

# Index

## A

A3 reports, 113–114  
adaptive action, 180  
agile development style: adaptive action to correct plans in, 179–180; automating building and testing activities, 212; backlog vs. project completion in, 180; barely sufficient processes in, 209; coping with change in, 5; Extreme Programming, 14; feature replacement in, 33; loose-tight planning with iterations, 188–189; methodologies for, 14; pairing and, 144; process standardization with, 207; refactoring and technical debt, 226, 228; response to change vs. following plans, 179; robustness and agile concepts, 79; team meetings for, 135–136; using for other products, 25–26, 29; values of, 231–232, “you aren’t going to need it”, 228–230. *See also* Agile Manifesto; XP  
Agile Manifesto: design simplicity and, 16, 22, 26; development processes in, 205; emphasis on individual in, 182; feedback in XP practices and, 23; individuals acknowledged in, 127, 152; methods outlined in, 13, 14; response to change vs. following plans, 179; text of, 12; value of working products, 211  
Agile Project Leadership Network, 177  
*Agile Project Management* (Highsmith), 177, 190  
*Agile Retrospectives* (Derby and Larsen), 201  
analysis-think-change, 240  
anticipating customer needs, 47–52;

anticipating change, 5–6; forecasting vs., 47; immersing developers in customer environment for, 48–51, 55; project planning vs., 185–186

Apple Computer, 50

assessing explorations, 91, 93–94

authority and teams, 139–141, 152

automobile design: body panel decisions in, 119; delaying decisions on, 118–120; using modular architecture, 65–66

Avery, Christopher, 138

## B

backlog in agile development, 180, 188

Beck, Kent, 20, 23

Berra, Yogi, 107, 114, 177, 178

best practices: heavy process and, 208; management judgment and, 26–27; product development and, 13; project difficulties following, 129–130

Black & Decker, 66–67, 73, 170–171

Black-Scholes formula, 174

blogs, 49–50

BMW, 41

bodies of knowledge approach, 129–130

Boehm, Barry, 8–10, 24, 35–36, 125–127, 131, 215–221

Boeing, 189

bottlenecks and queues, 221–226;

identifying bottlenecks, 222–223;

impeding flexibility with queues, 226,

230; theory of constraints methodology, 222; wait times and myth of

capacity, 223–226

bottom-up change, 244–248; combining

with top-down, 245; top-down vs.,

233–234

Box, George, 163  
 Bridges, William, 237–238, 245  
 Brooks, Frederick, 231  
*Building the Bridge As You Walk On It*, 127  
 building vs. scaling down processes, 207–209, 230  
 burndown charts, 198–199  
 burnout, 18

## C

calculus, 110  
 call options, 173  
 Callahan, Jim, 150  
 Canadian pharmacy, 249  
 cascading change, 117  
 C-Cor, 150  
 CD-ROM drive architecture, 67–72, 80–81  
 change: accommodating predictable, 67–73, 84; agile skills coping with, 5; anticipating, 5–6; cascading, 117; cost of, 8–10; dealing with product, 2–7; engineering, 107, 119; experience dealing with, 129; feeling at core of successful, 239–240; guiding via teams, 241; higher level requirements reducing, 40–47, 55; identifying bottlenecks with proposed, 223; individual's comfort with, 128; influences requiring project, 4–5; innovation and, 2, 29; institutionalizing, 244; intrinsic risk management with rapid, 192; isolating volatility, 74; maintaining continuous, 243; managing projects and, 177–178; providing for product growth, 73, 80; top-down, 239–244; transition vs., 237; urgency leading to, 240; user stories keeping cost low for, 46–47; using flexibility tools offensively, 249. *See also* cost of change; organizational change  
 chicken test, 105  
 Chrysler, 20, 43, 119  
 chunk architectural elements, 57, 58–59, 60  
 clustering product elements, 76  
 Cockburn, Alistair: 144, 207, 212. *See also* mastery levels for team members  
 COCOMO (Constructive Cost Model), 125  
 Cohn, Mike, 156–157  
 collective code ownership: coding

