OTHER BACK MATTER

Notes (which provide detail without interrupting the flow), Bibliography (which backs up the notes), Customer Council (which advised us on content), and Index

from

Flexible Product Development

by

Preston G. Smith

Notes

Introduction

1. *Iteration* comes from the software development world (see Chapter One). In the lean manufacturing and lean product development fields, the related term is *small batch size,* for which see Reinertsen's *Managing the Design Factory.*

Chapter One

1. See the Boston Consulting Group's *Innovation 2006.*
2. Membership in the Project Management Institute stood at 485,000 with 762,000 credentialed PMPs as of March 2017 and has grown from 237,000 in 2007. See the Project Management Institute's Member Fact Sheet.
3. This is a short discussion of change in modern business. For a longer and more strategic treatment, See Brown and Eisenhardt's *Competing on the Edge.*
4. Arthur Fry and Spencer Silver invented the sticky note in 1970, based on Silver's 1968 invention of the "low-tack" adhesive involved. It was 1979 before 3M found a means for marketing it successfully. See en.wikipedia.org/wiki/Spencer_Silver and inventors.about.com/library/inventors/blpostit.htm (both accessed September 24, 2018).
5. See Smith and Reinertsen's *Developing Products in Half the Time.*
6. For a contemporary treatment, see Cooper, Edgett, and Kleinschmidt's *Portfolio Management for New Products.* For good coverage of the balance between new and mature products, see Roussel, Saad, and Erickson's *Third Generation R&D.*
7. See Boehm's *Software Engineering Economics,* p. 40.
8. See Boehm and Turner's *Balancing Agility and Discipline,* pp. 218–219.
9. See Terwiesch, Loch, and De Meyer's "Exchanging Preliminary Information in Concurrent Engineering." They provide detailed cost-of-change information for three automotive components, and their results support the 10x/phase rule. However, this is an example of the worst part of the Pareto principle. They only investigate the two most expensive transitions in the development process (toolmaking), which together yield a 100x cost increase.

10. See Larman's *Agile and Iterative Development.*
11. See agilemanifesto.org/principles.html, accessed September 24, 2018.
12. This discussion draws from Beck's *Extreme Programming Explained.* Although this book comes in two editions (dated 2000 and 2005), it is actually more like parts 1 and 2. The first edition lays out the basics, and the second edition, requiring some existing knowledge of XP, describes it more broadly, provides case studies, and covers more advanced topics, such as the roles of project planning and software testing and the scalability of XP to large teams, dispersed teams, and life-critical applications. Consequently, this discussion follows the first edition more than the second one. Also see en.wikipedia.org/wiki/Extreme_Programming (accessed September 24, 2018).
13. Although commonly used, the term *pair programming* is misleading even for software development, because these pairs not only program but design, test, integrate, and document. Compare to the shift from *concurrent engineering* to *concurrent development.* Thus I use simply *pairing* from here on.
14. See Boehm and Turner's *Balancing Agility and Discipline,* pp. 230–232.
15. See Beck's *Extreme Programming Explained* (First Edition), pp. 64–70, for more such relationships..
16. en.wikipedia.org/wiki/Extreme_Programming (Origins section) or informit.com/articles/article.asp?p=20972&rl=1 (both accessed September 24, 2018).
17. See Beck's *Extreme Programming Explained* (First Edition), pp. 29–34. The second edition adds respect and suggests that others, such as safety, predictability, and quality of life, may be important in certain situations.
18. See Boehm and Turner's *Balancing Agility and Discipline,* p. 231.
19. See Eckstein's *Agile Software Development in the Large.*

Chapter Two

1. I use *specification* and *requirement* interchangeably to mean the written description of the product going into development, but I know that some people make a clear distinction here. Some say that requirements are the wish list at the beginning of the project, while specifications appear in the catalog after commercialization. Others believe that requirements are for your product, and specifications relate to purchased components or materials. Please picture the term you prefer for the up-front description of what the product should be.
2. Basic data are from Thomke and Reinertsen's "Agile Product Development," updated via personal communication with Donald Reinertsen, December 11, 2006.
3. See MacCormack's "Developing Products on 'Internet Time': The Anatomy of a Flexible Development Process," "How Internet Companies Build Software," and "Creating a Fast and Flexible Process: Empirical Research Suggests Keys to Success."

