, 14
- Atozed Software FinalBuilder, 241, 241–243
- AutoIt tool, 252

### B
- Beck, Kent, 17, 72, 85
- Beta testing, 29, 71
- Billing and time tools, 13–14, 14
- BitKeeper software, 38, 39
- Black-box testing, 164
- Boggs, Michael, 23
- Boggs, Wendy, 23
- Branch command, 36–37, 36–37
- Branching, 47–48
- Breadth-first coding, 27
- Brooks, Fred, 11
- BSD (Berkeley Source Distribution) licenses, 262
- Bug tracking, 156–173, See also debug checklist, 173
  - Excess process in, 170
  - Overview, 156
  - Politics of, 162–163
  - With quality assurance testing, See also testing; unit testing; black-box testing, 164
  - Compatibility testing, 164
  - Conformance testing, 163–164
  - Functional testing, 163
  - On networked computers, 166–168
  - Overview, 163
  - Performance testing, 164
  - Regression testing, 164
  - Smoke testing, 164
  - Stress testing, 164
  - Unit testing, 163
  - When working solo, 165–166
  - White-box testing, 164
- By design, 161
- By fixing, 162
- By not reproducible, 161–162
- By postponing, 161
- By reassigning, 162
- As won’t fix, 162
- Risk management and bug triage, 160–163
- Overview, 156
- Risk assessment, 156–158
- Risk control, 158–159
- Steps in, 158–160
- Top five risks lists, 159–160
- Tools
  - Choosing, 169–171
  - FogBUGZ, 171–173, 172
  - Overview, 169
  - Workflow in, 162
- Building code. See code; coding; source code
- Building Secure Assemblies article, 188
- Buzan, Tony, 4

### C
- C# in a Nutshell (Drayton, Albahari, and Neward), 133
- C# refactoring tools, 92
- C#Builder IDE, 102
- Caching application block, 134
- Camel casing, 137
- Check In command, 35
- Check Out command, 34
- Check-out/edit/check-in development style, 42–43
- ClearCase software, 38, 39
- Cloak command, 37
coding software installers—Demeanor tool

end-user help files
choosing topics, 218–220, 219
conceptual help, 217
context-sensitive help, 218
using FAR, 223
indexes, 217–218, 221
using Microsoft HTML Help, 222, 223
organizing information, 220–221
overview, 216
using professionals, 221, 224
using RoboHelp, 223, 224
task-specific help, 218, 219
and testing, 221
understanding user needs, 216–218
with user feedback links, 222, 222
user interface and, 225
manuals, 229–230
overview, 216
team documentation
development logs, 230–231
overview, 230
postmortems, 231–232
coding software installers, See also delivering
choosing install creators, 280–281
example, 282–286, 284–286
including licenses, 283
using Inno Setup, 277
install-on-demand, 280
using InstallAnywhere.NET, 277–278
using InstallShield DevStudio, 278, 279, 279
using InstallShield Express, 278
using NSIS, 278
overview, 276–277
and testing, 282
tips for, 281–282, 282
and uninstallers, 282
Windows Installer resources on, 279, 280
using Wise for VS .NET, 278–279
using Wise for Windows Installer, 278–279
coding with small teams, 198–213
checklist, 213
versus coding solo, 199
and documenting
development logs, 230–231
overview, 230
postmortems, 231–232
legalities, 199
managing teams
choosing team structures, 200
overview, 199–200
the Peter Principle and, 201–202
processes, 202
tracking progress, 200–201
overview, 198
reasons for, 198
tools for distributed teams
CodeMeeting, 212
CodeReview, 213
CodeWright, 212
e-mail, 203
GetDotNet, 206–207, 208
Groove Workspace, 211, 212
instant messages, 204
online workspaces, 204–206, 206, 208
overview, 203
SharePoint Services, 210–211
SourceForge, 205, 206
wikis, 209, 210
comments, See also coding defensively
intent comments, 65
noise comments, 61–62
overview, 61
placeholder comments, 63, 63
summary comments, 64–65
synchronization problem, 66
XML comment tags, 225–228, 227
compatibility testing, 164
Compuware DevPartner Studio add-in, 116, 117
ConceptDraw MINDMAP software, 5
concurrent development style, 42–43
Configuration Management
application block, 134
conformance testing, 163–164
continuous integration, 250–251
contracts, 256–259, See also protecting
Create Project command, 34
Creative Commons project, 261
CruiseControl.NET tool, 251
csUnit testing tool, 73
“Customizing VS .NET Start Page” article, 98
CVS (Concurrent Versions System) software, 34, 38, 39–40
“Deploying .NET Framework-Based Applications” article—KVM switches