standards and, 19–20; defined, 18; safety net for, 21; translating practice to other products, 26  
 co-located teams: communication patterns for, 148–150; dispersed teams and, 141, 142; meeting together initially, 147; partially, 146–150; productivity and, 143–144; working with, 14, 141–146  
 communications: analyzing co-located team, 148–150; cascading changes in corporate, 117; communicating organizational change, 241–242; cultural factors and, 22; managing team's electronic, 150–151, 152; as XP value, 20  
 computers. *See* desktop computers  
 conducting retrospectives, 202–203  
 consensus in group decisions, 159–161  
 Constantine, Larry, 57  
 constraints: set-based design, 109–113; Toyota constraints checklists, 112  
 constructing exploration models, 91, 92–93  
 Constructive Cost Model (COCOMO), 125  
 continuous integration, 18, 21, 26  
 convergence, 10–11, 116  
 cooked meat analogy, 218–221  
 cordless screwdrivers, 66–67, 170–171  
 corporate culture. *See* cultural factors  
 cost of change: factors influencing labor costs, 125–127; front-loaded prototyping and, 95; growing during project, 8–10; in hardware vs. software, 65, 83, 211; lowering with delayed decisions, 119; modular architecture and, 64, 67; psychological, 123; set-based design and, 114–115; simple design and, 16, 17, 22, 228–229  
 costs of iterations: economics of prototyping, 101; lowering, 39–40; 3-D printer prototyping and, 97–100  
 courage, 21, 23  
 Credit Suisse, 49  
 critical path, 156  
 critical success factors, 2  
 cross-functional teams, 138–139; product architecture and, 73; product requirements and, 34  
 cultural factors, 232; communication and, 22; corporate cultures developing Level Xers, 131; corporate discouragement

- of failure, 90; impeding flexible development, 38; inertia and organizational change, 96; set-based design and Western mindset, 115
- customer iteration cycle. *See* iterations
- customers: anticipating needs of, 5–6, 47–52, 55; changes in product requirements by, 4; combining iterations with feedback from, 35–36, 37–38, 55; completion measured by feedback from, 180–181; developing product requirements from needs, 31; having on team, 26; IKIWISI, 4, 31; immersing developers in customers' environment, 48–51, 55; incremental releases for, 209–210; internal, 54; lead user research, 51–52; misleading feedback from expert, 53; pitfalls of feedback from, 52–54, 55; users vs., 254; value of feedback in product development, 34–40; value of incremental development, 210; vision statements developed from surveys of, 42; VOC research with, 53–54
- D**
- Darwin, Charles, 1
- DeCarlo, Doug, 177
- decision gradients, 159–160
- decision making, 153–176; choices for improving, 153–154; critical path in, 156; cross-functional, 138–139; decision gradients, 159–160; deferring, 108; finding linked decisions, 171–173; imperfect nature of, 161; last responsible moment, 154–157, 175; mastery levels and quality of, 129–131; point-based vs. set-based, 109–111; preventing premature, 115, 118–122; procrastination vs. last responsible moment, 157–158; progressive, 120–122; pushing decisions down to increase response time, 194; real options thinking, 173–175; set-based design and better, 115; team structure and, 158–161; timeboxes and, 190; uncertainty and, 161–162, 165, 167, 192, 204. *See also* decision trees; delaying decisions
- decision tree software, 170–171
- decision trees, 162–173; about, 162–164, 176; benefits of, 172–173, 176; employing, 164–169; finding linked decisions in, 171–173; flipping, 170–171; illustrated, 163, 166, 167, 168; value of perfect information in, 169–171
- delaying decisions: accommodating change in cordless screwdriver, 67; better information in set-based design with, 115; making last responsible moment decisions, 154–157, 175; photocopiers as example of, 120, 121; procrastination vs., 157–158; set-based design and deferring decisions, 108; Toyota's style of, 118–122
- Denso, 66
- Derby, Esther, 201, 203
- Design of Experiments methodology, 101
- design simplicity, 16, 22, 26
- desktop computers: CD-ROM drive development for, 67–72, 80–81; designing interface of, 79; drawing architectural schematics of, 75–76; improvements in prototyping using technology of, 86; modular architecture in, 64–65
- Developing Products in Half the Time* (Smith and Reinertsen), ix, 121
- development processes, 205–230; balancing anticipation and adaptation, 212–213; bottlenecks and queues, 221–226; build rather than scale down, 207–209; emergent, 205–209; flexible processes basics, 209–215; low-level standardization, 206–207, 230; managing change with YAGNI, 228–230; refactoring and technical debt, 226, 228; risk balancing in, 215–221, 230; shifts from flexible to structured, 221, 222, 230; software methodology research on, 208–209; structure balanced with flexibility, 215–221; tacit knowledge, 213–215, 230; wait times and myth of capacity, 223–226
- development teams. *See* teams
- disclaimers, 26–27
- dispersed teams: about, 141, 142; changes due to, 5; complications running, 146–147
- disruption: flexibility and lack of, 2; systemwide impact of change and, 8, 9
- Duarte, Deborah, 146, 150