4. See Boehm's "Prototyping Versus Specifying: A Multiproject Experiment" for details.
5. Source: The Standish Group, standishgroup.com, as reported at the 3rd International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2002), Alghero, Sardinia, Italy, May 2002.
6. Lynn and Reilly's *Blockbusters* lists a clear and stable product vision as one of five factors that distinguished what they call "blockbuster" products from others in a ten-year study of seven hundred development projects. For more examples of vision statements and how to create them, see their book.
7. From Jeffrey K. Liker, *The Toyota Way,* McGraw-Hill, 2004 (Figure 5-4). Reproduced with permission of The McGraw-Hill Companies.
8. See Morgan and Liker's *Toyota Product Development System,* p. 123.
9. See Morgan and Liker's *Toyota Product Development System,* p. 260. Also, private communication with the authors, November 6, 2006.
10. See Sobek's "Principles That Shape Product Development Systems," p. 226.
11. Cooper's *The Inmates Are Running the Asylum* describes this technique.
12. For more on this topic, see Cockburn's *Writing Effective Use Cases.*
13. For more on this topic, see Cohn's *User Stories Applied.*
14. You can often drive user stories to a higher, more change-resistant level by writing "epics," as Cohn describes.
15. See Hohmann's *Innovation Games.*
16. *Customers* (those who pay for your products) are not necessarily *users* (those who use them). For consumer products, the two are often the same, but for industrial products, they are usually different: a buyer or a manager may purchase the computer on your desk at work, but you have to use it and cope with it every day. I use the terms interchangeably, but please apply the pertinent one.
17. Sometimes Marketing and Sales personnel are nervous about letting engineers talk to customers, fearing that they might say something inappropriate. If so, train those involved using Hohmann's guidance described in *Beyond Software Architecture,* p. 60.
18. See Scoble and Israel's *Naked Conversations* for much more on business blogging.
19. In Hohmann's *Innovation Games.*
20. Both Toyota examples are from Morgan and Liker's *Toyota Product Development System,* p. 30.
21. For example, see Mariampolski's *Ethnography for Marketers,* and Perry and her colleagues' "Creating the Customer Connection."
22. For more on lead users, see von Hippel's *Sources of Innovation,* and for other examples and the 3M infection study, see his "Creating Breakthroughs at 3M."
23. See Christensen's *The Innovator's Dilemma.*
24. See Highsmith's *Agile Project Management,* p. 13.

Chapter Three

1. From Stevens, Myers, and Constantine, "Structured Design," p. 117. Quotation attributed to Constantine, who is considered the originator of the structured design technique used in software development and was the teacher of Stevens and Myers.
2. Adapted from The PDMA Glossary for New Product Development. See pdma.org/page/glossary_access2#P (accessed September 24, 2018).
3. For more on platforms and platform architecture, see Meyer and Lehnerd's *Power of Product Platforms* or Feitzinger and Lee's "Mass Customization at Hewlett-Packard: The Power of Postponement."
4. This shift between integral and modular architectures may appear to contradict Christensen and Raynor's observations in *The Innovator's Solution* (pp. 127–137). These authors work at the level of industries and describe how, in the early stages of an innovation (when I suggest that the need for flexibility and thus modularity is greatest), the product's performance falls below customer desires, so designers squeeze performance from the design. (integral architecture). Later, as the technology improves, performance exceeds needs and customers are unwilling to pay extra for performance, so the industry shifts to a modular architecture that allows more flexibility in tuning the product to individual needs at minimal cost. Observe that this strategic modularity aims more at changes during manufacture and distribution, that is, it is more of a platform architecture approach. In the early stages of an innovation, there is still a great need for designers to isolate areas of uncertainty and provide for reserve performance where needs are likely to grow, so wise designers apply modularity selectively at the design level, especially in the early stages of an innovation. This illustrates why flexibility techniques must be applied selectively so that performance will not suffer excessively just when Christensen and Raynor advise that it is most important.
5. Quotation from ferrariusa.com/design_f430text.php, accessed April 23, 2007.
6. These techniques are old. David Parnas provided an elegant (but technical) description of them for software nearly three decades ago. See Parnas' "Designing Software for Ease of Extension and Contraction."
7. Here I follow Ulrich and Eppinger's *Product Design and Development.*
8. For more on considering interfaces as design rules, see Baldwin and Clark's *Design Rules.*
9. A similar example is the naming of Windows XP covered earlier in this chapter.
10. See Thomke's "Role of Flexibility in the Development of New Products" for further discussion.

