Index

A
Abstract Factory pattern, 186
Abstractions, 1–2, 55
mini-frameworks and, 129–30
Adapter pattern, 129–30
Advisory councils, 50
Alles in Ordnung pattern, 11, 19, 20–29, 34, 46, 48n8, 50, 52, 81, 106, 137, 145, 165, 184, 189, 199
applicability of, 28
context of, 20–21
example of, 22–24
intent of, 20
problem addressed by, 24–27
approaches to, 24–27
related patterns, 29
use/non-use of, 27–29
Ambler, Scott, iii
Analysis, 87–107
Eating the Elephant (One Byte at a Time...) pattern, 40, 88–96, 102, 193
Something Is Better Than Nothing pattern, 87, 96, 97–102, 107, 173
Where’s Mr. Spock When You Need Him? pattern, 87, 102–7
Analysis models, 54, 58, 202
Analysis Patterns (Fowler), 44
Anarchy (lack of process), 24–25
Antisolocial behavior, 157–59
Application developer, 96
documentation for, 147
Applications frameworks as, 8
sample, 55–56
Appropriateness, determining, 63–64
Architects, 203
guideline refinement by, 169
Artifacts, 3–4
consistency of, 23–24
cost and delivery of, 56
development, 9
documentation of, 56–57, 202–3
of IBM SanFrancisco project, 5, 58, 201–3
internal, 6
missing, 22–24
as part of product, 26
process, 5
user documentation, 202–3
Attributes, inclusion decision, 84

B
Beta testers, 51
“Big-brained” individuals, 35, 162
“Big-M Methodology.” 25
Build verification test, 213
Business application frameworks, 5
Business requirements
decomposing, 197
defining, 196

C
Candidate pattern, 122–23
Capabilities, deciding on, 81–82. See also
Stupid Test pattern
Capability Maturity Model, 26
Category level, customization at, 187
CBP (Core Business Process), 94–96
Chain of Responsibility Driven Strategy pattern, 119–24
Change, isolating, 88
Change Only What You Need to Change pattern. See Color Inside the Lines pattern
Checkpoints, 31–32, 104–5, 106
combined review, 6, 26, 32, 208–10
in framework customization, 187
requirements handoff, 32
validation, 105
Cheesman, John, 194–95
Classes
"chunking" of, 88
multiple dependencies among, 89–90
Class libraries, 2, 8, 114
Class replacement, 136, 186
Class-Responsibility-Collaboration (CRC), 98
Clothing management domain, 13–18
cleaning clothing, 14, 16–17
overview of, 14–15
purchasing clothing, 15, 17–18
repairing clothing, 15, 17
selecting clothing, 14, 15–16
Coarse-grained components, 38, 194, 195–98
Cockburn, Alistair, 25, 162
Code fix process, 201
Code formatting rules, 170
Code generation
pattern iteration and, 124, 125
tools, 44
Code review, 211–12
Code test, 210–11
Cognitive dissonance, 21
Color Inside the Lines pattern, 175, 185–89
applicability of, 189
context of, 185
example of, 186
intent of, 185
problem addressed by, 186–88
approaches to, 187–88
solution to, 188
related patterns, 189
use/non-use of, 188, 189
Combined review, 6, 26, 32, 208–10
Communication, 160. See also Great Communicator pattern
between domain and technical experts,
103–4
with domain experts, 161–65
in IBM SanFrancisco project, 165
planned, 191–92
Complexity
tradeoff between flexibility and, i, iii
unnecessary, 125–26
Component(s), 193–98
characteristics of, 193–95
coarse-grained, 38, 194, 195–98
dependencies among, 91–93, 195
fine-grained, 195, 196, 197
objects vs., 193–94
warehousing, 194–95
Component-level decomposition, 93–94
Component regression test, 215–16
Component test, 213–15
Component test cases, 202
Component test team member, 203
Connecting Domain and Technical Experts pattern. See Innocent Questions pattern
Consciously competent individual, 163, 164
Consciously incompetent individual, 164
Consistency
of artifacts, 23–24
identifying opportunities for, 42–43
importance of, 192
of internal artifacts, 6
"in the large," 43–44
"in the small," 43
Consistency Czar pattern, 43, 44, 46, 52, 56,
59, 68, 74, 127, 153, 159, 165–73
applicability of, 172
context of, 165–66
example of, 166–67
intent of, 165
problem addressed by, 167–72
approaches to, 167–71
solution to, 171–72
related patterns, 173
use/non-use of, 172
Consistency Is King pattern, 11, 19, 29, 40–46,
74, 166, 173
applicability of, 46
context of, 40–41
example of, 41–42
intent of, 40
problem addressed by, 42–45
approaches to, 42–45
solution to, 45
related patterns, 46
use/non-use of, 45, 46
Consulting, reducing multiple documents through, 149, 150
Contextual documents, 57
Contingency management, 110. See also Pass the Buck pattern
Contravariance
breaking, 134–35, 137, 188
customizing frameworks and, 187–88
Core guidelines, 167–68
Core values and principles, 155–56
Cost, artifact delivery and, 56
CRC (Class-Responsibility-Collaboration), 98
Creativity within guidelines, 170–71, 172
Creeping elegance, 64
Cross-component dependencies, 91–93
Cross-domain architect, 203
Cross-domain (utility) layer, 36–37
Cross-education sessions, 162
Culture, team dynamics and, 157, 158
Currency exchange rates, calculating, 66–67
Customization, 186. See also Color Inside the Lines pattern
areas of, 187
contravariance and, 187–88

