

# Index

## A

---

Acceptance testing, 87, 177, 186  
Access (Microsoft), 238  
Active listening, 91. *See also* Communication;  
Listening  
Active Server Pages (Microsoft). *See* Microsoft  
Active Server Pages (ASP)  
ActiveX controls (Microsoft), 238  
“Adopt a Skeptic” strategy, 55  
African Americans, 199  
Agile software development, 25, 228. *See also*  
XP (eXtreme Programming); SCRUM  
*Agile Software Development* (Cockburn),  
177–178  
Alexander, Christopher, 213  
*Analysis Patterns* (Fowler), 213  
Apprenticeship, 30–31, 112–114. *See also*  
Mentoring  
Articulation, reflective, 78, 206. *See also*  
Communication  
ASP (Active Server Pages). *See* Microsoft  
Active Server Pages (ASP)  
AT&T (American Telephone & Telegraph),  
223  
Attention span, 71  
Auer, K., 10, 70  
Average programmers  
pairing of, with experts, 105–109  
two types of, 106

## B

---

Basic comparison model, 223–224

BDUF (“big design up front”), use of the  
term, 174–176  
Beck, Kent, 9, 10, 14, 172  
on cultural issues, 150  
on pair rotation, 30–31  
on TDD, 346  
Beedle, M., 10  
Bellcore, 198, 199  
Bell Labs Research, 9, 195  
Benbasat, I., 53  
Bengtsson, Anders, 157  
Bennerhult, Erik, 115, 150  
Bergin, Thomas J., 13  
Black box tests, 185  
“Blob class” designs, 39  
Boehm, B., 172  
“Bozo bit, flipping the,” 165–166  
Brainstorming, 4, 27, 61, 185  
pair rotation and, 75, 85  
timely completion and, 36  
Breakeven Unit Value. *See* BUW (Breakeven  
Unit Value)  
Breakeven Unit Value Ratio. *See* BUVR  
(Breakeven Unit Value Ratio)  
Breaks, taking, 203  
Brooks, Fred P., 8, 41, 43  
Brooks’s Law, 41  
Brown bag lunches, 43, 49, 54  
BUW (Breakeven Unit Value), 230–231, 234  
BUVR (Breakeven Unit Value Ratio), 231–234

## C

---

C++ (high-level language), 250

- Canada, 148
- Canna, Jeff, 157
- Capability Maturity Model. *See* CMM (Capability Maturity Model)
- Chairs, positioning of, 67
- Challenge(s)
- expert-average pairs and, 106–107
  - expert-expert pairs and, 99–103
  - expert-novice pairs and, 113–114
  - extrovert-extrovert pairs and, 124–125
  - extrovert-introvert pairs and, 131
  - introvert-introvert pairs and, 135
  - novice-novice pairs and, 119
  - use of the term, 94
- Chaplin, Dave, 126, 167
- Charlton, Stu, 114
- Chrysler Comprehensive Compensation (C3) team, 15, 136, 175
- CMM (Capability Maturity Model), 181–182
- Coaches. *See also* Managers
- conflict resolution and, 90
  - encouragement from, 63–64
  - novice-novice pairs and, 119
- Cockburn, Alistair, 30, 78
- Agile Software Development*, 177–178
  - on communication, 205–206
  - on training, 81
  - on use cases, 185
  - on XP, as a “high discipline” process, 177–178
- Code. *See also* Debugging; Defects
- enhancing, 87–88
  - and-fix development model, 171, 172
  - implementation, 176–177
  - inspections, 196
  - maintaining, 87–88
  - ownership, collective, 176–177
  - refactoring, 176–177
  - standards, 25, 91, 177
- Collaborative Software Process. *See* CSP (Collaborative Software Process)
- Colocation, 59–60, 197–199
- Common Lisp, 8
- Communication. *See also* Listening
- conflict resolution and, 61, 90–92, 207–208
  - CSP and, 184
  - ego problems and, 165–167
  - formal, 84–85
  - gender and, 139–143
  - group size and, 84–85
  - improved, within teams, 43–44
  - inter-, cost of, 44
  - overall importance of, 205–206
  - pair rotation and, 73–82
  - reflective articulation and, 78, 206
- Communities of Practice (CoP), 77, 78
- Competition, among pairs, 205
- Compiling, 18–19, 184
- CSP and, 183
  - quality assurance and, 85–86
- Compromises, ability to make, 26–27, 207–208
- Concentration, 60–61
- Confidence, 61–63, 205
- ego problems and, 164, 165–167
  - lack of, as a impediment, 63
  - over-, 61–62
  - professional driver problem and, 154
- Conflicts, handling, 61, 90–92, 207–208
- Conrad, Wayne, 101, 108, 114
- on ego problems, 162, 166
  - on handling conflicts, 90
- Constantine, Larry, 8–9
- Continuous
- code reviews, 18–19, 30
  - integration, 177
- Cooke, Matthew, 125
- CoP (Communities of Practice), 77, 78
- Coplien, James, 9, 41, 196
- on organizational patterns, 212–213
  - on programming with a projector, 197
  - on triplet programming, 195
- Cost of Change curve, 176, 247
- Courage, building, 27–28