Chapter Four

1. Sources, respectively: Thomke's *Experimentation Matters,* p. 6; nmlites.org/standards/science/glossary_2.htm; and www. math.tamu.edu/FiniteMath/FinalBuild/Fall2001/Module9/ Introduction0.html.
2. Source: Archibald Lemon Cochrane. See his *Effectiveness and Efficiency,* p. 43.
3. See Thomke's *Experimentation Matters,* pp. 211–214.
4. For an extensive discussion of exploration and hypothesis-based experimentation, see Garvin's *Learning in Action,* chapter 5 (specifically what he calls the probe-and-learn process).
5. See Iansiti and MacCormack's "Developing Products on Internet Time."
6. The Orion case study was published by Z Corporation in 2001 but is no longer available.
7. See Thomke's "Capturing the Real Value of Innovation Tools."
8. See Smith and Reinertsen's *Developing Products in Half the Time,* chapter 2.
9. See, for example, Montgomery's *Design and Analysis of Experiments.*
10. The lock analogy was inspired by Thomke's *Experimentation Matters,* p. 110. For more on parallel versus sequential strategies, see Thomke and Bell's "Sequential Testing in Product Development."
11. *Testing* in its broadest sense includes testing an idea, a prototype, or even a hunch. Here I consider the narrower interpretation of testing a design, a product, or a part of a product.
12. See Peters and Austin's *Passion for Excellence,* p. 130. My second engineering job was for a manufacturer of aircraft engines twenty-plus years before their book appeared. At that time the test used a two-pound seagull, but the procedure was the same. This was regarded as the final test of an engine.

Chapter Five

1. Yogi Berra was a famous American baseball player from several decades ago, but today he is known better for his illogical but direct manner of speaking.
2. Engineering changes do *not* constitute flexibility. Engineering changes originate internal to engineering and in most cases arise from poor engineering judgment. Flexibility stems from changes external to engineering. Thus flexibility is beneficial when external change is likely, but engineering changes are usually an indication of mistakes.
3. See Ward, Liker, Cristiano, and Sobek's "Second Toyota Paradox." This article is also the source of the concept for Figure 5.4.
4. See Pugh's *Total Design,* section 4.8.

7. For additional details, see Morgan and Liker's *Toyota Product Development System,* pp. 269–274. For examples of A3 reports, guidance on writing them, and templates for writing the basic types, search the Web for "A3 report."
8. See Sobek, Ward, and Liker's "Toyota's Principles of Set-Based Concurrent Engineering."
9. The examples in this and the following few paragraphs come from Durward Sobek's Ph.D. dissertation.
10. Suggested by Katherine Radeka of Whittier Consulting Group.
11. See Ward, Liker, Cristiano, and Sobek's "Second Toyota Paradox."

Chapter Six

1. From Turner and Boehm's "People Factors in Software Management."
2. The book is Boehm and Turner's *Balancing Agility and Discipline.*
3 See Boehm's *Software Engineering Economics.*
4. See Quinn's *Building the Bridge As You Walk On It.* However, only the title of this book is pertinent here, as the book deals with personal change rather than project change.
5. See MacCormack's "How Internet Companies Build Software" for more on experience.
6. See Loch, DeMeyer, and Pich's *Managing the Unknown,* p. 41.
7 See Cockburn's *Agile Software Development,* pp. 14–18, and Boehm and Turner's *Balancing Agility and Discipline,* p. 48.
8. This has no connection with Douglas McGregor's Theory X, which postulates managers who regard workers as lazy and unwilling to work without strong structure and control.
9 This figure results from a multi-year study in a major electronics company that categorized engineers' activities as either adding or not adding value to their development projects.
10. In a 2006 article of the same name, Scott Ambler uses the term *generalizing specialists* and discusses both their benefits to the team and the techniques for cultivating them.
11. See agilemanifesto.org/principles.html, accessed September 24, 2018.
12. See Figure 8-7 (p. 156) of Smith and Reinertsen's *Developing Products in Half the Time.*
13. Many people call this a "virtual team," but I dislike this term. Teams exist to enhance performance, and it clouds the performance issue to use a term that means "being such in essence or effect though not formally recognized or admitted" (webster.com, accessed September 24, 2018). Are the members virtual, are the activities virtual, or what is it that exists only in essence or effect?

