

# Index

## A

Adaptive Development Lifecycle, 107  
Adaptive Software Development (ASD),  
147, 149  
Adaptive walk, 52  
Agile Manifesto, 148–49, 155  
*Agile Software Development Ecosystems*  
(Highsmith), 147  
Agility  
  agile methods, 147  
  agile organizations, 148  
  agile values and principles, 148–49  
  defined, 147–48  
  factors emphasized, 148  
  and growing software, 154–56,  
  157–59  
Analysis paralysis, 72  
*On the Art of Cutting Metals* (Taylor), 10  
Arthur, Brian, 57, 93–94  
Assembly line mentality, 6  
Assumptions, simplification of, 23  
*At Home in the Universe* (Kauffman), 47  
Attractors, 41, 52  
Autocatalysis, 48–49, 133  
Autocatalytic set, 49

## B

Bak, Per, 54, 64  
Barth, Carl, 12  
Beck, Kent, 90, 150  
Behaviorist psychology, 14  
Bethlehem Steel Company, 12  
Boids (computer program), 61  
Bonaparte, Napoleon, 25  
Bossert, Patrick, 12

Branch method, 60  
Business Process Re-engineering (BPR),  
24

## C

Capability Maturity Model (CMM), 24  
CAS (complex adaptive systems)  
  autocatalysis, 48–49  
  coevolution, 51–55  
  and complexity science, 34  
  concepts, 47  
  edge of chaos, 50–51  
  natural selection, 49–50, 52–53  
  “order for free,” 47, 50, 51  
  ordered versus chaotic regimes, 50–51  
  perpetual novelty, 42  
  self-organization, 47–49, 52–53  
  self-organizing criticality, 54–55  
  social and technical evolution, 55–56  
Catalysis, 48  
Catalytic closure, 49  
Change  
  breaking the patterns, 136–37  
  confrontation, 136  
  and growing software, 162  
  new ideas, impact of, 161  
  translation, 137–39  
Chaos  
  attractors, 41, 52  
  and complexity science, 35  
  disproportionality, 38–39  
  edge of, 49–51  
  interdependent system feedback,  
  39–40  
  Lorenz’s simulated weather, 37–38,  
  54

- Chaos (*cont.*):  
 nonlinearity, 39–40  
 predictability and control, 14–15  
 sensitive dependence on initial conditions, 38  
 theory, 33–34  
*Chaos* (Gleick), 36  
 Christensen, Clayton, 124  
 Coevolution, 51–55  
 Collective emergent order, 53  
 Complex adaptive behavior, 62  
 Complex adaptive systems. *See* CAS (complex adaptive systems)  
 Complex Responsive Processes of relating, 99  
 Complexity. *See* Complexity science  
*Complexity and Management* (Stacey), 7  
 Complexity science. *See also* CAS (complex adaptive systems)  
 Branch Method, 60  
 CAS and complex problems, 42–43  
 complex adaptive behavior, 62  
 conflicting constraints, 44–45  
 continuous emergence, 63–64  
 explained, 34  
 interacting agents, 43  
 limited knowledge, 58  
 as a management fad, 141–42  
 muddling through, 59–60  
 and oversimplification, 95  
 perfect rationality, 57–58  
 possibilities and options, vast space of, 44, 58–59  
 Root method, 59–60  
 self-organization, 62–63  
 simple rules, 61, 62  
 stability, CAS *versus* equilibrium-based, 63–64  
 and thinking differently, 96–97  
 Complexity trap  
 avoiding, 97, 143–44  
 explained, 142–43  
 and managers, 144–45  
*Complexity* (Waldrop), 21–22  
 Conditioning, 14  
 Constantine, Larry, 62  
 Continuous emergence, 63–64  
 Control  
 Adaptive Development Lifecycle, 107  
 anxiety, reduction of, 14–15  
 both in and not in at once, 105, 106–07  
 and deviation, 107  
 giving up to gain, 106–07  
 hierarchy based on, 110  
 ISO-9000 family of standards, 29–31  
 and the manufacturing mindset, 3–4  
 novel change and emergence, 104–05  
 old-time, 103–04  
 predictability, 14–15, 27  
 in production process, 9, 28  
 unpredictable consequences of, 28–29  
 Control slack, 114  
 Conversations  
 birth of the project, 82–83  
 death of project, 90  
 “everybody” defined, 82  
 features, cutting and adding “just enough,” 86–89  
 feedback cycle, short, 84, 86  
 first step, figuring out, 84  
 and growing software, 70–71  
 kitchen sink thinking, 85–86  
 in the living present, 98  
 meetings as, 75–76  
 never-ending what/why/how, 84–85  
 project ideas and conception, 81–82  
 and project requirements, 78  
 Correlated landscapes, 52  
 Crichton, Michael, 33–34
- D**
- Darwin, Charles, 49, 54, 55  
 de Bono, Edward, 15  
 DeMarco, Tom, 24, 26, 62, 91, 113, 114, 128  
*Dilbert* (comic strip), 8, 162  
 Disproportionality, 38–39  
 Dynamic reality, 21
- E**
- Edge of chaos, 49–51  
 Efficiency optimization, 12–13, 14, 18, 19, 23–24  
 Emergent patterns, 41  
*Extreme Programming Explained* (Beck), 90, 150  
 Extreme Programming (XP). *See* XP (Extreme Programming)
- F**
- Failures, explanations for, 16  
 Feedback, interdependent system, 39–40