D
Daniels, John, 195
"Data-up" approach, 103
Decision making
decision deferral vs., 111–14
documentation of, 113
in IBM SanFrancisco project, 115
Decomposing the Problem pattern.
See Eating the Elephant (One Byte at a Time...) pattern
Decomposition
at analysis level, 88
of business requirements, 197
component-level, 93–94
Deliverables, relationships between, 5–6
Dependencies
cross-component, 91–93, 195
intercomponent, 195
upward, 96
Design, 109–37
in IBM SanFrancisco project, 207–8
It's Still OO to Me pattern, 109, 132–37, 189
Missed It by That Much pattern, 67, 109,
116–27, 132, 173
Pass the Buck pattern, 52, 96, 109, 110–15
pattern development and, 116
That's the Way the Cookie Crumbles
pattern, 109, 127–32, 137
Design models, 54–55, 58, 202
Design Patterns for Object-Oriented Software Development (Pree), 126
Design Patterns (Gamma), 117, 125, 186
Developer, 203
Developing and Applying Patterns. See Missed It by That Much pattern
Development artifacts, 9
Development group lead, 203
Development process, 4–7, 19–59. See also IBM SanFrancisco Frameworks
Development process
Alles in Ordnung pattern, 11, 19, 20–29, 34, 46, 48n8, 50, 52, 81, 106, 137, 145, 165, 184, 189, 199
Consistency Is King pattern, 11, 19, 29,
40–46, 74, 166, 173
Divide and Conquer pattern, 11, 19, 35–40, 59, 96, 165
Exposing It All pattern, 11, 19, 26, 29,
52–59, 181
Innocent Questions pattern, 11, 19, 29–34, 48, 52, 67, 81, 101, 107, 115, 165
Iterate, Iterate, Iterate pattern, 11, 46–52, 91, 96, 101, 145
special events in, 26
Different Framework Audiences Have Different Needs pattern. See Give 'Em What They Want pattern
Disruptive behavior, 157–59
Divide and Conquer pattern, 11, 19, 35–40, 59, 88, 96, 165
applicability of, 39
context of, 35
example of, 35
intent of, 35
problem addressed by, 36–39
approaches to, 36–38
solution to, 38–39
related pattern, 40
use/non-use of, 39–40
Documentation, 3, 54–55, 139–51, 191. See also Exposing It All pattern
of artifacts, 56–57, 202–3
audiences of, 146–47
of decision making, 113
end-to-end, 9
entry points into, 177–78
Give 'Em What They Want pattern, 59, 139, 145–51, 181
for IBM SanFrancisco project, 144, 145, 151, 179
interface, 177
iteration and, 141
layered, 149
learning from, 177–80
meta-documentation, 179
of patterns, 54–55, 126
Souvenirs pattern, 59, 127, 139, 140–45, 173
synchronization of, 142–43
Document guidelines, 168, 171
Documents, contextual, 57
Document What You Know When You Know It pattern. See Something Is Better Than Nothing pattern
Domain(s), 4
clothing management, 13–18
cleaning clothing, 14, 16–17
overview of, 14–15
purchasing clothing, 15, 17–18
repairing clothing, 15, 17
selecting clothing, 14, 15–16
neutral, 13
Domain concepts, names of, 55
Domain expert(s)/expertise, 203
analysis by, 97–100
arguments among, 63, 65
communication between technical experts and, 103–4. See also Where’s Mr. Spock When You Need Him? pattern
information transfer, 160–65
knowledge transfer, 102, 106
communication with, 161–65
culture of, 157
in design discussions, 117
documentation for, 147
extreme requirements and, 69–70
implementation bias of, 75–77
initial framework analysis by, 105
leveraging ignorance on, 62–63, 65
personality type of, 78, 79
questioning, 65–66, 77–78, 79
refinement iterations and, 49–50, 51
split between technical expertise and, 5–6, 34
Domain lead, 203
Domain-specific layer, 36, 37
Domain teams, 99
social aspect of, 49
Domain-Technical Cross-Team Communication pattern. See Where’s Mr. Spock When You Need Him? pattern
Eating the Elephant (One Byte at a Time...) pattern, 40, 88–96, 102, 193
applicability of, 94
context of, 88
example of, 89
intent of, 88
problem addressed by, 90–93
approaches to, 90–93
solution to, 93
related patterns, 96
use/non-use of, 93–96
Education
in consistency rules, 44
cross-education sessions, 162
as substitute for documentation, 149, 150
Elegance, creeping, 64
End-to-end documentation, 9
Enforcement of consistency rules, 44–45
Entities, domain responsibilities of, 90
EntityBeans, 195
Evaluating requirements, 75. See also What, Not How pattern
Examples, frameworks as, 9
Exposing It All pattern, 11, 19, 26, 29, 52–59, 181
applicability of, 58
context of, 52–53
example of, 53
intent of, 52
problem addressed by, 54–57
approaches to, 54–57
solution to, 57
related patterns, 59
use/non-use of, 57–59
Extending frameworks, 185. See also Color
Inside the Lines pattern
Extension guide, 59, 202
Extension points (hotspots), 2, 47, 62, 116–17.
See also It Depends pattern
Extreme Programming process, 26
Extreme requirements, 67. See also Tor’s Second Cousin pattern
domain expert and, 69–70
in IBM SanFrancisco project, 71–74