7. See Allen's *Managing the Flow of Technology*. His more contemporary *Organization and Architecture of Innovation* (pp. 58–61) reinforces his earlier work and concludes that electronic communication media, such as e-mail, lack the fidelity to substitute for face-to-face communication as distance increases.
8. As suggested at the beginning of this section, because effectiveness drops off gradually, various distances are used to define co-location. Reinertsen is somewhat more liberal in his definition than I am. Stephanie Teasley and her colleagues (quoted a few paragraphs later) are still more liberal.
9. The Olsons' article appeared in *Human-Computer Interaction* in 2000.
10. This work was published by Teasley, Covi, Krishnan, and Olson as "How Does Radical Collocation Help a Team Succeed?" Both of the quotations that follow this paragraph are reprinted with permission from that article, copyright ACM.
11. See Cockburn's *Agile Software Development,* pp. 84–88.
12. See Williams and Kessler's *Pair Programming Illuminated.*
13. From Duarte and Snyder's *Mastering Virtual Teams.* Used with permission.
14. See Duarte and Snyder's *Mastering Virtual Teams,* p. 42.

Chapter Seven

1. Quoted in Maier and Rechtin's *Art of Systems Architecting,* p. 272. Robert Spinrad is a retired vice president of technology strategy and director of the legendary PARC laboratories at Xerox. He is also a member of the prestigious National Academy of Engineering in the United States.
2. For example, see Deck's "Decision Making: The Overlooked Competency in Product Development."
3. "The last responsible moment" was coined in about 2000 by the Lean Construction Institute in work they were doing with the British Airports Authority to create the construction process for Terminal 5 at London's Heathrow Airport. They needed flexibility because the airlines' strategic plans were likely to change numerous times during the eight years needed to build the terminal. In *Lean Software Development,* Poppendieck and Poppendieck brought the term into agile software development, and I elaborate further.
4. In developing this section, I consulted some experts on building consensus. Each had a somewhat different way of using the consensus gradient. Consequently, what you see here is truly a consensus version of the topic, keeping the elements separate so that you can assemble them to suit your needs.
5. For this section, I am indebted to chapter 6 of Savage's excellent book, *Decision Making with Insight.* This chapter introduces decision trees and provides XLTree, an Excel add-in, for creating them. All decision trees in this chapter were created using XLTree. Unfortunately, XLTree isn't compatible with contemporary operating systems (Windows 10, for example), but plenty of other such tools are available.

7. For an introduction to utility theory and utility measures, see Savage's *Decision Making with Insight*, pp. 194–195.
8. For an example of real options applied to the product development process, see Huchzermeier and Loch's "Project Management Under Risk."
9. For a starter on real options applied to new product development, see Faulkner's "Applying 'Options Thinking' to R&D Valuation," Angelis' "Capturing the Option Value of R&D," and van Putten's "Making Real Options Really Work." Several other helpful articles are available in *Research-Technology Management* and the *Harvard Business Review*.
10. Traditionally, such valuations are made using discounted cash flow (DCF) methods, which assume the project will be completed according to the original plan. Even when used with a phased development process, where the purpose of the phases is that one can kill the project rather than continuing to invest, the project is still evaluated on the assumption of completing the plan. In contrast, the real options approach allows a project to be evaluated more flexibly by making future investments contingent on interim results.
11. For a discussion of how real options are equivalent to decision trees, see Faulkner's "Applying 'Options Thinking' to R&D Valuation."

Chapter Eight

1. The referenced guide is the Project Management Institute's *Guide to the Project Management Body of Knowledge*.
2. Several other books on agile project management have appeared in the last few years. I list DeCarlo's and Highsmith's because they generalize beyond software development better than most of the others, I believe. Interestingly, an *Agile Practice Guide*, jointly published by PMI and the Agile Alliance, appeared in 2017, apparently because agile had become such an important topic in PMI that it could no longer be ignored. This guide covers a few agile practices, such as stand-up meetings (Chapter Six in this book) and retrospectives (Chapter Eight), but it doesn't mention core topics, such as customer involvement (Chapter Two), experimentation (Chapter Four), or decision making (Chapter Seven).
3. Chapter Nine shows how to combine structured and flexible development processes in the same project, depending on the project's specific demands. You can use the same approach to adjust project management to the specific characteristics of a project.
4. See the Project Management Institute's *Guide to the Project Management Body of Knowledge*, p. 356.
5. See Portny's *Project Management for Dummies* (Second Edition), p. 14.
6. See the Project Management Institute's *Guide to the Project Management Body of Knowledge*, p. 369.
7. Chapter Nine covers anticipation relative to adaptation, which is investing in the capability to react quickly when anticipation is not possible or cost-effective.