“Figure it out” step, 3  
 Fitness landscapes, 51–55  
 Fitness peaks, 52  
 Forward, Karen, 141n

## G

Gantt charts, 25, 123–24  
 Gleick, James, 36, 40  
 Grisham, John, 139  
 Grobstein, Paul, 77  
 Growing software. *See also* Complexity science; Conversations; Software development  
   and agile methods, 154–56, 157–59  
   conversation and participation, 70–71  
   designing as you go, 79–80  
   emergence, 69–70, 72–73  
   feeling our way, with conviction, 73–74  
   “just enough” plans, 76–78  
   manufacturing mindset, inadequacy of, 67–68  
   meetings, 75–76  
   minimalist software, 80, 86  
   movement and learning, 71–72  
   realism versus silver bullet, 146  
   requirements, 78–79  
   resistance to, 139  
   “right now” time horizon, 72  
   self-organization, 70  
   stopping, 73  
   success as an option, 145–46  
   top-down pressure and rigidity, 74–75  
   unpredictability, 72–73

## H

Hamel, Gary, 76, 106, 109, 134  
 Heisenberg, Werner von, 31  
 Highsmith, Jim, 3–4, 124, 147–48, 149, 155  
 Holland, John, 34  
 Hula-Hoop, 141n

## I

Industrial Age, 11, 33  
*The Innovator’s Dilemma* (Christensen), 124  
 Interacting agents, 43  
 Interdependent system feedback, 39–40  
 International Organization for Standardization (ISO), 29

Iran-Contra scandal, 16  
 ISO-9000 family of standards  
   good management practices, consensus on, 29  
   quality, defined, 30  
   Quality Management Principles, 30  
   software development (ISO-9003) and control, 30–31

## J

Jelled teams, 128  
*Jurassic Park* (Crichton), 34  
 “Just enough” plans, 76–78

## K

Kanigel, Robert, 17–18  
 Kant, Immanuel, 94  
 Kauffman, Stuart, 44, 45, 47–49, 50–51, 52–56, 57, 62, 96, 120, 133  
 Kitchen sink thinking, 85–86  
 Knerr, Richard, 141  
 Knowledge work, 24, 74

## L

Langton, Chris, 50  
 Laws of work, 12  
*Leading the Revolution* (Hamel), 109, 134  
 Lean Development (LD), 147  
 Leibniz, Gottfried Wilhelm, 13  
*Life Strategies* (McGraw), 13  
 Lincoln, Abraham, 44  
 Lindblom, Charles E., 59–60  
 Linear approximations, 40  
 Linear path plans, 6  
 Linear problems, 35–36  
 Lister, Tim, 62, 128  
 Living present, 97–99  
 Lorenz, Edward, 37

## M

Macro-temporal structures, 98  
 Management theories  
   “rule of thumb” management, 8–9, 60  
   Scientific Management, 9–10, 17–18  
   systems thinking, 7–8  
 Managers  
   and change, 91–92