“Deploying .NET Framework-Based Applications” article, 288

depth-first coding, 27

Design Patterns: Elements of Reusable Object-Oriented Software (Gang of Four), 24
design patterns, See also code project organization
benefits, 24–25
defined, 23–24

Gang of Four patterns, 24–25
Microsoft patterns, 25–26
developers
coding help files for, 225–228, 227–228
defined, xix
handling idiosyncracies of, 252
testing code for, 71–72

Developing an Automated Software Testing Tool (Li and Wu), 71
development logs, 230–231
DevPartner Studio add-in, 116, 117
diagnostic tools, See also logging
overview, 193
Product Support Reporting Tools, 193–194
System Information utility, 193, 193
domain, public, 260
domains, 168
dotEasy tool, 92
Dotfuscator tool, 128, 267–272, 269–270
Draco.NET tool, 251
Drayton, Peter, 133
Dynamics of Software Development (McCarthy), 202
e-mail, as team tool, 203
Eclipse IDE, 103
EIF (Enterprise Instrumentation Framework), See also logging
architecture, 186, 186
configuring, 188–189
defined, 185
event sinks, 186, 186, 187–188
event sources, 186–187, 186
elevator pitches, 3–4
Enterprise Architect software, 22
EventLog class, 182–185, 184
Exception Management application block, 134
exceptions, See also coding defensively
custom exceptions, 58–60
defined, 56
even, 56–57
guidelines for, 60–61
mechanics of in C#, 57
testing with NUnit tool, 80
Extreme Programming Explained (Beck), 17
Extreme Programming (XP), 16, 17, 29–30

FAR tool, 223

FCL (Framework Class Library), .NET, 132–133, 137
FinalBuilder tool, 241, 241–243
firewalls, 168
Fishman, Stephen, 264
Flywheel tool, 93
FogBUGZ tool, 171–173, 172
Fowler, Martin, 16, 20, 89, 91
FreeMind software, 5
full-domain language, 146
functional testing, 163
FxCop add-in, 110, 137–140, 139–140

Gamma, Erich, 24, 72
Gang of Four patterns, 24–25
Get command, 34
Google searches for reusable code, 137
GotDotNet online workspace, 206–207, 208
GotDotNet User Samples website, 137
GPLs (General Public Licenses), GNU, 261–262
Griffith, Matt, 253
Groove Workspace tool, 211, 212

HarnessIt tool, 73
Helm, Richard, 24
help. See coding documentation;
problems
Herrington, Jack, 144
Hippo.NET tool, 243

IBM Rational XDE software, 22,
114–115, 114–116
IDE alternatives, 102–103,
See also Visual Studio .NET IDE
IDE integration, 44, 44–45
Ilasm utility, 120–124, 123, 267–272,
269–270
inch-pebbles, 12
indexes for help files, 217–218, 221
infectious licenses, 261
inline-code expander, 146
Inno Setup tool, 277
Inside Microsoft Visual Studio .NET
(Johnson, Skibo, and Young), 106
installers. See delivering
instant messages, 204
integration testing, 70
intellectual property. See protecting
IntelliJ IDEA, 103
intent comments, 65
Iron Speed Designer tool, 149
iterative methodologies, 16