## E

- Edison, Thomas, 85, 87
- e-mail communication standards, 150–151, 152
- emergence: process and, 127, 205–206; requirements and, 4, 38
- employee blogs, 49–50
- empowering teams, 234, 243
- Evangelists, 246
- examples. *See* product scenarios
- exhaust systems, 118
- expectations management, 190–191
- experimentation, 85–106; corporate discouragement of failure, 90; defined, 85, 86–87; exploration as, 90–94; failure in, 87–90; flexibility and, 85–86; mistakes vs. failures, 88; observing experiments, 91, 93; prototyping and, 94–104; solving uncertainty in decision making with, 162; technologies enabling prototyping, 97–98, 99; testing prototypes, 104–106
- expert customer feedback, 53
- explicit knowledge, 182, 213–214
- exploration: assessment step for, 91, 93–94; constructing models in, 91, 92–93; defined, 90–92; iterative process in, 91; managing risks with, 194; planning for, 91, 92; run step for, 91, 93
- Extreme Programming. *See* XP
- eXtreme Project Management* (DeCarlo), 177
- F
- failure: corporate discouragement of, 90; experimentation, 87–90; finding bigger problems ahead from, 193; learning from prototyping, 89; mistakes vs., 88; testing for product, 104–106
- FDA (U.S. Food and Drug Administration), 32
- feedback, 34–40; combining iterations with customers', 35–36, 37–38, 55; measuring product completion by, 180–181; pitfalls of expert customer, 52–54, 55; XP and, 21, 23
- feelings and change, 239–240
- Ferrari F430 sports car, 62, 63
- final testing, 106
- flexibility: applying selectively, 9; benefits of, 7–8; calculus' limiting approach on, 110; convergence and, 10–11, 116; cultural factors impeding, 38; decision trees and, 162–163; definitions of, 1–2, 196–198; difficulties implementing set-based, 122–124; discretionary use of, 29; downside of, 11–12, 114–115; enhancing with experimentation, 85–86; excessive, 10–11; importance of teams in, 127–128; lack of disruption and, 2; last responsible moment and, 154–157, 175; levels of project, 10–11; managing unknown risks and, 193–195; modular architecture and, 58–59, 63–64; organizational changes required for, 90, 96; paradoxes in implementing, 233–237, 248; postponement and, 59; project management as key in, 177–178; reducing costs of with set-based design, 114–115; strengthening tacit knowledge for, 182; XP practices and, 14. *See also* flexible processes; implementing flexibility; XP
- Flexibility Index, 196–198
- flexible processes, 205–215; about, 205–206; balancing risk in, 215–221, 230; emergence of, 205–209, 230; iterative and incremental innovation, 209–212, 230; low-level standardization in, 206–207, 230; queues as impediment to, 226; scaling down processes, 207–209, 230; shifting to structured processes, 221, 222, 230
- flexible project management: institutionalizing, 244; mainstream vs., 178, 203; manager's role in, 182–189; project plans in, 178–180; rolling-wave planning, 186–188
- flipping decision trees, 170–171
- following plans: allegiance to, 2, 178–180; response to change vs., 179; rewards for, 181
- forecasting, 6, 47
- Frameworks LLC, 155
- front-loaded prototyping: guidelines for, 100–104; traditional vs., 95–96; using for surgical laser, 98–100
- functional architectural elements, 57, 58–59, 60