- Managers (*cont.*):  
 as change catalysts, 111, 132, 133–34  
 conditioning of, 135  
 as distraction filters, 128  
 emergence, inability to control, 72–73, 95, 104–05  
 as glue in an organization, 128  
 modeling better behavior, 135  
 pacing the team, 128  
 as participants, 110–11  
 planning, as victims of, 4  
 predictability and control, 14–15, 27, 105–06  
 and the project plan, 26–27  
 roles and responsibilities, Taylor’s view of, 9  
 systems thinking, 7–8, 95, 97  
 unrealistic expectations, subjection to, 161
- Manufacturing mindset. *See also* Taylorism  
 assumptions of, 19–20, 25  
 automobiles and software, compared, 4–6  
 birth of, 1  
 creativity and quality, 4–5  
 defined, 1  
 as dominant project approach, 2  
 “figure it out” step, 3  
 management fads based on, 24  
 planning and control, 3–4, 16  
 predictability and control, 14–15  
 process, explained, 4–5  
 project failure rates, 4  
 rewards and payoffs of, 14  
 scope creep, 3–4  
 as simplifying assumption, 23  
 software development, imposition on, 5–6, 10–11, 18–19, 23
- Manufacturing process, 4–5  
 McDonald’s, 8  
 McGraw, Phil, 13  
 Melin, Arthur “Spud”, 141  
 Messy problems, 22–23, 33  
 Microsoft Project, 5  
 Micro-temporal structures, 98  
 Midvale Steel Company, 8  
 Minimalist software, 80  
 Morgan, J. P., 58, 120
- N**
- Natural selection, 49–50, 52–53
- Newton, Sir Isaac, 36  
 Newtonian science  
 assumption of predictability, 36–37  
 linearity, 35–36, 37  
 Nonlinearity, 39–40
- O**
- O Brother, Where Art Thou?* (film), 106  
 One Best Way, 9–10, 12, 13, 21  
*The One Best Way* (Kanigel), 17–18
- P**
- Pair programming, 152  
*The Paradox of Control in Organizations* (Streatfield), 15  
 Pavlov, Ivan, 14  
*Peopleware* (DeMarco and Lister), 62–63  
 Perpetual novelty, 42  
 Phase transition, 49, 51  
 Phased plans, 5–6  
 Planning. *See also* Project plans  
 future, predictability of, 20–21  
 linear paths, 6  
 managers, as victims of, 3  
 and the manufacturing mindset, 3–6  
 phases, 5–6  
 plausible deniability, 16  
 and safety, 14–15, 16  
 scope creep, 3–4  
 up-front analysis, excessive, 76
- Plausible deniability, 16  
 Power relations, 114  
 Predictability  
 assumption of, 36–37  
 conditioned need for, 14–15  
 and control, 27  
 of future, 20–21  
 and Taylor’s problem, 13  
*The Principles of Scientific Management* (Taylor), 17
- Problems  
 efficiency optimization, 12–13, 14, 19, 23–24  
 linear, 35–36  
 messy, 22–23, 33. *See also* Complexity science  
 reductionist approach to solving, 10–11  
 Rubik’s Cube, 11–12  
 simple versus complicated, 11–12  
 simplifying assumptions, 22–23