- Coverage, as a component of defect recovery efficiency, 228–229
  - Covey, Stephen, 206
  - Crashes, cause of, 39
  - CRC card sessions, 61, 175, 178, 185
    - CSP and, 188
    - TDD and, 248
  - Creativity, 60–61
  - CSP (Collaborative Software Process)
    - case study, 181–191
      - Focus Area 0 (baselining your process), 182–184, 186, 189
      - Focus Area 1 (quality management), 182, 184–187
      - Focus Area 2 (project management), 182, 187–190
  - Cubicles, arrangement of, 60, 69
  - Cultural issues, 56, 145–151
  - Cunningham, Ward, 14–15, 78, 172, 206
  - Customers
    - cost of defects for, 39
    - pairing of developers with, 196
    - requirements definition and, 173–174
- D**
- 
- Database(s)
    - experts, 59
    - knowledge management and, 77
  - Debugging, 29, 70, 199, 247. *See also* Defects
  - Defect(s), 35, 40, 88. *See also* Code; Debugging; Quality
    - CSP and, 186–187
    - cost of, for customers, 39
    - economic analysis and, 223, 228–229
    - ego-less programming and, 204
    - filters, 85–86
    - multidisciplinary pairs and, 196
    - pair debugging and, 29
    - pair learning and, 30
    - pair reviews and, 28–29
    - recovery efficiency, 228–229
    - quality assurance and, 85–86
  - DeMarco, Tom, 1, 15
    - E-Factor in, 24–25
    - flow state in, 19, 20
  - Dependency problems, 58
  - Desks, arrangement of, 60
  - Disagreements, resolving, 61, 90–92, 207–208
  - Disbelief, suspending, 56
  - Discounting, 227
  - Discount rate, 227
  - Distributed
    - cognition, 26
    - pair programming, 60, 197–199
  - “Do Food” strategy, 49, 54
  - Drink vending machine program, 74
  - Driver(s). *See also* Pairs
    - asking questions and, 92
    - conflict resolution and, 90
    - ego problems and, 159–167
    - expert-average pairs and, 106
    - multidisciplinary pairs and, 196
    - professional, problem of, 153–156
    - projection screens and, 196–197
    - role of, rotating, 63
    - tips for working with, 89
    - use of the term, 4
    - workplace layout and, 67, 69
    - XP and, 178
  - Dual video cards, 68–69
- E**
- 
- Early Adopters, 54, 55
  - Early Majority, use of the term, 48
  - Earned value. *See* EV (Earned Value)
  - Economic analysis, 221–236
    - economic comparison model for, 226–232
    - introduction to, 221–223
  - Edge Learning Institute, 35
  - Education. *See also* Teaching; Training; Tutorials
    - benefits of pair programming to, 237–243
    - issues to consider, 240–241