Java Programming 10-Minute Solutions (Watson), 72
Johnson, Brian, 106
Johnson, Ralph, 24
JUnit testing tool, 72

Kinook Software’s Visual Build Professional, 248–250, 249
Korzybski, Alfred, 23
KVM (keyboard-video-mouse)
switches, 168
Label command, 36
labels in source code control, 47
Learning UML (Alhir), 23
Li, Kanglin, 71
licenses, See also protecting infectious (viral) licenses, 261
open-source licenses
BSD license, 262
Creative Commons project and, 261
GNU General Public License, 261–262
Mozilla Public License, 263
overview, 260–261
overview, 259
proprietary licenses, 264–265
versus public domain, 260
shared-source licenses, 263–264
tools for building, 283
Lister, Timothy, 200
literate programming tools, 66
LOC (lines of code), 108–109
logging application activity, 176–195, See also coding defensively after shipping, 177
checklist, 194–195
using Debug class, 179
during development, 176, 230–231
diagnostic tools and overview, 193
Product Support Reporting Tools, 193–194
System Information utility, 193, 193
using Enterprise Instrumentation Framework architecture of, 186, 186
configuring, 188–189
defined, 185
event sinks, 186, 186, 187–188
event sources, 186–187, 186
using EventLog class, 182–185, 184
using log4net, 190–192
using Logging application block, 134, 189
overview, 176, 177–178
thinking before, 194
using Trace class methods/properties of, 178–179
overview, 178
steps in, 180–182
trace listeners, 179
trace switches, 179–180
Logidex .NET Library, 135–136, 136
m3rlin code generator, 149
McCarthy, Jim, 202
McConnell, Steve, 15, 156, 202
macros, See also Visual Studio .NET IDE
binding to keystrokes, 106
defined, 101
editing, 104–105, 105
recording, 104, 104
saving, 106
testing, 104
Manager’s Handbook for Software Development (NASA), 16
manuals, writing, 229–230
Merant Build tool, 240
Merge command, 36–37, 37
Microsoft, See also websites application blocks, 26, 134–135
build tools, 239–240
code libraries, 152–134
HTTP Help, 222, 223
Intermediate Language (MSIL), 120–124, 125, 268
Product Support Reporting Tools, 193–194
Project, 15
Solutions Framework (MSF), 16, 17
Windows Installer, 278–279, 280
Windows SharePoint Services, 210–211
mind maps, 4–6, 6
MindManager software, 5
mixed-code generator, 146
mock objects, unit testing with, 80–84
Morton, Rogers, 156
Move command, 38
MPL (Mozilla Public License), 263
MSF (Microsoft Solutions Framework), 16, 17
MSIL (Microsoft Intermediate Language), 120–124, 125, 268
The Mythical Man-Month (Brooks), 11
N
NAnt tool, 239, 243–248
NASA’s Software Engineering Laboratory (SEL), 15, 16
NDAs (nondisclosure agreements), 258
NDoc tool, 225, 228, 228–229
.NET. See Visual Studio networks, testing software on, 166–168
Neward, Ted, 133
“The New Methodology” article (Fowler), 16
Nielsen, Jakob, 204
NMock tool, 84
no-touch deployment, 287–288
nogoop .NET Component Inspector tool, 129–130, 130–131
noise comments, 61–62
Notepad as IDE, 102
NSIS (Null Scriptable Install System), 278
JUnit testing tool, See also unit testing debugging tests, 78–80, 79
defined, 72
first steps, 73–75, 75
testing download engine, 76–78, 78
testing exceptions, 80
testing with mock objects, 80–84
obfuscation, See also protecting approaches to, 266–267
defined, 128–129, 265–266
using Dotfuscator, 128, 267–272, 269–270
other .NET obfuscators, 128, 267
versus reflection, 128
Olero ORM.NET code generator, 149
online workspaces, 204–208, 206, 208
See also coding with open-source licenses, See also protecting BSD license, 262
OpenMake tool—source code