**G**

Gates, Bill, 31  
 generalists, 136–137  
 Generation X lifestyle, 50  
 Google, 50  
 Griffiths, Mike, 147  
 Grove, Andrew, 194–195

**H**

heating pad decisions, 167–169  
 Hewlett, Bill, 183, 240  
 Hewlett-Packard, 41–42, 195–196, 240, 241, 242  
 Highsmith, Jim, 54, 177, 190  
 Hohmann, Luke, 48

**I**

Iansiti, Marco, 91  
 IBM, 88  
 IKIWISI (“I’ll know it when I see it”), 4, 31  
 IMNIL (“I might need it later”), 229  
 imperfect decision making, 161  
 implementing flexibility, 231–249;  
   bottom-up change, 233–234, 244–248; building vision of change, 241; celebrating successes in, 243; empowerment and, 234, 243; exposing or sheltering projects, 237; gradual or ambitious goals for, 236–237; guiding change via teams, 241; paradoxes in, 233–237, 248; starting small or big, 234–235; starting with piece or whole package, 235–236; summary, 248; top-down change, 233–234, 239–244; transitions in, 237–239. *See also* flexible project management  
 improving decision making, 153–154  
 incremental innovation, 209–212, 230  
 indecision and flexibility, 11–12  
 inefficiencies in over-dedication, 133–135  
 information technology (IT): 213–214, 215; customers and, 19, 180  
 innovation: areas where flexibility can aid, 7–8; change required for, 2, 29; decline in, 2–3; iterative and incremental, 209–212, 230. *See also* product development  
 integral product architecture: defined, 59; modular vs., 58–59, 63; performance

advantages, 60–61, 63; shifting design from modular to, 61. *See also* modular product architecture  
 integrated risk management, 191–192  
 Intel, 131, 194  
 interaction of product modules, 80  
 internal customers, 54  
 intrinsic risk management, 191–192  
 ISO 9000, 3  
 iterations: combining with customer feedback, 35–36, 37–38, 55; exploration and iterative process, 91; feature replacements during, 33–34; having working product at end of, 14; incremental innovation processes and, 209–212, 230; iterative organizational change, 234, 235; loose-tight planning and, 188–189; lowering costs of, 39–40; managing risks with, 194; measuring completion in, 180–181; retrospectives for, 202; risk management and, 192–193; story cards for, 46; velocity of, 189. *See also* costs of iterations

**J**

Jaguar, 41  
 jeepney, 61–63  
 Johnson & Johnson Worldwide Emerging Markets Innovation Center, 26

**K**

Kerth, Norman, 201  
 Kessler, Robert, 144–145, 146  
 keypad, 81–82  
 knowledge management, 213  
 Kotter, John, 234, 239–243, 245, 248

**L**

language layers, 206–207  
 Larman, Craig, 12  
 Larsen, Diana, 201, 203  
 last responsible moment: defined, 155; procrastination vs., 157–158; using, 154–158, 175  
 lead users, 51–52, 53  
 leadership for implementing flexibility, 233–234  
 leaf nodes in decision trees, 164  
 Level X developers, 130, 131, 145, 159

levels: project flexibility, 10–11. *See also* mastery levels  
 Lexus, 41, 42–43  
 Liker, Jeffrey, 200  
 linked decisions, 171–173  
 lock analogy, 102–104  
 loose-tight planning, 188–189  
 low-level standardization, 206–207