- through mentoring, 36, 74, 80–81, 92, 106–109, 112–116
  - Edvinsson, Leif, 77
  - E-Factor (Environmental Factor), 24–25
  - Efficiency
    - defect-recovery, 228–229
    - metrics, 224–225
  - Ego(s), 16, 100, 118–119, 184. *See also*
    - Personality types
    - excessive, problems with, 159–162, 204
    - less programming, 203–204
    - making compromises and, 207–208
    - too little, problems with, 163–167
  - Eliot, Scott, 79
  - Encouragement, importance of, 52–53, 56, 63
  - Environmental Factor (E-Factor), 24–25
  - Erdogmus, Hakan, 221
  - Ergonomics, 67. *See also* Workplace layout
  - Errors. *See* Debugging; Defects
  - eSoft, 88
  - Euchner, James, 77–78
  - EV (Earned Value), 189–190, 227
  - Evaluation phase, 55. *See also* Reviews
  - Evangelists, use of the term, 46, 54
  - Expert(s)
    - asking for help and, 28
    - characterization of, 95
    - ego-less programming and, 204
    - ever-popular, problem of, 59
    - expert pairs, 17, 97–103
    - in-Earshot management pattern, 30
    - mentoring by, 16, 62–63, 80–81
    - novice pairs, 16–17, 62–63, 80–81, 111–116, 120–121
    - pair learning and, 30–31
  - eXtreme Programming (XP)
    - acceptance testing and, 177, 186
    - books on, 10, 14, 16, 58, 70
    - case study, 171–179
    - code implementation, 176–177
    - coding standards, 177
    - commenting guidelines, 177
    - continuous integration and, 177
    - CRC cards and, 185
    - CSP and, 181, 183, 189–190
    - design philosophy, 175–176
    - development of, 9, 172–173
    - distributed pair programming and, 198
    - economic analysis and, 228
    - meetings, 76
    - pair programming as an integral part of, 177–178
    - pair rotation and, 76
    - quality metrics and, 187
    - release planning, 172–173
    - requirements definition, 173–174
    - testing and, 87, 91, 176–178, 186, 245, 248–249
    - working alone and, 15
  - Extreme Programming Applied: Playing to Win* (Auer and Miller), 10, 16, 58
  - Extreme Programming Explained: Embrace Change* (Beck), 10, 14
  - eXtreme Programming Software Studio, 69–70
- 
- ## F
- 
- Fear, 54, 88
  - Feedback mechanisms, 18, 55
    - CSP and, 188, 190
    - development models and, 172
    - PSP and, 187
    - TDD and, 249
  - Felleisen, Matthaias, 3
  - Flex hours, 60
  - “Flipping the bozo bit,” use of the term, 165–166
  - Flor, N. V., 26
  - Flow
    - of Events, 184–185
    - state, 20–21
  - Form validation, 6, 7–8
  - Fowler, Martin, 176, 213, 249
  - France, 148

Franklin, Aretha, 98  
Functional testing, 87, 185

## G

Gabriel, Dick, 8, 12–13  
“Gang of Four,” 213  
Gartner Group, 39  
Gender issues, 139–143  
George, Bobby, 248  
Germany, 148, 149, 198  
Gibson, Richard G., 13  
Glass, Robert, 60–61  
Goals  
    changing behavior to reach, 35–36  
    pair negotiation and, 26  
    WIFM thoughts and, 35–36  
*Gold Collar Worker* (Kelley), 25  
“Go/no-go” decisions, 188  
Green, Gina, 51–53  
Griss, Martin, 9, 100–101  
Group(s)  
    size of, 84–85  
    testing, 86  
GUI (graphical user interface) programming,  
    59, 74–75, 102

## H

Handicapped individuals, 154, 155  
Hartman, Bob, 88  
Harvard University, 39, 53  
Hayes, Steve, 95, 102, 120–121, 142  
Herman, Eric, 125, 162  
Hevner, Alan, 51–53  
Hewlett-Packard, 100, 223  
Hill Air Force Base, 36–37  
Hometown Stories, use of the term, 49, 54  
Houston, Greg, 109, 114  
HTML (HyperText Markup Language),  
    237–238  
Humility, practicing, 203–204  
Humphrey, Watts S., 85, 181–182, 190, 224  
Hutchins, E. L., 26

## I

IBM (International Business Machines), 35,  
    83, 223  
*IEEE Software* (journal), 60–61, 246  
Incremental realized value. *See* IRV  
    (Incrementally Realized Value)  
Increments, defined, 172  
India, 40, 41, 149, 198  
India Technology, 40, 41  
Innovation  
    adoption rates, 47  
    transitioning to pair programming and,  
        46–49, 52–57  
Innovators, use of the term, 46–49, 54  
Integration issues, 69  
Intent, use of the term, 94  
Intercommunication costs, 44  
Interruptions, avoiding, 19–20, 24–25  
*Introducing Patterns into Organizations*  
    (Manns and Rising), 46–47  
Introvert(s)  
    excess ego problems and, 160  
    importance of communication skills to,  
        205–206  
    -introvert pairs, 133–138  
“Involve Everyone” strategy, 55  
IRV (Incrementally Realized Value), 229,  
    232–233