8. See Figure 9.4 for an illustration of the way a project naturally shifts from anticipation to planning as it progresses.
9. For additional information on this method, see Githens's "Using a Rolling Wave for Fast and Flexible Development."
10. For more detail on agile loose-tight planning, see Cohn's *Agile Estimating and Planning.*
11. See Thomke's *Experimentation Matters,* pp. 168–169.
12. See Highsmith's *Agile Project Management,* p. 42.
13. Smith and Merritt's *Proactive Risk Management* is a good example of this literature.
14. Although most agilists have replaced waterfall (phased) processes with iterative ones, Jim Highsmith has created a means of combining a phased governance process (for managing project investments) with iterative development (for flexibility). See his "Agile for the Enterprise: From Agile Teams to Agile Organizations."
15. See Loch, DeMeyer, and Pich's *Managing the Unknown.* Although valuable, this book is unfortunately also expensive. However, you can read a lengthy review of it gratis at https://www.strategy2market.com/Preston-Smith/Book-Reviews/Managing-the-Unknown/ (accessed September 25, 2018).
16. See Weick and Sutcliffe's *Managing the Unexpected.*
17. Summarized with permission from Weick and Sutcliffe's *Managing the Unexpected,* pp. 10–17.
18. This metric was inspired by Iansiti's "Shooting the Rapids," Figure 1.
19. I intentionally use *feel* here, but I recognize that some people are quite uneasy in expressing their feelings or hearing about others' feelings. In this case, please see Derby and Larsen's *Agile Retrospectives* (p. 10) for ways of obtaining this information without using that *f* word directly.
20. See Morgan and Liker's *Toyota Product Development System,* p. 211.

Chapter Nine

1. See Boehm and Turner's *Balancing Agility and Discipline,* pp. 36–37 and p. 152.
2. See note 14 in Chapter Eight.
3. See Cockburn's "Learning from Agile Software Development."
4. For a description of how IT might be used to capture and manage product development knowledge, see McGrath's *Next Generation Product Development.*
5. For more on how Toyota manages tacit knowledge, see Morgan and Liker's *Toyota Product Development System,* pp. 204–205, 229, and 279–280. I describe A3 reports and engineering checklists in Chapter Five.
6. As discussed in the Introduction, using *structured* as the opposite of *flexible* has difficulties in that flexible development is very structured in certain

subtle ways. Agilists speak of high- and low-ceremony processes. Nevertheless, *traditional* (used in the rest of the book) seems to miss the mark when discussing process, so I will proceed with *structured* here, knowing that it is not a perfect antonym.

7. See Boehm and Turner's *Balancing Agility and Discipline*, chapter 5. For non-software projects, their approach must be modified in two ways. One is to shift it to a physical product, and the other is to narrow it to a focus on flexibility (Boehm and Turner consider other differences between agile and traditional approaches, such as scalability). Also, I do not follow their complete five-step process. For an alternative means of adjusting the process used to the needs of a specific project, also see the Project Analyzer discussed at the end of Chapter One.
8. The infrared measurement technologies and algorithms described here are fictitious.
9. Goldratt and Cox's *The Goal* (a business novel) is the classic on this topic, but more descriptive material can be found on the Web, for instance, en.wikipedia.org/wiki/Theory_of_Constraints (accessed September 25, 2018).
10. For bulk arrivals, there is no formula for the answer, so I used the Extend discrete-event simulation package from Imagine That, Inc. and covered in Savage's *Decision Making with Insight.* I assumed that arrivals appear in uniformly distributed clumps of one to five items (mean of three), as well as being exponentially distributed in time. Extend generated Figure 9.7 too.
11. From "What Testers Can Do About Technical Debt" by Johanna Rothman, https://www.cmcrossroads.com/article/what-testers-can-do-about-technical-debt-part-1 and https://www.cmcrossroads.com/article/what-testers-can-do-about-technical-debt-part-2, accessed September 25, 2018.

Chapter Ten

1. Brooks managed development of the OS/360 software system. The quote comes from his *Mythical Man-Month,* p. 242. "System" to him means software or hardware.
2. From Kotter and Cohen's *Heart of Change,* p. 40.
3. For more on pilot projects, see Smith and Reinertsen's *Developing Products in Half the Time,* chapter 15.
4. From Bridges' *Managing Transitions,* p. 37.
5. From Kotter and Cohen's *Heart of Change,* p. 2.
6. From Patterson's *Leading Product Innovation,* pp. 262–264.
7. See Schaffer's *Breakthrough Strategy* for more on this momentum-building process.
8. From Morgan and Liker's *Toyota Product Development System,* p. 227.
9. For more on patterns, see en.wikipedia.org/wiki/Software_ pattern, accessed September 25, 2018.
10. From Manns and Rising's *Fearless Change,* p. 5.