## M

M/M/1/∞ queue, 224, 225, 226  
 MacCormack, Alan, 34–36, 91, 128–129, 181, 215, 220  
 mainstream project management: about, 177; allegiance to project plans, 178–180; emphasis on processes in, 182; flexible vs., 178, 203; views of project completion, 180–181  
 management by walking around (MBWA), 183–184  
*Managing the Design Factory* (Donald Reinertsen), 140  
 Manns, Mary Lynn, 245, 247–248  
 manufacturing, 1  
 market changes, 4  
 Marshall, Jon, 155  
 mass customization, 1, 59  
 mastery levels for team members: Cockburn, 129–130, 131–132, 206, 208, 209, 214. *See also* Level X developers  
 McQuillen, David, 49  
 Mercedes Benz, 41, 42  
 methodologies: agile software development, 14; Design of Experiments, 101; development processes in software, 208–209; Pugh concept selection, 112; theory of constraints, 222. *See also* agile development style; Agile Manifesto  
 metrics: burndown charts, 198–199; Flexibility Index, 196–198; sharing and acting on, 200; strategic vs. tactical, 195–196; team mood as, 199–200  
 Microsoft, 49, 73  
 mindfulness, 194  
 mistakes: exploration vs., 90; failures vs., 88  
 modular product architecture, 57–84; aligning organization boundaries with, 74–75; automobile design using, 62, 65–66; CD-ROM drive, 67–72, 80–81; cordless screwdrivers, 66–67;

decisions in approaches to, 72–74; defined, 58–59; designing interface, 79; desktop computers' use of, 64–65; isolating volatility, 73; objectives for, 63–64; placing functionality in modules, 78; providing for product growth, 73, 81; reducing coupling, 73–74; shifting boundaries in architectural domains, 83; steps in designing, 74–78. *See also* integral product architecture  
 money-for-adaptability (MFA), 212, 213  
 money-for-information (MFI), 212, 213  
 Morgan, James, 200  
 Mountain Goat Software, 156–157  
 mules, 40, 93

## N

NASA (National Aeronautics and Space Administration), 218  
 needle-in-a-haystack metaphor, 6

## O

OEM (original equipment manufacturer), 169, 219  
 Olson, Gary, 143, 146  
 Olson, Judith, 143, 146  
*Only the Paranoid Survive* (Grove), 194–195  
 organizational change: bottom-up, 233–234, 244–248; building vision of, 241; celebrating successes, 243; combining bottom-up and top-down, 245; communicating and building ownership in, 241–242; continuing, 243; empowerment and, 234, 243; exposing or sheltering projects, 237; gradual or ambitious goals for, 236–237; guiding via a team, 241; leadership required for, 233–234; overview, 232–233; patterns for, 246–248; risks in iterative, 235; starting small or big, 234–235; starting with piece or whole package, 235–236; top-down, 233–234, 239–244, 245. *See also* analysis-think-change, see-feel-change  
 Orion, 98–100