## J

Jacobson, Ivar, 9  
Jangr, Jeff, 248  
Japan, 148  
Java, 250  
JavaScript, 238  
Jeffries, Ron, 98, 115, 120, 125, 172  
    on ego problems, 166  
    on gender issues, 142  
    on the professional driver problem, 156  
Jones, C., 223, 228  
Jonker, Todd, 136, 166

Jonsson, Patrick, 9  
JUnit (Parasoft), 250  
“Just Do It” principle, 47, 54

## K

---

Kernel, 59  
Keyboards  
    professional driver problem and, 153–156  
    using two, 68–69  
    wireless, 68  
    workplace layout and, 67–69  
Kleb, Bil, 41, 69, 91  
Knowledge. *See also* Education; Learning;  
    Training  
    expert-expert pairing and, 102  
    management strategies, 17, 77–79  
    pair learning and, 30–31  
    pair rotation and, 73–82  
    pair understanding exercises and, 212–214  
    social construction of, 25–26  
    synergy and, 25–26  
    transfer, improved, through pair  
        programming, 5, 30–31, 73–82  
Kohnke, Jennifer, 94  
Kotter, John, 53

## L

---

Labor cost, 226  
Lance, Mike, 120, 135–136  
Langr, Jeff, 126, 136  
Language(s)  
    foreign, 148–149  
    pattern, 8, 9, 46–49, 53–54  
Laptop computers, 69  
Latency, 228–229  
Lave, J., 30  
LCDs (liquid crystal displays), 8. *See also*  
    Monitors  
Leaders, local, use of the term, 48, 53–54. *See also*  
    Coaches; Managers  
Learning. *See also* Education; Knowledge;  
    Training

    enhanced, through pair programming, 5,  
        237–243  
    expert-expert pairing and, 102  
    listening and, 207  
    through mentoring, 36, 74, 80–81, 92,  
        106–109, 112–116  
    pair, 30–31, 119–121, 199–200  
    pair rotation and, 77–81  
    paradigms, 199  
Lichtenwalner, Lee, 149  
Life-cycle evolution, 171–172, 182  
Lindner, Michael, 102, 108–109, 142, 157  
Lisp, 8, 12, 13, 100–101  
Listening. *See also* Communication  
    active, 91  
    introverts and, 134–135  
    overall importance of, 206–207  
Lister, T., 15, 19, 20, 24–25  
Local Leaders, use of the term, 48, 53–54  
Lowe, Iain, 167  
Lucid, 12

## M

---

McBreen, Pete, 205–206  
McCarthy, Jim, 56, 165–166  
McCarthy, Michele, 56  
MacCormack, Alan, 39  
Mackinnon, Tim, 157  
MacLisp, 12  
Malik, William, 39  
Management. *See also* Managers  
    knowledge, 17, 77–79  
    organizational patterns and, 46–47, 76,  
        212–214  
    project, 187–190  
    quality, 184–187  
    time, strategies for, 25  
    by Walking Around (MBWA), 63, 160–161,  
        240–241  
Managers. *See also* Management  
    advice for, 53–56  
    conflict resolution and, 90