Chapter Eleven

1. This method of managing changes is strongly connected with the technique used to manage risk in a project. In fact, changes are just a subset of the risks in a project. A strong body of knowledge is available in the field of project risk management field. I have written a book on this subject that parallels the technique described in this chapter, and The Project Management Institute has published many books about project risk management, including their *PMBOK Guide*. If you search the Internet for help on the subject, be sure to search for *project* risk management; *risk management* alone will bring you information mostly about the insurance industry. But please remember that none of this substitutes for building an environment that accommodates change.
2. Note 1 mentions that managing anticipated changes is quite like project risk management. Those managing project risks may not have a template like Figure 11.1, but they follow the same steps. The big difference is that they do not have a Benefits box. Although they sometimes mention benefits (called opportunities in project risk management), very seldom do benefits actually occur. Anticipated changes, in contrast, can have substantial benefits, sometimes outweighing the costs.
3. Although this book is written in U.S. English, it is international in scope. Thus, I use a variety of currencies. U.S. dollars, euros, and Japanese yen have already been used, so the examples in this chapter employ the Chinese yuan or RMB (which translates into "people's money" in Chinese).
4. This value was obtained by using the cost of delay calculation described in chapter 2 of Smith and Reinertsen.
5. See chapter 2 of Smith and Reinertsen for a detailed explanation.
6. Although this may look complicated, it is actually a relatively simple model. It allows for only one benefit and one cost item per anticipated change. If you have more than one benefit or multiple costs associated with a single change, you can handle this by placing an i subscript on pertinent quantities below and summing over i, where i runs from 1 to the number of benefits or costs involved.
7. Chosen because it is the only change of the three to have both a benefit and a cost.
8. There are often multiple benefits or costs for a given change. Usually, some simple analysis will show that one benefit and one cost are dominant, and you can proceed with them alone. If one doesn't stand out, see note 6 above.
9. See the section entitled Tacit Knowledge in Chapter Nine.
10. Readers outside North America may need an explanation. Canada has a national health care system, which the United States lacks, so drug prices are regulated at much lower levels (for the same drugs) in Canada. Consequently, Canadian pharmacies have a huge market serving U.S. customers online. But drug companies and the regulatory authority (the Food and Drug Administration) in the United States oppose this, as it undercuts their power and authority. This creates volatile conditions as the various opposing parties act constantly to enhance their positions.

Bibliography

Allen, Thomas J. *Managing the Flow of Technology*. Cambridge, MA: MIT Press, 1977.

Allen, Thomas J., and Gunter W. Herr. *The Organization and Architecture of Innovation: Managing the Flow of Technology*. Amsterdam: Butterworth-Heinemann, 2007.

Ambler, Scott W. "Generalizing Specialists: Improving Your IT Career Skills," 2006. http://agilemodeling.com/essays/generalizingSpecialists.htm.

Angelis, Diana I. "Capturing the Option Value of R&D." *Research-Technology Management* 43(4): 31–34 (July–August 2000).

Avery, Christopher M. *Teamwork Is an Individual Skill: Getting Your Work Done When Sharing Responsibility*. San Francisco: Berrett-Koehler, 2001.

Baldwin, Carliss Y., and Kim B. Clark. *Design Rules: The Power of Modularity*. Cambridge, MA: MIT Press, 2000.

Beck, Kent. *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley, 2000.

Beck, Kent, and Cynthia Andres. *Extreme Programming Explained: Embrace Change. (*Second Edition). Boston: Addison-Wesley, 2005.

Boehm, Barry W. *Software Engineering Economics*. Upper Saddle River, NJ: Prentice Hall, 1981.

Boehm, Barry W., Terance E. Gray, and Thomas Seewaldt. "Prototyping Versus Specifying: A Multiproject Experiment." *IEEE Transactions on Software Engineering* SE-10(3): 290–302 (May 1984).

Boehm, Barry W., Ellis Horowitz, Ray Madachy, Donald Reifer, Bradford K. Clark, Bert Steece, A. Winsor Brown, Sunita Chulani, and Chris Abts. *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall, 2000.

Boehm, Barry, and Richard Turner. *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston: Addison-Wesley, 2004.

Boston Consulting Group. *Innovation 2006*. Boston: Boston Consulting Group, Inc., 2006.

Bridges, William, *Managing Transitions: Making the Most of Change* (Second Edition). Cambridge, MA: Da Capo Press, 2003.

Brooks, Frederick P., Jr., *The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition*. Reading, MA: Addison-Wesley, 1995.