Creative Commons project and, 261
GNU General Public License, 261–262
Mozilla Public License, 263
overview, 260–261
OpenMake tool, 240
organizing projects. See code project organization

P

“Painless Software Schedules” article (Spolsky), 11
partial-class generator, 146
Pascal casing, 137
passive code generators, 144–145
A Pattern Language: Towns, Buildings, Construction (Alexander), 24
Patterns of Enterprise Application Architecture (Fowler), 20
patterns. See design patterns
Peopleware (DeMarco and Lister), 200
Perforce software, 38, 40
performance testing, 164
Peter, Dr. Laurence J., 201
Peter Principle, 201–202
PIAs (Primary Interop Assemblies), Office, 133
Pin command, 38
placeholder comments, 63, 63
planning projects. See code project planning
POCMock tool, 84
Polyvision’s Whiteboard Photo, 4
Positive-g’s Task Tracker, 12, 13
postmortem documentation, 231–232
PowerToys, customizing VS .NET with, 100–101
PrimalCode IDE, 102
problems and solutions
automating intractable tools, 252
broken builds, 252–253
code/comment synchronization, 66
developer idiosyncracies, 252
managing version numbers, 253
overview, 251
in refactoring, 92
SQL injection, 145
Product Support Reporting Tools, 193–194
profiles, customizing IDE with, 96–98, 97
Project Line Counter add-in, 107, 107–108
proprietary licenses, 264–265
protecting intellectual property,
256–273
checklist, 272–273
by not running on user machines, 272
with obfuscation
approaches to, 266–267
defined, 128–129, 265–266
using Dotfuscator, 128,
267–272, 269–270
other .NET obfuscators, 128, 267
overview, 256
with software contracts
overview, 256
staking claims, 256–257
what to include in, 257–259
with software licenses
BSD licenses, 262
General Public Licenses, 261–262
infectious (viral) licenses, 261
license-building tools, 283
Mozilla Public Licenses, 263
open-source licenses, 260–263
overview, 259
proprietary licenses, 264–265
versus public domain, 260
shared-source licenses, 263–264
warning, 256
public domain, 260

Q

QA (quality assurance). See testing code
QuickJump.NET add-in, 109, 109

R

Rapid Development (McConnell), 15, 202
Rational Unified Process (RUP), 16
Rational XDE software, 22, 114–115,
114–116
Recommended Approach to Software Development (NASA), 16
Refactoring: Improving the Design of Existing Code (Fowler), 89
refactoring. See also unit testing
defined, 89
easy, 90–91
problems with, 92
tools, 91–93
reflection, 128
Reflector utility, 124, 124–128, 127
regression testing, 164
release candidate builds, 29
Rename command, 38
repository, code control and, 43
rights. See protecting
risk management. See bug tracking
RoboHelp tool, 223, 224
Roeder, Lutz, 124
Roff, Jason T., 23
Rollback command, 38
RUP (Rational Unified Process), 16

S

Safari Software’s MasterList-XL, 12
Salamander tool, 267
scheduling software tools, 11–13, 13
searches for reusable code, 137
setup (installer) programs. See delivering
Share command, 36
shared-source licenses, 263–264
SharePoint Services, 210–211
SharpDevelop IDE, 102
Skibo, Craig, 106
SmartDraw software, 5
smoke testing, 164
Snipet Compiler tool, 103, 130–132,
131
Software Engineering Laboratory (SEL), 15, 16
Software Requirements (Wieger), 7, 9
Solutions Design LLBLGen Pro tool, 149
source code, 120–140, See also code;
coding
camel casing, 137
checklist, 140
evaluating design guidelines with
FxCop, 137–140, 139–140
source code bugs—unit testing