- divide-and-conquer tactics used by, 69
  - ego problems and, 160–161, 164–167
  - encouragement offered by, 52–53, 56, 63
  - expert-expert pairs and, 99
  - objectives of, 34–45
  - pair rotation and, 73–82
  - peer acceptance strategies and, 46, 48
  - performance appraisals and, 83–84
  - resistance to pair programming among, 14–21, 34–45
  - who mandate the use of pair programming, 50–56
- Manns, Mary Lynn, 46, 53
- Marginal Value Earned. *See* MVE (Marginal Value Earned)
- Master/apprentice relationship, 30–31, 112–114. *See also* Mentoring
- Mavericks, use of the term, 93. *See also* Solo programmers
- MBWA (Management by Walking Around), 63, 160–161, 240–241
- Mediagenix, 196, 197
- Meetings, 5, 76
- Mental flow, 19–20
- Mentoring. *See also* Education; Teaching; Training
  - asking questions and, 92
  - expert-average pairs and, 106–109
  - expert-novice pairs and, 112–116
  - pair rotation and, 74, 80–81
  - timely completion and, 36
- Merel, Peter, 162
- Meyers-Briggs personality test, 134. *See also* Personality types
- Mice
  - using two, 68–69
  - wireless, 68
- Michaelsen, L., 27
- Microphones, 197, 198–199
- Microsoft Access, 238
- Microsoft Active Server Pages (ASP), 237, 238
- Microsoft ActiveX controls, 238
- Microsoft NetMeeting, 7, 197, 198
  - Microsoft PowerPoint, 215–216
- Microsoft VBScript, 238
- Microsoft Visual Basic, 100, 250
- Microsoft Windows XP, 83
- Miller, Roy W., 10, 70, 89, 136–137
- Minimalist approach, 177
- MIT (Massachusetts Institute of Technology), 12, 150
- Molteni, A. K., 102, 150
- Monitors, 8, 67, 196–197. *See also* Projection screens
- Moore, G., 53
- Morale, 5, 42
- Motivation, 22–25, 35–45
- Motorola, 223
- Murphy, Jim, 109, 115, 126, 148
- MVE (Marginal Value Earned), 232
- Mythical Man, The* (Brooks), 8
- Mythical Man Month, The* (Brooks), 43

## N

- 
- NASA Langley, 41, 69, 91
  - National Science Foundation, 199
  - Navigator(s). *See also* Pairs
    - conflict resolution and, 90
    - ego problems and, 164
    - expert-average pairs and, 106
    - misconceptions about, 18–19
    - professional driver problem and, 154–156
    - quality assurance and, 85
    - role of, rotating, 63
    - use of the term, 4, 18
    - workplace layout and, 67, 69
    - XP and, 178
  - NetMeeting (Microsoft), 7, 197, 198
  - Net Present Value. *See* NPV (Net Present Value)
  - Neumann, John von, 204–205
  - “Never look back” policy, 172
  - New York Stock Exchange, 39
  - Noise, problems created by, 60, 71

Nonverbal behavior, 26  
North Carolina State University, 197, 199–200, 248  
Northeastern University, 3  
Nosek, J. T., 37  
Novice(s)  
  asking for help and, 28  
  characterization of, 95  
  -expert pairs, 16–17, 62–63, 80–81, 111–116, 120–121  
  -novice pairs, 117–121, 238  
  pair learning and, 30–31  
NPV (Net Present Value), 222, 229–234  
Nynex, 77–78

## O

Objectives  
  of management, 34–45  
  of pair programming tutorials, 212  
Organizational  
  guidelines, 55  
  patterns, 46–47, 76, 212–214  
*Organizational Patterns* (Coplien), 212  
Overconfidence, 61–62. *See also* Confidence

## P

Pair(s). *See also* Pair programming  
  brainstorming, 27  
  courage, 27–28  
  cultural issues and, 145–151  
  debugging, 10, 29  
  developer-customer, 196  
  ego problems and, 159–162  
  expert-average, 105–109  
  expert-expert, 17, 97–103  
  expert-novice, 16–17, 62–63, 80–81, 111–116, 120–121  
  extrovert-extrovert, 123–127  
  extrovert-introvert, 128–133  
  flow state and, 19–20  
  introvert-introvert, 133–138  
  learning, 30–31, 119–121, 199–200

  multidisciplinary, 195–196  
  negotiation, 25–27, 89–90  
  novice-novice, 117–121, 238  
  pressure, 22–25, 38, 53, 62  
  problems faced by, 16–17, 58–64  
  reviews, 28–29  
  selecting, principles for, 93–96  
  seven habits of effective, 203–209  
  synergistic behavior of, 22–33  
  tips and tricks for, 89–92  
Pair programming. *See also* Pairs  
  advantages/disadvantages of, 4–5  
  casual use of, 34, 63  
  defined, 4  
  as enjoyable, 22  
  Gabriel's historical account of, 12–13  
  as an integral part of XP, 177–178  
  intensity of, 16  
  introduction to, 1, 3–13  
  life-cycle affordability of, 40–41  
  mandated, 50–56  
  presentations, 50–51, 215–217  
  readiness surveys, 50  
  seven myths of, 14–21  
  seven synergistic behaviors of, 22–33  
  timelines, 8–9  
  tips and tricks, 89–92  
  transitioning to, by choice, 50–57  
  two levels of, 34  
Pair rotation  
  advantages of, 73  
  assignment of, 75–76  
  Beck on, 30–31  
  daily meetings and, 76  
  examples of, 5–8, 217–219  
  group size and, 84–85  
  intercommunication costs and, 44  
  knowledge management and, 77–79  
  logistics and, 75–76  
  overview of, 73–82  
  pair learning and, 30–31  
  survey data on, 16–17

