Denne_IN_p187-190 9/10/03 4:42 PM Page 187

Index

A

Agile development, 28
  architecture defined, 44
  incremental funding methodology and benefits to, 122–123
  misconceptions of, 122
  simple versus look-ahead solution, 134–139
Agile Software Development: Principles, Patterns, and Practices (Martin), 132
Alexander, Christopher, 45, 46
Application service providers (ASPs), 154–155
Architectural elements (AEs), 50
  case example, 161–163
  constructing, 116–118
  sequencing, 74–75, 113–114
Architecture
  codependency, 48–49
  coherency, 54–55
  decomposition, 50–52
  defined, 43–44
  dependency sequencing, 53–54, 163
  eXtreme programming, 44, 132–133
  lifecycle, 112–113
  multiple inheritance, 55–57
  problem with, 45–46
  selecting, 110–111
  spiral, 57, 104
  tradeoff analysis method (ATAM), 111
  usefulness, 46–47
  value-driven approach, 47
  versus rules, 44–45

B

Bass, L., 43, 45
Beck, Kent, 122
Behavior, categories of, 79–81
Boehm, Barry, 3, 36, 47, 57, 115
Booch, G., 45
Bottom-up approach, 31, 36
Breakeven time, 16, 18, 19, 147–148, 176–180
Business case, 15, 106
Busse, K., 46

C

Capablanca Y Granperra, J. R., 56
Cash flow projections, 15–16, 176–180
Chaos Report, 106, 112
Clements, P., 43, 45
Coad, Peter, 3
Codependency, architecture, 48–49
Coherency, architecture, 54–55
Construction, 116–118
Constructive cost model (COCOMO), 36, 37, 57
Cost-benefit analysis
  case example, 164–166
  of delivery sequence, 63–66
  of intangibles, 97–99
Costs
  of developing minimum marketable features, 36–37
  estimating, 111–112
  lost opportunity, 100–101
  versus value analysis, 62–63
Customer
  requirements, 10
  voice of, 10

D

Dataquest CIO, 14
Decision making, 10
  collaboration, need for, 145–146
  getting a project funded, 146–148
  project characteristics, manipulating, 149–150
Decomposition
  See also Function-class decomposition
  architecture, 50–52
  of minimum marketable features into use cases, 107–108

Denne_IN_p187-190 9/10/03 4:42 PM Page 188

Delivery
cost-benefit analysis of, 63–66
parallel development, 83–87
time-sensitive, 81–82
Dependency sequencing, architecture, 53–54
Design, 114–116
Development time, 37
Discounted cash flow (DCF), 17–18, 19
Dollar amounts, translating value to, 32–34

E

Elaboration, 110–113
eXtreme programming (XP), 3, 122
architecture defined, 44, 132–133
release, 123–124, 127–129
user stories, 124–129
Extreme Programming Explained: Embrace Change (Beck), 122
Extreme Programming Explored (Wake), 127

F

Feature
design, 116
operational capability, 117
release, 118
use of term, 29
Feature-driven development (FDD), 3, 29, 122, 140–142
Function-class decomposition, 30–31
Funding of a project, getting, 146–148

G

General Electric (GE), 9
Gilb, Tom, 57, 151
Glossary, developing a project, 109
Green Door Services, 3
Growth patterns, 82–83
Guadagno, L., 143

H

Hecksel, D., 46

I

IBM, 151
Inception, 105–110
Incremental funding methodology (IFM), 8, 27–28
See also Agile development; Rational unified process; Sequencing strategies
architectural elements, 50
benefits to management, 155
coherency, 54–55
decomposition, 50–52
heuristic, 170–175
impact of, 153–155
implementing, 152
multiple inheritance, 55–57
unexpected, dealing with the, 155–157
window, 151–152
Incremental funding methodology (IFM), case example
architectural elements, eliciting, 161–163
case flow and breakeven time, 176–180
computation phase, 166–173
dependencies, defining, 163
financial phase, 164–166
heuristic, 170–175
minimum marketable features, selecting, 160–161
minimum marketable feature strands, defining, 161
precedence graph, construction of, 163–164
return on investment analysis, 175–176
Information Week, 34
Initial operational capability (ICO), 117
Intangible(s)
benefits, 34–35, 90
cost-benefit analysis and, 97–99
defined, 89
hybrid, 97
lost opportunity costs, 100–101
managing, 90–91
net present value, 99
pairwise comparisons, 91, 92–97
Internal rate of return (IRR), 18–19