F
Features (functional units), 200, 201
integration of, 212–13
Feature test, 210–11
Feature test cases, 202
Fine-grained components, 195, 196, 197
Flexibility, 55
extreme, 68
points of. See Extension points (hotspots);
It Depends pattern
tradeoff between complexity and, i, iii
Ford, Henry, 62
Follow a methodical development process.
See Alles in Ordnung
Foundation layer, 36
Framelets (mini-frameworks), 127. See also That’s the Way the Cookie Crumbles pattern
abstractions and, 129–30
in IBM SanFrancisco project, 132
Framework developers, 147
Frameworks, 1–11
defined, 1–3
developing, 4–7
process patterns, 9–11
target area of, 2
using, 7–9
Frameworks Aren’t Exempt from Good and Bad Object-Oriented Practices. See It’s Still OO to Me pattern
A Frameworks’ Customer Is Its Partner pattern. See Exposing It All pattern
Functional overview, 59, 202
Functional units (features), 200, 201
integration of, 212–13

G
Gap analysis, 54
Generally accepted accounting principles (GAAPs), 149–50
Give ‘Em What They Want pattern, 59, 139, 145–51, 181
applicability of, 151
context of, 146–47
example of, 147
intent of, 145
problem addressed by, 148–50
approaches to, 148–50
solution to, 150
use/non-use of, 150–51
Glossary, 59, 202
Great Communicator pattern, 107, 127, 153, 160–65, 181
applicability of, 164
context of, 160–61
example of, 161
intent of, 160
problem addressed by, 161–63
approaches to, 161–62
solution to, 162–63
related patterns, 165
use/non-use of, 163–64, 165
Guidelines, 167–73
adapting, 169–70, 171
core, 167–68
creativity within, 170–71, 172
Guidelines, continued
document, 168, 171
extending, 169–70
in IBM SanFrancisco project, 46, 172
overspecifying, 170, 172
responsibility and accountability for, 169, 171
software component design process, 196–98

H
“Hammer and nail” syndrome, 125, 170
Heisenberg uncertainty principle, 65
Hotspots. See Extension points (hotspots)
How Extreme Is Too Extreme? pattern. See
Tor’s Second Cousin pattern
Human aspects of software development, 191. See also Social aspects
Humor, 69–71
team dynamics and, 156