- training and, 79–81
  - Parasoft JTest, 250
  - Parkinson's Law, 22, 86
  - Pasteur project, 9
  - Patterns
    - language, 8, 9, 46–49, 53–54
    - organizational, 46–47, 76, 212–214
    - pair understanding and, 212–214
  - Paulk, Mark, 181
  - pcAnywhere, 198
  - Peer(s)
    - acceptance strategies, 46–49
    - evaluations/reviews, 18, 83–84, 181
    - gaining the support of, 46–49
  - Peeters, Vera, 125, 141
  - Pekeler, Christian, 148
  - Pentagon, 223
  - Peopleware* (DeMarco and Lister), 15, 19–20, 24–25
  - Performance appraisals, 83–84. *See also*
    - Reviews
  - Personality types, 61–63, 71. *See also* Egos;
    - Extroverts; Introverts
    - Meyers-Briggs test for, 134
    - performance appraisals and, 83–84
    - skill imbalances and, 62
  - Personal Software Process. *See* PSP (Personal Software Process)
  - Personal Touch, use of the term, 47, 54
  - Pfister, Markus, 158
  - Pieces of Clay, use of the term, 49
  - Pilots, 35, 55
  - Portland Pattern Repository* (Cunningham), 14–15
  - Power, desire for, 154
  - PowerPoint (Microsoft), 215–216
  - Presentations, 50–51, 215–216. *See also*
    - Tutorials
  - Present Value. *See* PV (Present Value)
  - Pressure, increased, 22–25, 38, 53, 62
  - Privacy policy, 6
  - Procedures. *See* Standards
  - Processes, baselining, 182–184. *See also* PSP (Personal Software Process)
  - Productivity measures, 36–37, 224, 224, 225. *See also* Reviews
  - Programming, use of the term, 10
  - Projection screens, 196–197, 198–199. *See also* Monitors
  - PSP (Personal Software Process), 181–190
    - earned value in, 189–190
    - quality metrics and, 187
    - test cases and, 186
    - time-tracking procedures, 183
  - Psychology, 26, 203–204. *See also* Personality types
    - Psychology of Computer Programming, The* (Weinberg), 203–204
  - PV (Present Value), 226–227
- Q**
- 
- Quality. *See also* Defects
    - assurance (QA), 85–86, 88
    - CSP and, 184–187
    - improved, through pair programming, 4
    - of Service parameters, 41
  - Questions, asking, 28, 92, 107
- R**
- 
- Raytheon, 223
  - “Read my Lips: No New Process Models” (Weigers), 182
  - Recognition, sharing, 17–18, 56
  - Refactoring, 94, 155, 161–162, 165–166, 250
  - Reflective articulation, 78, 206. *See also*
    - Communication
  - Regression testing, 247, 249–250
  - Rekitt, Norman, 120, 162
  - Release
    - defined, 172
    - planning, 172–173
  - Religion, 102
  - repetitive strain injuries (RSIs), 98, 100, 154