- software development *versus* Taylorism, 21–22
  - Taylor's, characteristics of, 12–13, 21
  - Productivity, maximizing, 9–10
  - Progressively less wrong, 77
  - Project failure rates, 4
  - Project plans. *See also* Planning
    - deliver frequently, 138
    - deviation and failure, 124–25
    - failures, as explanation for, 16
    - fear of being wrong, 76
    - honesty, need for, 138–39
    - making reality conform to, 76
    - and predictability assumptions, 37
    - pressure, using to apply, 26
    - progressively less wrong, 77
    - as schedules, 26
    - small and simple, 77
    - thinking ahead, 78
    - translating, 137–38
    - unrealistic, 26
  - Projects, finishing
    - being helpful, 129
    - good enough programmers, 130–31
    - nudging, 131
    - pressure and control, 132
    - programmers, relating to, 130
    - responsibilities of managers, 127–28
  - Projects, starting
    - break even point for services, 122
    - experience, importance of, 121
    - features, attaching a dollar value to, 117–18
    - goals versus detailed plans, 121
    - level of detail and just enough information, 119–21
    - money, asking for, 118
    - plans, 123–25
- Q**
- Quality Management Principles (ISO-9000 standards), 30
- R**
- The Rainmaker* (Grisham), 139
  - Reagan, Ronald, 16
  - Redi, Francesco, 48
  - Reductionist approach, 10–11
  - Reification, 134–35
  - Requirements documents, 78
  - Reynolds, Craig, 61
  - Rifkin, Jeremy, 10
  - Rittel, Horst, 22n
  - Roosevelt, Teddy, 156
  - Root Method, 59–60
  - Rubik's Cube, 11–12
  - "Rule of thumb" management, 8–9, 60
  - Rules of thumb, 59, 60, 64
- S**
- Safety, 14–15, 16
  - Schedules, 25–27
  - The "Science" of Muddling Through* (Lindblom), 59–60
  - Scientific Management, 9–10, 12–13, 17–18. *See also* Taylorism
  - Scope creep, 3–4
  - Scrum, 147
  - Self-organization, 47–49, 52–53, 62–63
  - Self-organizing criticality, 54–55, 64, 120
  - Sensitive dependence on initial conditions, 38
  - Simplicity* (de Bono), 15
  - Simplifying assumptions, 23, 95
  - Six Sigma, 28
  - Slack* (DeMarco), 24, 26, 91
  - Smith, Adam, 43, 62
  - Software development. *See also* Complexity science; Growing software assembly line mentality, 6
    - control, conditioned need for, 14–15
    - doing what works, 13–14
    - as a dynamic reality, 21
    - as exploration, 32
    - iterative process of refinement, 60
    - as knowledge work, 24, 74
    - linear approximations, simplifying with, 40
    - and the manufacturing mindset, 5–6, 14, 17, 19–20, 67–68
    - manufacturing model, logic behind using, 18–19
    - maturing of, 1
    - realistic, summarized, 64–65
    - reductionist approach, application of, 10–11
    - schedules, 26
    - and simplifying assumptions, 22–23
    - standard ISO-9003, 30–31
    - traditional thinking about, 95
    - unpredictability of, 20–21, 28–29
  - Software Development Uncertainty Principle, 31–32

Software, growing. *See* Growing software  
 Spontaneous generation, 48  
 Stacey, Ralph D., 7, 12–13, 61, 62, 63,  
 64, 95, 96, 97, 98, 99, 105–06, 109,  
 110, 112, 114, 142, 143  
 Streatfield, Philip, 15, 97, 103, 105, 135,  
 143  
 Successive limited comparisons, 60  
 Systems thinking, 7–8, 95, 97

### T

Taylor, Frederick W., 8–10, 12–13, 17,  
 21, 24, 40, 60  
 Taylorism. *See also* Manufacturing mind-  
 set  
   American thinking, integration into,  
   17–18  
   and dehumanization, 10  
   inefficiency, causes of, 8  
   laws of work, 12  
   One Best Way, 9–10, 12, 13, 21  
   problems, characteristics of, 12–13, 21  
   productivity, maximizing, 9  
   reductionist approach, use of, 10  
   “rule of thumb” management, 8–9  
   time studies, 10  
   and workers, 9  
 Test-first development, 152  
 Thinking differently  
   Complex Responsive Processes of re-  
   lating, 99  
   complexity science, the challenge of,  
   96–97  
   and getting stuck, 100–101  
   interactive relating, 99–100  
   living present, 97–99  
   macro- and micro-temporal structures,  
   98  
   oversimplification, 95  
   pretending, 101–02  
   realistic thinking, 97, 99  
   systems thinking, 7–8, 95, 97

  thought lock-in, 93–94  
 Thought lock-in, 93–94  
 Time studies, 10  
 Toy weather, 37–38, 54  
 Transformative Teleology, 97  
 Turing, Alan, 21

### U

Uncertainty, managing, 112  
   general purpose, defining initial,  
   111–12  
   managers, as participants, 110–11  
   org charts versus boundaries, 113  
   personal risk, 114–15  
   power, sharing of, 114  
   principles, contributing initial, 112  
 Uncertainty Principle (Heisenberg), 31  
*Undecidability theorem* (Godel), 21

### W

Waldrop, Mitchell, 21–22  
 Waterfall process, 19  
 Watson, John, 14  
 Webber, Melvin, 22n  
 WHAM-O toy company, 141  
 Wicked problems, 22n  
 Workers, 9

### X

XP (Extreme Programming)  
   defined, 150  
   one team, 153–54  
   practices, 150–51, 152–53  
   testing, customer and programmer,  
   151  
   values, 150

### Y

Yorke, James, 22–23