I
IBM SanFrancisco Frameworks Development
process, 199–216
advisory councils in, 50
artifacts of, 5, 58, 201–3
attributes inclusion rules in, 85
build verification test in, 213
class replacement pattern in, 186
code and feature test in, 210–11
code review in, 211–12
combined review in, 208–10
communication in, 165
component-level decomposition in, 94–96
component regression test in, 215–16
component test in, 213–15
consistency guidelines in, 46, 172
decision making in, 115
design of, 207–8
documentation for, 144, 145, 151, 179
domain validation support in, 66
extreme requirements in, 71–74
implementations creeping into
requirements statements in, 80
initial analysis model in, 104
integration in, 212–13
layers used in, iv-v, 37
mapping in, 184

mini-frameworks in, 132
participants in, 203
patterns design in, 119
process flow, 200–201
requirements development in, 204
scenario development in, 204–5
upward dependencies in, 96
user guides for, 148–49

Identifying Customization pattern. See It Depends pattern
Ignorance, leveraging, 63, 65
Implementations Masquerading as Requirements pattern. See What, Not How pattern
Importance of Team Dynamics pattern. See There Is No “I” in Team pattern
Include Obvious Domain Capabilities While Keeping the Framework Focused pattern. See Stupid Test pattern
Incompleteness, 192
Information architect, 203
Information development team leader, 203
Information transfer between technical and domain members, 102, 106, 160–65. See also Communication
Initial analysis model(s), 97–98, 103–4. See also Something Is Better Than Nothing pattern
for educating technical team, 105
Initial model, refinement of, 104
Innocent Questions pattern, 11, 19, 29–34, 48, 52, 67, 81, 101, 107, 115, 165
applicability of, 34
cost of, 29–30
element of, 30
problem addressed by, 30–33
approaches to, 31–33
solution to, 33
related patterns, 34
use/non-use of, 33–34
Integration of features, 212–13
Intellectual property, 56
Interaction diagrams, 58, 202
Interaction points. See Checkpoints
Intercomponent dependencies, 195
Interface documentation, 177
Internal artifacts, 6
Isolating change, 88
It Depends pattern, 62–67, 81, 85
  applicability of, 66
  context of, 62–63
  example of, 63
  intent of, 62
  problem addressed by, 64–65
    approaches to, 64–65
    solution to, 65
  use/non-use of, 65–67
Iterate, Iterate, Iterate pattern, 11, 46–52, 91, 96, 101, 145
  applicability of, 51
  context of, 46–47
  example of, 47–48
  intent of, 46
  problem addressed by, 48–51
    approaches to, 48–51
    solution to, 51
  related patterns, 52
  use/non-use of, 51, 52
Iteration, 192
  documentation and, 141
  mapping and, 183
  in pattern development, 124, 125
It’s Still OO to Me pattern, 109, 132–37, 189
  applicability of, 137
  context of, 133
  example of, 134–35
  intent of, 132
  problem addressed by, 135–36
    approaches to, 135–36
    solution to, 136
  related patterns, 137
  use/non-use of, 136, 137
J
Java, v
Java2 Enterprise Edition (J2EE), 195
Javadoc, 54, 55, 58
Johnson, Ralph, 47
Just Learn It pattern, 175, 176–81, 185
  applicability of, 180
  context of, 176
  example of, 176–77
  intent of, 176
  problem addressed by, 177–80
    approaches to, 177–79
    solution to, 179–80
  related patterns, 181
  use/non-use of, 180
K
Keeping Just Enough to Write Documentation Later pattern. See Souvenirs pattern
Knowing When a Framework Shouldn’t Do Something pattern. See Pass the Buck pattern
Knowledge transfer between domain and technical members, 102, 106. See also
Where’s Mr. Spock When You Need Him? pattern
L
Latitude, encouraging, 155–56
Layers of framework, 36–38, 39
Leader(s)
  project, 154–55, 156, 158, 167
  team, 169, 203
Learning. See also Just Learn It pattern
  from documentation, 177–80
  four stages of, 163–64
  of frameworks, 176
  up-front, 180
Look-and-feel rules, 40–41
M
Maintain Consistency throughout the Framework. See Consistency Is King pattern
Maintenance, 7
Making Sure Consistency Happens pattern. See Consistency Czar pattern
Making the Framework Consumable pattern. See Divide and Conquer pattern
Map Early, Map Often pattern, 59, 175, 181–85
  applicability of, 184
  context of, 181–82
  example of, 182
  intent of, 181
  problem addressed by, 182–84
    approaches to, 182–83
    solution to, 183–84
  related patterns, 184–85
  use/non-use of, 184
Mapping, 8–9, 192
  to fine-grained business components, 197
IBM SanFrancisco project and, 184
iteration and, 183
Meta-documentation, 179
Meta-patterns, 126
Methodical Development Process pattern.
See Alles in Ordnung pattern
Mini-frameworks (framelets), 127. See also
That's the Way the Cookie Crumbles pattern
abstractions and, 129–30
in IBM SanFrancisco project, 132
Missed It by That Much pattern, 67, 109,
116–27, 132, 173
applicability of, 126
context of, 116–17
example of, 117–18
intent of, 116
problem addressed by, 118–25
approaches to, 118–24
solution to, 124–25
related patterns, 127
use/non-use of, 125–26, 127