- Requirements
    - CSP and, 185, 186
    - definition, 173–174
  - Resistance, to paired programming, 14–21, 34–45
    - inviting, as a strategy, 54–55
    - by management, 14–21, 34–45
    - misconceptions and, 14–21
    - pair rotation and, 75–76
    - transitioning to pair programming and, 50–56
  - Respect, importance of, 98–99
  - Respected Techie, use of the term, 49, 54
  - Reviews
    - continuous, 18–19, 30
    - design, 85–86
    - formal code, 35
    - pair learning and, 30
    - peer, 18, 83–84, 181
    - quality assurance and, 85–86
  - Rework
    - effort, 224–225
    - time, 183
  - Rice University, 3
  - Rising, Linda, 46, 53
  - Risk(s)
    - assessments, 172
    - of losing key team members, reducing, 41–42
      - mitigation strategies, 46
  - Rogers, Jason, 47, 49, 125–126, 149
  - Root causes, use of the term, 94
  - Royce, Winton, 171–172
  - RSIs (repetitive strain injuries), 98, 100, 154
- S**
- 
- Scalability, 41
  - Schedules, 16, 36–41
    - coordinating, between pairs, 22
    - flex hours and, 60
    - pair rotation and, 74
    - problems with, 58–59, 60
  - Schwaber, K., 10
  - SCRUM
    - books on, 10
    - economic analysis and, 228
    - meetings, 5, 76
    - pair rotation and, 76
  - SEI (Software Engineering Institute), 181, 182, 223
  - Self-esteem, 165. *See also* Confidence
  - Shopping carts, 6
  - Shukla, Anuja, 80
  - Siemens, 198
  - Silvermark Test Mentor, 250
  - Sisk, John, 167
  - Skepticism, being prepared for, 54–55
  - Sociology, 206
  - Socratic dialogues, 92
  - Software Development with SCRUM* (Beedle and Schwaber), 10
  - Software Factory model, 227–232, 234
  - Software Reuse* (Griss, Jacobson, and Johnson), 9
  - Solo programmers, 15–16, 19–20, 28
    - creativity and, 60–61
    - economic analysis and, 222–226
    - experience of mental flow by, 19
    - Glass on, 60–61
    - impact of noise on, 71
    - problems encountered by, after getting used to pair programming, 58
    - references to, as mavericks, 93
  - Spiral development model, 172
  - Standards, 25, 91, 177
  - Stand-up meetings, 5
  - Stanford University, 12
  - Steele, Guy L., Jr., 13
  - Stereotypes, 205–206
  - Stotts, David, 197, 198
  - Strategic
    - defects, defined, 4
    - programmers, 102

- Stutzke, Richard, 80
- Submit button, 8
- Success
- characteristics of, 94, 98–99
  - of the diffusion of software development techniques, 52–53
    - expert-average pairs and, 106–107
    - expert-expert pairs and, 98–99
    - expert-novice pairs and, 112–113
    - extrovert-extrovert pairs and, 124
    - extrovert-introvert pairs and, 130
    - introvert-introvert pairs and, 134–135
    - novice-novice pairs and, 117–118
    - performance appraisals and, 83–84
    - sharing, importance of, 18
- Synergy
- in expert-expert pairs, 17
  - flow state and, 19–20
  - listening and, 207
  - seven behaviors characterized by, 22–33
- System
- metaphor, 175
  - testing, 87
- T**
- 
- Tables, positioning of, 67, 69–70
- Tactical
- defects, defined, 4
  - programmers, 102
- TAs (teaching assistants), 240–241
- TDD (test driven development), 91, 345–351
  - with high/low-level design, 249–250
  - without high/low-level design, 248–249
- Teaching. *See also* Education; Learning
- assistants (TAs), 240–241
    - expert-average pairs and, 106–109
    - expert-novice pairs and, 112–116
    - through mentoring, 36, 74, 80–81, 92, 106–109, 112–116
- Team(s). *See also* Pairs
- building, 5, 207
    - distributed pair programming and, 197–199
    - experts on, problems with, 58
    - improved communication within, 43–44
    - losing key persons on, reducing the risk of, 41–42
    - morale, 42
    - player, role of, importance of, 207
    - problems faced by, 58–64
    - recognition, shifting from individual recognition to, 17–18, 56
    - scaling rules for, 84–85
    - Software Process (TSP), 190
    - splitting, into progress and training teams, 81
- Telecommuting. *See* Distributed pair programming
- Temple University, 37
- Test cases, 87, 91. *See also* Testing
- black box, 185
  - CSP and, 185, 186
  - white box, 186
  - XP, 176–177
  - unit, 176–177, 178
- Testing. *See also* TDD (test driven development); Test cases
- acceptance, 87, 177, 186
  - CSP and, 183, 185, 186
  - development models and, 172
  - extensive, combining pair programming with, 41
  - functional, 87, 185
  - multidisciplinary pairs and, 196
  - peer acceptance strategies and, 48
  - quality assurance and, 85–86
  - regression, 247, 249–250
  - system, 87
  - unit, 176–178, 185, 186, 250
  - XP and, 87, 91, 176–178, 186, 245, 248–249
- Test the waters, notion of, 46, 55
- Time
- assimilation, 80–81
  - cycles, shortened, 4, 8–9, 15
  - lines, pair programming, 8–9