## P

Packard, Dave, 183  
 pair programming. *See* pairing

- pairing, 17–18; agile software development and, 144; coding standards, 19–20; continuous integration with, 18, 21, 26; non-software example, 145–146; origins of, 20; safety net for collective code ownership, 21; translating practice to other products, 26; types of, 145
- parallel development: parallel prototyping, 102–104; set-based design vs., 112–113
- Pareto principle, 8
- patterns: defined, 245; organizational change using, 246–248
- people factors: commitment to team, 135–136; cultivating specific areas of depth, 137; dedication of time, 133–135; desirable qualities for teams, 132–137; establishing paths for mastery, 132; evaluating mastery levels, 129–131; importance of feelings in change, 239–240; individual's comfort with change, 128; influence on development costs, 125–126, 132; organizational change and, 237–238; phases in transition process, 235; technical and social skills, 132–133; types of pairing by, 145; useful experiences, 128–129. *See also* cultural factors
- personal computers. *See* desktop computers
- personas, 44
- Peters, Tom, 105
- phased development: x, 3–4, 5–6, 10; decisions and, 153, 154–155; documentation heavy, 32, 211; iteration and, 193, 209; layers of development process and, 207
- photocopiers, 120, 121
- pig, chicken, and cow analogy, 135–136
- planning: anticipation vs., 185–186; exploration process, 91, 92; loose-tight, 188–189; need for 38; rolling-wave, 186–188; XP project, 15. *See also* project plans
- platform architecture, 59
- platforms, 1
- point-based design, 108–110, 111, 113
- postpartums/postmortems, *See* retrospectives
- postponement, 59
- printers; 48; 3-D, 97–98, 99; Hewlett-Packard DeskJet, 41–42, 241, 242
- problem-solving, 129, 244
- processes. *See* development processes
- procrastination: anticipation vs., 186; last responsible moment vs., 157–158
- product architecture: attention to, 34; defining, 57–58; design-level changes in, 81–83; modular vs. integral, 58–63. *See also* integral product architecture; modular product architecture
- product audit, 201–202
- product development: creating working products early in cycle, 34; customer feedback in, 34–40; decision making objectives and methods for, 153–154; designing to specification, 31–32; factors influencing costs of, 125–127; implementing flexibility in small or big projects, 234–235; loose-tight planning with iterations, 188–189; overspecification trap in, 36–37; prototyping and successful, 35–36; requirements creep, 32–34; specifying at higher level, 40–47, 55. *See also* innovation; project management; specifications
- product elements: chunks and functional, 57, 58–59, 60; clustering, 76
- product vision: project manager's preservation of, 184; statements of, 41–44
- productivity and co-location, 17–18; 143–144
- products: accommodating predictable change for, 67–73, 84; changes to improve reliability, 240–241; designing interface for, 79; drawing architectural schematics of, 74–76; emergence of requirements, 4, 38, 127; isolating volatility in, 74; metaphor used in XP, 16; objectives for modular, 63–64; pitfalls of customer feedback on, 52–54, 55; placing functionality in modules, 78; project personas for, 44–45; providing for growth of, 73–74, 81; reorienting quality of, 181–182; unused features in, 37; vision statements for, 41–44; working, 13, 30, 34–35, 188, 209, 211. *See also* modular product architecture
- progressive decision making, 120–122
- Project Analyzer, 27–29
- project management, 177–204; aligning organization with product architecture, 75; allegiance to project plans,

- 178–180; applying to innovative projects, 4; authority of teams, 139–141; burndown charts, 198–199; Flexibility Index, 196–198; flexible vs. mainstream, 178, 203; manager's role in flexible, 182–184; managing set-based design, 115–117; measuring team mood, 199–200; planning vs. anticipation, 185–186; project completion, 38; reorienting product quality, 181–182; risk management, 191–195; sharing and acting on metrics, 200; shifting emphasis from processes to individuals, 182; strategic vs. tactical metrics for, 195–196; tacit vs. explicit knowledge, 182, 213–214; timeboxing, 190–191; using retrospectives in, 200–203; working with Level Xers, 131. *See also* planning; project managers; project plans
- Project Management Institute, 177, 191, 251
- project management software, 179
- project managers: managing by walking around, 183–184; preserving product vision, 184; role in flexible project management, 182–184; supporting and shielding team members, 184
- project plans: allegiance to, 2, 178–180; measuring success and rewards by, 181; response to change vs., 179
- Project Retrospectives* (Kerth), 201
- project risk management, 191–195; about, 191; integrated vs. intrinsic, 191–192; iterative development and, 192–193; managing unknown risks, 193–195
- projects: establishing thumbprints for, 27–29; evaluating value of undertaking, 174; exposing or sheltering transitional, 237; Flexibility Index for, 196–198; maintaining sustainable pace, 18; process shift over life of, 221, 222, 230; tracking scope with burndown charts, 199; views of project completion, 180–181
- prototyping: defined, 94; delaying decisions during automotive, 118–120; economics of, 101; experimentation and, 94–104; learning from failures in, 89; lowering costs of, 39–40; MacCommack's findings on, 34–35; parallel vs. sequential, 102–104; refinement of, 100–101; technologies enabling, 97–98, 99; traditional vs. front-loaded, 95–96; user feedback and, 35–36
- pruning, 115–118
- Pugh concept selection method, 112
- put options, 173
- ## Q
- Quadrus Development, 147, 249
- queues: impeding flexibility with, 226, 230; M/M/1/∞ model for, 224, 225, 226; wait times in, 224, 225, 226, 227
- queuing theory, 224
- ## R
- radical collocation, 143
- reading decision trees, 164
- real options thinking, 173–175
- reducing coupling, 73–74
- refactoring: about, 17, 226, 228; arresting architectural decay and, 80; safety net for collective code ownership, 21
- refinement of prototypes, 100–101
- Reinertsen, Donald, 32, 34, 121, 140, 142
- reliability of products, 240–241
- requirements: emergent, 4, 38, 127; requirements creep, 32–34. *See also* specifications
- retrospectives, 200–203; benefits of, 200–201; conducting, 202–203; iteration, 202; postmortems and, 201; project, 202
- rice cooker use case, 45–47
- Rising, Linda, 245, 247–248
- risks: balancing opposing, 215–221, 230; decision making triggered by, 156; found in iterative organizational change, 235; loose-tight planning and managing, 189; organizational change and, 234. *See also* project risk management
- robustness, 79
- rolling-wave planning, 186–188
- Rolls-Royce, 105
- ## S
- scaffolding, 40, 93, 105
- scaling down processes, 207–209, 230
- schematics, 57–58, 70–71, 75–76
- Schrader tire valve, 80