Experimenting with new code
using nogoop .NET
Component Inspector, 129–130, 130–131
overview, 132
using Snippet Compiler, 130–132, 131
Pascal casing, 137
Reusable code sources
balancing costs/benefits, 135
The Code Project, 137
Google searches, 137
GotDotNet User Samples, 137
legalities and, 120
Logidex .NET Library, 135–136, 136
Microsoft application blocks, 134–135
Microsoft .NET libraries, 132–134
Spelunking utilities
Ildasm, 120–124, 123
overview, 120
Reflector, 124–128, 124, 127
source code bugs. See bug; debug
source code control, 34–50
benefits, 35, 35
best practices
branching, 47–48
etiquette, 46–47
labels, 47
overview, 45, 48
storage, 45–46
checklist, 50
command levels
1: basic, 34–35, 35
2: code tree management, 36–37, 36–37
3: advanced, 37–38
example, 48–49, 49
overview, 34, 38
protection code, 35
software tool factors
advanced commands, 45
concurrent development style, 42–43
cross-platform support, 45
IDE integration, 44–45, 44
Internet friendliness, 43
price, 41–42
repository, 43
software tools for
AccuRev, 38, 39
BitKeeper, 38, 39
ClearCase, 38, 39
CVS, 38, 39, 39–40
Perforce, 38, 40
StarTeam, 39, 40
Subversion, 39, 40
Vault, 39, 39, 40, 48–49, 49
VSS, 39, 39, 40–41, 42, 42
viewing code history, 35, 35
in Visual Studio .NET, 44–45, 44
source code IDEs, 102–103, 103–103, See also
Visual Studio .NET IDE
source code rights. See protecting
Source Control Provider Selector
utility, 44
SourceForge tool, 205, 206
SourceGear Vault software, 35, 39, 39, 40, 48–49, 49
Spices tool, 267
Spolsky, Joel, 11, 12
SQL injection problem, 145
SQLXML code library, 133
StarTeam software, 39, 40
storage in code control system, 45–46
stress testing, 71, 164
“Structuring Solutions and Projects”
article, 45
Subversion software, 39, 40
summary comments, 64–65
System Information utility, 193, 193
system testing, 70
Task Tracker software, 12, 13
TDD (test-driven development)
benefits, 89
defined, 85
example, 85–88, 87
tools. See coding with
.TEST tool, 73
Test-Driven Development: By
Example (Beck), 85
testing code, See also bug tracking; unit
testing
acceptance testing, 71
beta testing, 29, 71
black-box testing, 164
compatibility testing, 164
on computer networks, 166–168
conformance testing, 163–164
for developers, 71–72
end-user help files, 221
exceptions, 80
functional testing, 163
installer programs, 282
integration testing, 70
macros, 104
overview, 70, 163
performance testing, 164
regression testing, 164
smoke testing, 164
stress testing, 71, 164
system testing, 70
unit testing, 70, 163
warning, 72
when working solo, 165–166
white-box testing, 164
tier generator, 146
time and billing tools, 13–14, 14
tooltips, 225
Total .NET Analyzer add-in, 110–111
Total .NET XRef add-in, 110, 110
Trace class, See also logging
logging with, 180–182
methods/properties, 178–179
overview, 178
trace listeners, 179
trace switches, 179–180
tracking project progress, See also bug
tracking
inch-pebbles and, 12
overview, 11
with scheduling tools, 11–13, 13
team progress, 200–201
with time and billing tools, 13–14, 14
Troubleshooting. See problems
UML: A Beginner’s Guide (Roff), 23
UML (Unified Modeling Language), 22–23, 114–115, 115
Undo Check Out command, 35
uninstallers, 282
unit testing, 70–93, 70
checklist, 91
defined, 70, 163
using JUnit tool, 72
Updater application block—websites