N
Neutral domain, 13
"Not invented here" syndrome, 21n2
Noun identification in use cases, 98

O
Object-oriented framework, 132. See also It's
Still OO to Me pattern
Object-oriented principles, v, 192
Objects, components vs., 193–94
"Object spaghetti," 88
OO lead, 203

P
Pass the Buck pattern, 52, 96, 109, 110–15
applicability of, 115
context of, 110–11
example of, 111
intent of, 110
problem addressed by, 111–14
approaches to, 111–13
solution to, 113–14
related patterns, 115
use/non-use of, 114–15

Patterns. See also Alles in Ordnung pattern;
Color Inside the Lines pattern;
Consistency Czar pattern; Consistency Is King pattern;
Divide and Conquer pattern; Eating the Elephant (One Byte
at a Time...) pattern; Exposing It All pattern; Give 'Em What They Want
pattern; Great Communicator pattern;
Innocent Questions pattern; It Depends pattern; Iterate, Iterate, Iterate pattern;
It's Still OO to Me pattern; Just Learn It pattern;
Map Early, Map Often pattern;
Missed It by That Much pattern; Pass the Buck pattern;
Something Is Better Than Nothing pattern; Souvenirs
pattern; Stupid Test pattern; That's the Way the Cookie Crumbles pattern;
There Is No 'I' in Team pattern; Tor's Second Cousin pattern; What, Not
How pattern; Where's Mr. Spock When You Need Him? pattern
development of, 116
documentation of, 54–55, 126
Patterns-based approach, 9–11
Patterns Can Be Mini-Frameworks pattern.
See That's the Way the Cookie Crumbles pattern
Performance team member, 203
Persistence layers, 84
Persistent object planning guide, 59, 202
Pick lists, 72–74
Prima donnas, 158–59
Principles, core, 155–56
Problem domain, 88. See also Eating the
Elephant (One Byte at a Time...) pattern
Process artifacts, 5
Process pattern, defined, iii
"Programmer object" syndrome, 31n5
Programming languages, 2
Programming model, 54
Project leaders, 154–55, 156, 158, 167
Project management, 27
Prototype framework usage team, 178–79, 180

R
Rational Unified Process, 5. See also Unified
Software Development Process
Refinement iterations. See Iterate, Iterate, Iterate pattern
Replenishment processing, 119–24
Requirements, 54, 58, 61–85, 201
   business, 196, 197
evaluating, 75. See also What, Not How pattern
extreme, 67. See also Tor’s Second Cousin pattern
domain expert and, 69–70
   in IBM SanFrancisco project, 71–74
It Depends pattern, 62–67, 81, 85
refinement iterations and, 49
Stupid Test pattern, 81–85, 102
Tor’s Second Cousin pattern, ii-iii, 48, 52, 66, 67–74, 85, 115, 127, 159
What, Not How pattern, 49, 52, 75–81, 116, 127, 145, 173
Requirements development in IBM SanFrancisco project, 204
Requirements handoff checkpoint, 32
Reusability, 7
Reusable software, 41
Review, combined, 6, 26, 32, 208–10
Rewarding team successes, 156
Roles of team members, 31
Rules, overspecifying, 170, 172