Brown, Shona L, and Kathleen M. Eisenhardt. *Competing on the Edge: Strategy as Structured Chaos*. Boston: Harvard Business School Press, 1998.

Christensen, Clayton M. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Boston: Harvard Business School Press, 1997.

Christensen, Clayton M., and Michael E. Raynor. *The Innovator's Solution: Creating and Sustaining Successful Growth*. Boston: Harvard Business School Press, 2003.

Cochrane, Archibald Lemon. *Effectiveness and Efficiency: Random Reflections on Health Services*. London: Nuffield Provincial Hospitals Trust, 1972.

Cockburn, Alistair. *Writing Effective Use Cases*. Boston: Addison-Wesley, 2000.

Cockburn, Alistair. *Agile Software Development*. Boston: Addison-Wesley, 2002.

Cockburn, Alistair. "Learning from Agile Software Development—Part One." *Crosstalk: The Journal of Defense Software Engineering 15*(10): 10–14 (October 2002).

Cohn, Mike. *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley, 2004.

Cohn, Mike. *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice Hall, 2006.

Cooper, Alan. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis: SAMS, 1999.

Cooper, Robert G. "Your NPD Portfolio May Be Harmful to Your Business's Health," *Visions* 29(2): 22–26 (April 2005).

Cooper, Robert G., Scott J. Edgett, and Elko J. Kleinschmidt. *Portfolio Management for New Products, (*Second Edition). Cambridge, MA: Perseus Books, 2001.

DeCarlo, Doug. *eXtreme Project Management: Using Leadership, Principles, and Tools to Deliver Value in the Face of Reality*. San Francisco: Jossey-Bass, 2004.

Deck, Mark J. "Decision Making: The Overlooked Competency in Product Development." In *The PDMA ToolBook 1 for New Product Development,* Paul Belliveau, Abbie Griffin, and Stephen Somermeyer, eds. New York: Wiley, 2002, pp. 165–185.

Derby, Esther, and Diana Larsen. *Agile Retrospectives: Making Good Teams Great*. Raleigh, NC: Pragmatic Bookshelf, 2006.

Duarte, Deborah L., and Nancy Tennant Snyder. *Mastering Virtual Teams: Strategies, Tools, and Techniques That Succeed* (Third Edition). San Francisco: Jossey-Bass, 2006.

Eckstein, Jutta. *Agile Software Development in the Large: Diving into the Deep.* New York: Dorset House, 2004.

Faulkner, Terrence W. "Applying 'Options Thinking' to R&D Valuation." *Research-Technology Management* 39(3): 50–56 (May–June 1996).

Feitzinger, Edward, and Hau L. Lee. "Mass Customization at Hewlett-Packard: The Power of Postponement." *Harvard Business Review* 75(1): 116–121 (January-February 1997).

Garvin, David A. *Learning in Action: A Guide to Putting the Learning Organization to Work*. Boston: Harvard Business School Press, 2000.

Githens, Gregory D. "Using a Rolling Wave for Fast and Flexible Development." In *The PDMA ToolBook 3 for New Product Development,* Abbie Griffin and Stephen Somermeyer, eds. 397–415. Hoboken, NJ: Wiley, 2007.

Goldratt, Eliyahu M., and Jeff Cox, *The Goal* (Third Edition). Great Barrington, MA: North River Press, 2004.

Highsmith, Jim. *Agile Project Management: Creating Innovative Products*. Boston: Addison-Wesley, 2004.

Highsmith, Jim. "Agile for the Enterprise: From Agile Teams to Agile Organizations." *Cutter Consortium Agile Project Management Advisory Service Executive Report* 6(1) (2005).

Hohmann, Luke. *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Boston: Addison-Wesley, 2003.

Hohmann, Luke. *Innovation Games: Creating Breakthrough Products Through Collaborative Play*. Upper Saddle River, NJ: Addison-Wesley, 2007.

Huchzermeier, Arnd, and Christoph H. Loch. "Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D." *Management Science* 47(1): 85–101 (January 2001).

Iansiti, Marco. "Shooting the Rapids: Managing Product Development in Turbulent Environments." *California Management Review* 38(1): 37–58 (Fall 1995).

Iansiti, Marco, and Alan MacCormack. "Developing Products on Internet Time." *Harvard Business Review* 75(5): 108–117 (September-October 1997).

Kerth, Norman L. *Project Retrospectives: A Handbook for Team Reviews*. New York: Dorset House, 2001.

Kotter, John P., and Dan S. Cohen. *The Heart of Change: Real-Life Stories of How People Change Their Organizations*. Boston: Harvard Business School Press, 2002.