.NET tools for
  csUnit, 72
  HarnessIt, 72
  NUnit, 72
  .TEST, 73
  X-Unity, 73
NUnit use example
debugging tests, 78–80, 79
  defined, 72
  first steps, 73–75, 75
  testing download engine, 76–78, 78
  testing exceptions, 80
  testing with mock objects, 80–84
overview, 93
using refactoring
  defined, 89
  example, 90–91
  problems with, 92
  tools, 91–93
using TDD
  benefits, 89
  defined, 85
  example, 85–88, 87
Updater application block, 134
UpdateVersion utility, 253
use case documentation, 9–10
user feedback links, 222
user help files. See coding documentation
User Interface application block, 134
user interface, coding help into, 225

W

waterfall methodologies, 15–16
Watson, Mark, 72
Web & Software Development: A Legal Guide (Fishman), 264
web parts, 210–211
Web Services Enhancements (WSE) library, 133
websites
  agile methodologies, 16–17
  articles
    “Building Secure Assemblies”, 188
    “Customizing VS .NET Start Page”, 98
    “Deploying .NET Framework-Based Applications”, 288
    “The New Methodology”, 16
    “Painless Software Schedules”, 11
    “Structuring Solutions and Projects”, 45
  Bruce Eckel web log, 22
  bug tracking tools, 169, 171
  build tools, 239–241, 243, 248, 251–253
  code generation tools, 143, 144, 149–150
  CodeReview, 213
  CodeWright, 212
  design patterns, 24
  developer documentation tool, 225
  FxCop, 138
  Groove Workspace, 211
  help authoring tools, 222–223
  InstallSite, 279
  iterative methodologies, 16
  Joel Spolsky article, 11
  literate programming tools, 66
  Logidex .NET Library, 135
white-box testing—Young, Marc 297

Martin Fowler article, 16
Messenger Plus, 204
Microsoft
application logging tools, 185, 190
code libraries, 133
code-building tools, 239–240
HTML Help, 222
shared-source licenses, 263, 264
SharePoint Services, 210
Windows Installer SDK, 279
Microsoft Patterns & Practices
application blocks, 26, 134, 189
articles, 188, 288
defined, 25
mind mapping software, 4–5
NDoc, 225
.NET
continuous integration tools, 251
design guidelines for developers, 137
Framework Class Library, 133
IDE alternatives to, 102–103
installation creator tools, 277
license-building tools, 283
mock object creation tools, 84
obfuscator utilities, 128
PowerToys, 100
refactoring tools, 92–93
unit testing tools, 72–73
.NET add-ins
CodeSMART, 113
CodeWright, 112
Project Line Counter, 107
QuickJump.NET, 109
Rational XDE, 114
from SharpToolbox, 116
Total .NET Analyzer, 110
Total .NET XRef, 110
VB Commenter, 226
XMLSpy, 112
nogoop .NET Component Inspector, 129
Nolo Press, 199, 264
Norton Ghost, 167
online workspace tools, 205, 206
open-source licenses, 260–263
ProFont and Sheldon fonts, 99
project scheduling software, 12
Snippet Compiler, 130
software testing, 164
source code control software, 38–39
Source Control Provider Selector, 44
spam filters, 203
SQL injection white papers, 145
time and billing software, 13
Tony Buzan, 4
Trillian service, 204
UML tools, 22
UpdateVersion utility, 253
user-interface automation tools, 252
virtual machine software, 167
waterfall methodologies, 16
Whiteboard Photo, 4
wikis, 209, 210
white-box testing, 164
Whiteboard Photo, 4
Wiegert, Karl, 7
wikis, 209, 210
WinBatch tool, 252
Windows Installer service, 278–279, 280
Windows SharePoint Services, 210–211
Wisestar creation tools, 278–279
Working Time Tracker software, 13
workspaces online, 204–208, 206, 208,
see also coding with
Workstate Codify tool, 150
WSE (Web Services Enhancements) library, 133
Wu, Mengqi, 71

X

X-Unity testing tool, 73
XCOPY deployment, 286–287
XML comment tags, 225–228, 227
XMLSpy add-in, 112–113
XP (Extreme Programming), 16, 17, 29–30

Y

Yes! Software CodeCharge Studio tool, 150
Young, Marc, 106