- scope creep, 32–34  
 see-feel-change, 240, 241, 248  
 selective flexibility, 9  
 self-organizing teams, 25, 138  
 sequential prototyping, 102–104  
 set-based design, 107–124; benefits of, 114–115; constraints basis, 109–113; defined, 108; delaying decisions for better information, 115, 118–122; difficulties implementing, 122–124; point-based vs., 108–110; problem solving with, 111–113; pruning, 115–118; technical reports supporting, 113–114  
 single-minute exchange of die (SMED), 1  
 Six Sigma, 3, 5  
 sketching design layout, 76–78  
 skills: cultivating specific areas of depth, 137; establishing paths for gaining mastery and, 132; evaluating mastery levels, 129–131; specialists vs. generalists, 136–137; technical and social, 132–133; types of pairing by, 145  
 small releases, 15–16, 26  
 Smith, Preston G., 121, 275–276  
 Snyder, Nancy, 146, 150  
 social skills, 132–133  
 software, special characteristics of, 25  
 Sony Walkman, 60, 61  
 specifications: avoiding change with higher level, 40–47, 55; designing product to, 31–32; development using prototyping vs., 34–36; excessive, 36–37; frozen, 31–32, 54–55; vs. requirements, 253  
 Spinrad, Robert, 153, 154  
 Standish Group, 37  
 stand-up meetings: agilists use of, 135–136; formats for, 148–150  
 stereolithography (SL), 97  
 story cards for iterations, 46  
 Stage-Gate. *See* phased development  
 strategic project metrics, 195–196  
 structured development processes: risks in, 216, 217; shifting from flexible to, 221, 222, 230  
 stubs, 93, 105  
 success: celebrating implementation, 243; commercial success vs. manager's view, 34–35; feelings at core of successful change, 239–240; improving chance of in implementation, 235; innovation and critical success factors, 2; measuring by project plan, 181; prototyping products, 35–36  
 surgical drapes, 52  
 surgical laser, 98–100  
 Suzuki, Ichiro, 42  
 synergy in whole-package transitions, 235–236
- ## T
- tacit knowledge, 182, 213–215, 230  
 tactical project metrics, 195–196  
 TEAC, 67–71  
 team leaders. *See* project managers  
 teams, 125–152; authority of, 139–141, 152; changes facing dispersed, 5; co-located, 14, 141–146; commitment and pig, chicken, and cow analogy, 135–136; creating great, 131–132; cross-functional, 138–139; decision making within, 158–161; dedicating people to projects, 133–135; desirable human qualities on, 132–137; difficulties with Level X members, 130, 131, 145, 152; electronic communications among, 150–151, 152; empowering to lead change, 234, 243; experience in product delivery, 34–35; factors influencing development costs, 125–127; guiding change via, 241; having customer on, 18–19; importance in flexibility, 127–128; managing by walking around, 183–184; measuring mood of, 199–200; meetings for agilists, 135–136, 148–150; partially co-located, 146–150; roles and responsibilities of members, 138; selecting people on, 128–132; self-organizing, 138; sharing metrics with, 200; specialists vs. generalists on, 136–137; subteams for set-based design, 123; supporting and shielding members of, 184; sustaining project pace, 18; training in flexible development techniques, 242. *See also* co-located teams; dispersed teams  
*Teamwork Is an Individual Skill* (Avery), 138  
 technical debt, 228  
 technology: change stemming from, 4; changes in, 5; choosing with decision trees, 163–164; enabling shifting