J

Jackson, I., 45
Java 2 Enterprise Edition (J2EE), 44

K

Kazman, R., 43, 45

L

Lee, C., 129
Lifecycle architecture (LCA), 112–113
Lifecycle objective (LCO), 109–110
Lost opportunity costs, 100–101

M

Marketable, use of term, 29
Martin, Robert, 132
Microsoft .Net, 44

Denne_IN_p187-190 9/10/03 4:42 PM Page 189

Minimum marketable features (MMFs), 5
behavior, categories of, 79–81
candidate, 30–31
construction, 116–118
cost-benefit analysis of delivery sequence, 63–66
costs of developing, 36–37
decompose, into use cases, 107–108
defined, 28–30
design, 114–116
development and delivery, 113–114
elicit, 106–107
graph, constructing, 108
how value is created, 6
hybrid, 97
precedence graph, 38–39
precursors, 37–38, 40
return on investment, 22–24
risk control and, 9
risk factors and, 35–36, 108–109
selecting, 6–7, 119, 160–161
sequencing architectural elements and, 74–75, 113–114
time needed to develop, 37
transition, 118
user stories, 124–127
value, determining, 31–37
Minimum marketable feature sequencing strategies
greedy approach, 67–69, 168–170
simple look-ahead approach, 69–72
weighted look-ahead approach, 73–74, 170–173
Motorola, 9
Multiple inheritance, 55–57

N

Net present value (NPV), 17–18, 19, 114
baseline, 99
lost opportunity costs, 100–101
potential, 99
release plan and assessment of, 129–131
sequence-adjusted (SANPVs), 33–34, 37, 95–97, 129–131, 166–168
Next move principle, 56–57
Nonfunctional requirements (NFRs), 115

O

Object Management Group, 44

P

Pairwise comparisons, 91, 92–97
Parallel development, 83–87
Payback time, 16
Perry, D. E., 45
Plans, mini, 115–116
Precedence graph, 38–39, 163–164
Precursors, 37–38, 40
Prepayment period, 147
Present value (PV), 17, 19
Principles of Software Engineering Management (Gilb), 57, 151
Project characteristics, manipulating, 149–150
Project funding, getting, 146–148

R

Rational unified process (RUP), 3
architecture defined, 44
elaboration, 110–113
inception, 105–110
minimum marketable feature construction, 116–118
minimum marketable feature design, 114–116
minimum marketable feature development and delivery, 113–114
phases of incremental funding methodology compared to, 103–105
Release, 118
assessment of net present value, 129–131
eXtreme programming and, 123–124
planning meetings, 131–132
user stories and, 127–129
Return on investment (ROI)
applications, 13–14
breakeven time, 16, 18, 19
business case, 15, 106
case example, 175–176
cash flow projections, 15–16
defined, 19
example, 19–21
importance of, 14
internal rate of return, 18–19
minimum marketable feature, 22–24
net present value, 17–18, 19
payback time, 16
present value, 17, 19
traditional, 4
Risk
control, 8–9, 24–25
factors with minimum marketable features, 35–36, 108–109
sequencing and mitigation of, 76
Rules/standards, 44–45
Rumbaugh, J., 45

Denne_IN_p187-190 9/10/03 4:42 PM Page 190

S

SCRUM, 122, 142
Self-funding status, 16, 19
Sequence-adjusted net present values (SANPVs), 33–34, 37, 95–97, 129–131, 166–168
Sequencing strategies
  See also Minimum marketable feature sequencing strategies
  architectural elements and, 74–75, 113–114
  cost-benefit analysis of delivery, 63–66
  cost versus value analysis, 62–63
  reevaluation of, 76
  risk mitigation, 76
Six Sigma approach, 9–11
Software Architecture in Practice (Bass, Clements, and Kazman), 43
Software development
  failure of, 1–3
  lack of customer involvement and failure of, 7–8
Spiral architecture, 57, 104
Story coupling, 129
Sun Microsystems, 9
  SunTone Architecture Methodology, 3, 46–47

T

Theory-W, 115
Time, development, 37
Timeless Way of Building, The (Alexander), 45
Top-down approach, 30–31
Transition, 118

U

Unified Modelling Language (UML), 45
Use cases, 107–108
User stories
  clustering, 124–127
  releases and, 127–129

V

Value
  cost versus analysis of, 62–63
  determining, 31–37
Vision statement, 106

W

Wake, Bill, 127
Waterfall approach, 4
Wolf, A. L., 45
Work-breakdown structure (WBS), 36, 37, 129

X

Xiaoping, J., 129
XP. *See* eXtreme programming

Z

Zierden, Troy, 13