S
Sample applications, 55–56
SanFrancisco project. See IBM SanFrancisco Frameworks Development process
Scenarios, 58, 202
   in IBM SanFrancisco project, 205–7
Serling, Rod, 165
Service and support, 56
SessionBeans, 195
   “7±2” rule, 93
Social aspects, 153–73
   Consistency Czar pattern, 43, 44, 46, 52, 56, 59, 68, 74, 127, 153, 159, 165–73
   of domain teams, 49
   Great Communicator pattern, 107, 127, 153, 160–65, 181
   of software development, 69
   There is No “I” in Team pattern, 49, 52, 107, 153, 154–59
Software developers, culture of, 157
Software development
   human aspects of, 191
   social dimensions of, 69. See also Social aspects
Something Is Better Than Nothing pattern, 87, 96, 97–102, 107, 173
   applicability of, 101
   context of, 97–98
   example of, 98
   intent of, 97
   problem addressed by, 98–100
      approaches to, 98–100
      solution to, 100
   related patterns, 101–2
   use/non-use of, 100–101
Source code, 56, 58
Souvenirs pattern, 59, 127, 139, 140–45, 173
   applicability of, 144
   context of, 140–41
   example of, 141
   intent of, 140
   problem addressed by, 141–43
      approaches to, 141–43
      solution to, 143
   related patterns, 145
   use/non-use of, 143–44, 145
Specifying consistency, 43–44
Standards for consistency, 43
Stock putaway rules, 85
Strategy pattern, 22n3, 117–18, 119–20, 129
Stupid Test pattern, 81–85, 102
   applicability of, 84
   context of, 81–82
   example of, 82
   intent of, 81
   problem addressed by, 82–84
   related patterns, 85
   use/non-use of, 84, 85
Szyperski, Clemens, 194

T
Team(s)
   managing, 31–33
   pairing of members of, 105
   prototype framework usage by, 178–79, 180
   roles of members, 31
Team dynamics. See also Social aspects; There Is No "I" in Team pattern
antisocial or disruptive behavior, 157–59
culture and, 157, 158
humor and, 156
individual idiosyncrasies and, 157, 158
Team leaders, 169, 203
Technical expert(s)/expertise
communication between domain experts and, 103–4. See also Where’s Mr. Spock When You Need Him? pattern
information transfer, 160–65
knowledge transfer, 102, 106
initial analysis model for, 105
refinement iterations and, 50, 51
split between domain expertise and, 5–6, 34
Template Method pattern, 118
Test(s)
build verification, 213
code, 210–11
component, 213–15
compoment regression, 215–16
feature, 210–11
Test cases, 56, 202
Testing teams, independent, 50
That’s the Way the Cookie Crumbles pattern, 109, 127–32, 137
applicability of, 132
context of, 127–28
example of, 128
intent of, 127
problem addressed by, 128–31
approaches to, 128–31
solution to, 131
related pattern, 132
use/non-use of, 131, 132
There Is No "I" in Team pattern, 49, 52, 107, 153, 154–59
applicability of, 159
context of, 154–55
example of, 155
intent of, 154
problem addressed by, 155–59
approaches to, 155–58
solution to, 158–59
related patterns, 159
use/non-use of, 159
Three Iterations to Validate pattern. See Iterate, Iterate, Iterate pattern
Tooling, 3
Tor’s Second Cousin pattern, ii-iii, 48, 52, 66, 67–74, 85, 115, 127, 159
applicability of, 71
context of, 67
example of, 67–68
intent of, 67
problem addressed by, 68–71
approaches to, 68–70
solution to, 70–71
related patterns, 74
use/non-use of, 71–74
Translating between Domain and Technical Terminology and Concepts pattern. See Great Communicator pattern
Twilight Zone, The (TV show), 165n3

U
Unconsciously competent individual, 164
Unconsciously incompetent individual, 163–64
Unified Software Development Process, 4, 5, 25–26
Upward dependencies, 96
Usage samples, 3
Use case(s), 54, 58, 202, 204
handoff of, 6
noun identification in, 98
Use Frameworks by Mapping pattern. See Map Early, Map Often pattern
User documentation artifacts, 202–3
User guides, 59, 202
Using frameworks, 175–89
Color inside the Lines pattern, 175, 185–89
Just Learn It pattern, 175, 176–81, 185
Map Early, Map Often pattern, 59, 175, 181–85
Using Frameworks Requires Up-Front Education pattern. See Just Learn It pattern
Utility (cross-domain) layer, 36–37

V
Validation checkpoint, 105
Values, core, 155–56

W
Warehouse Management Core Business Process (CBP), 94–96
Warehousing component, 194–95
WebSphere Advanced Server, 37
What, Not How pattern, 49, 52, 75–81, 116, 127, 145, 173
applicability of, 79–80
context of, 75
example of, 76
intent of, 75
problem addressed by, 76–79
approaches to, 76–78
solution to, 79
related patterns, 81
use/non-use of, 79, 80
"What if" questions, 77, 78, 79
Where’s Mr. Spock When You Need Him?
pattern, 87, 102–7
applicability of, 106
context of, 102
eexample of, 102–3
intent of, 102
problem addressed by, 103–6
approaches to, 103–5
solution to, 105
related patterns, 106–7
use/non-use of, 106
Writing Effective Use Cases (Cockburn), 44