- management strategies, 25
- tracking procedures, 183
- training, reducing, 36, 42–43, 80–81
- value of money, 226–227
- TogetherSoft Control Center, 197
- Total Discounted Cost, 229, 233–234
- Training. *See also* Education; Learning;  
Teaching
  - costs, 42–43
  - distributing the burden of, 80
  - through mentoring, 36, 74, 80–81, 92, 106–109, 112–116
  - pair rotation and, 73–82
  - “sink or swim” form of, 80
  - time, reducing, 36, 42–43, 80–81
- Transportation device design, 215–217
- Triplet programming, 195
- “Truck number” metric, 42
- Trust, 5, 44, 205, 206
  - ego-less programming and, 204
  - overall importance of, 31
  - pair rotation and, 76, 81
- TSP (Team Software Process), 190
- Tutorials, 49, 54–55, 211–219
  - individual design activity, 215
  - individuals working on teams activity, 216–218
  - outline of, 211–212
  - pair programming presentations, 215–217
  - pair rotation activity, 217–219
  - pair understanding activity, 212–215
  - research results, 214
  - summary and conclusion, 218–219
  - welcome and exercise objectives, 212
- Typing skills, 63, 154

## U

- UML (Unified Modeling Language), 248–249
- Unit
  - testing, 176–178, 185, 186, 250
  - value, 227
- University of California, Santa Cruz, 199, 200

- University of Cape Town, 28
- University of Hawaii, 28
- University of Illinois, 12
- University of North Carolina, Chapel Hill, 196, 197–198
- University of Utah, 15, 37–39, 53, 100, 125
  - economic analysis at, 221–236
  - handicapped student at, 155
- USENET, 28
- Use cases, 184–185

## V

- Value(s)
  - Breakeven Unit (BUV), 230–231, 234
  - Earned (EV), 189–190, 227
  - Incrementally Realized (IRV), 229, 232–233
  - Ratio, Breakeven Unit Value (BUVR), 231–234
  - Marginal, Earned (MVE), 232
  - Net Present (NPV), 222, 229–234
  - Present (PV), 226–227
    - realization models, 227–228
- VBScript (Microsoft), 238
- Video cards, 68–69
- VideoWindow, 199
- Visual Basic (Microsoft), 100, 250
- Voice-Over-IP, 41

## W

- Ward, Stephanie, 120, 141–143, 156–157
- Waterfall development model, 171–173
- Watson, M., 27
- Webcams, 197–199
- Web programming, 79, 118–119, 237–238
- Wege, Christian, 136
- Weigers, Karl, 182
- Weinberg, G. M., 42, 203–204
- Wells, Don, 29
- Wenger, E., 30
- White box test cases, 186
- Whitesmiths, Ltd., 8

- WIIFM (What's In It For Me) thoughts, 35–36, 42–43
- Win-win negotiating tactics, 188
- Windows XP (Microsoft), 83
- Wood, Bill, 41, 69, 91
- Work
- “expands to fill the time available” (Parkinson’s Law”), 22, 86
  - load, increased, concerns about, 39–40
  - unit, 222
- Workplace layout
- arrangement of cubicles, 60, 69
  - arrangement of desks, 60
  - overview of, 67–71
- positioning of chairs, 67
- positioning of keyboards, 67–69
  - use of projection screens in, 196–197, 198–199
- Wright, Bill, 8
- Writing Effective Use Cases* (Cockburn), 185
- 
- X**
- 
- XP (eXtreme Programming)
- acceptance testing and, 177, 186
  - books on, 10, 14, 16, 58, 70
  - case study, 171–179
  - code implementation, 176–177
  - coding standards, 177
  - commenting guidelines, 177
  - continuous integration and, 177
  - CRC cards and, 185
  - CSP and, 181, 183, 189–190
  - design philosophy, 175–176
  - development of, 9, 172–173
  - distributed pair programming and, 198
  - economic analysis and, 228
  - meetings, 76
  - pair programming as an integral part of, 177–178
  - pair rotation and, 76
  - quality metrics and, 187
  - release planning, 172–173
  - requirements definition, 173–174
  - testing and, 87, 91, 176–178, 186, 245, 248–249
  - working alone and, 15
- XP Applied* (Auer and Miller), 70
- xUnit, 250
- 
- Y**
- 
- YAGNI (You Aren’t Gonna Need It)
- philosophy, 175–177