boundaries in architectural designs, 83; prototyping enabled by, 97–98, 99

test-driven design, 16–17, 21, 26

testing: deferring for iterative development, 211–212; front-loaded vs. final product, 106; as experimentation, 104–106; value of failure in, 87–90

theory of constraints, 222

Thomke, Stefan, 88, 100

3-D printers, 97–98, 99

3M surgical drapes, 52

thumbprints for projects, 27–29

timeboxing, 190–191

tolerances, 119–120

tool kit metaphor, xi, 232

tool-safe dies, 122

top-down change: bottom-up vs., 233–234; combining with bottom-up, 245; Kotter's steps for, 239–244

tortillas, 48

Toyota: constraint checklists, 112; design of interfaces at, 79; immersing developers in U.S. environment, 50; isolating non-changing areas, 65–66; lack of employee complacency at, 243; managing rate of convergence, 116; preventing premature decisions, 122, 123; product vision statement at, 43; retrospectives at, 200; SMED technique at, 1; standardizing low-level activities at, 207; tradeoffs described in A3 reports, 113–114; use of set-based design, 107; use of tacit knowledge, 214, 215

trade-off curves, 112

training, 242

transitions, 237–239

T-shaped individuals, 137

Turner, Richard, 24, 125, 131, 215

## U

uncertainty: determining sensitivity of decisions to, 165, 167; intrinsic risk management and, 192, 204; reducing decision making, 161–162

underlying XP values, 20–23

U.S. Army jeep, 61, 62, 63

U.S. Food and Drug Administration, 32

U.S. National Aeronautics and Space Administration (NASA), 218

unk unk (unknown unknown) risks, 193–195

urgency for change, 240

use case technique, 45–46

user stories, 46–47

## V

values in agile development style, 20–23, 231–232

velocity, 189

Venn diagrams, 112, 113

virtual teams. *See* dispersed teams

vision: building for organizational change, 241, 242; product, 41–44, 184

voice of the customer (VOC), 53–54

## W

waterfall processes, 181, 193, 201

Watson, Sr., Thomas, 88

Williams, Laurie, 144, 146

Wohlers Associates, Inc., 97

## X

XP (Extreme Programming): about, 14–15; applied to personal care products, 26; assessment of, 23–25; coding standards, 19–20; collective code ownership, 18, 19–20, 21; continuous integration with, 18, 21, 26; design simplicity in, 16, 22, 26; feedback valued as practice, 21, 23; having customer on team, 18–19; lowering costs of iterations, 39–40; maintaining sustainable pace in, 18; origins of, 20; pairing, 17–18; product metaphor in, 16; project planning in, 15; refactoring, 17; small releases, 15–16; test-driven design, 16–17, 21; underlying values of, 20–23. *See also* iterations; pairing; refactoring

## Y

YAGNI (“you aren’t going to need it”), 228–230

Yahoo!, 50

Young, John, 240, 241, 243