Larman, Craig. *Agile and Iterative Development: A Manager's Guide*. Boston: Addison-Wesley, 2004.

Loch, Christoph, Arnoud DeMeyer, and Michael T. Pich. *Managing the Unknown: A New Approach to Managing High Uncertainty and Risk in Projects*. Hoboken, NJ: Wiley, 2006.

Lynn, Gary S., and Richard R. Reilly. *Blockbusters: The Five Keys to Developing Great New Products*. New York: Harper Business, 2002.

MacCormack, Alan. "How Internet Companies Build Software." *Sloan Management Review* 42(2): 75–84 (Winter 2001).

MacCormack, Alan. "Creating a Fast and Flexible Process: Empirical Research Suggests Keys to Success." *Product Development Best Practices Report* 10(8): 1–4 (August 2003).

MacCormack, Alan, Roberto Verganti, and Marco Iansiti. "Developing Products on 'Internet Time': The Anatomy of a Flexible Development Process." *Management Science* 47(1): 133–150 (January 2001).

Maier, Mark W., and Eberhardt Rechtin. *The Art of Systems Architecting* (Second Edition). Boca Raton, FL: CRC Press, 2002.

Manns, Mary Lynn, and Linda Rising. *Fearless Change: Patterns for Introducing New Ideas.* Boston: Addison-Wesley, 2005.

Mariampolski, Hy. *Ethnography for Marketers: A Guide to Consumer Immersion*. Thousand Oaks, CA: Sage, 2006.

McGrath, Michael E. *Next Generation Product Development: How to Increase Productivity, Cut Costs, and Reduce Cycle Times*. New York: McGraw-Hill, 2004.

Meyer, Marc H., and Alvin P. Lehnerd. *The Power of Product Platforms: Building Value and Cost Leadership*. New York: Free Press, 1997.

Montgomery, Douglas C. *Design and Analysis of Experiments*. Hoboken, NJ: Wiley, 2005.

Morgan, James M., and Jeffrey K. Liker. *The Toyota Product Development System: Integrating People, Process, and Technology*. New York: Productivity Press, 2006.

Olson, Gary M., and Judith S. Olson. "Distance Matters." *Human-Computer Interaction* 15(2–3): 139–178 (2000).

Parnas, David L. "Designing Software for Ease of Extension and Contraction." *IEEE Transactions on Software Engineering* SE-5(2): 128 –138 (March 1979).

Patterson, Marvin L. *Leading Product Innovation: Accelerating Growth in a Product-Based Business*. New York: Wiley, 1999.

Perry, Barbara, Cara L. Woodland, and Christopher W. Miller. " Creating the Customer Connection: Anthropological/Ethnographic Needs Discovery." In *The PDMA ToolBook 2 for New Product Development*, Paul Belliveau, Abbie Griffin, and Stephen M. Somermeyer, eds. 201–234. Hoboken, NJ: Wiley, 2004.

Peters, Tom, and Nancy Austin. *A Passion for Excellence*. New York: Random House, 1985.

Poppendieck, Mary, and Tom Poppendieck. *Lean Software Development: An Agile Toolkit*. Boston: Addison-Wesley, 2003.

Portny, Stanley E. *Project Management for Dummies* (Second Edition). Indianapolis, IN: Wiley, 2007.

Project Management Institute. *A Guide to the Project Management Body of Knowledge* (Third Edition). Newtown Square, PA: Project Management Institute, 2004.

Project Management Institute. *Agile Practice Guide*. Newtown Square, PA: Project Management Institute, 2017.

Project Management Institute. Member Fact Sheet. Available online (search on "PMI Fact File"). Accessed September 26, 2018.

Pugh, Stuart. *Total Design*. Wokingham, UK: Addison-Wesley, 1991.

Quinn, Robert E. *Building the Bridge As You Walk On It: A Guide for Leading Change*. San Francisco: Jossey-Bass, 2004.

Reinertsen, Donald G. *Managing the Design Factory: A Product Developer's Toolkit*. New York: Free Press, 1997.

Roussel, Philip A., Kamal N. Saad, and Tamara J. Erickson. *Third Generation R&D: Managing the Link to Corporate Strategy*. Boston: Harvard Business School Press, 1991.

Savage, Sam L. *Decision Making with Insight*. Belmont, CA: Brooks/Cole, 2003.

Schaffer, Robert H. *The Breakthrough Strategy: Using Short-Term Success to Build the High-Performance Organization*. Cambridge, MA: Ballinger, 1988.

Scoble, Robert, and Shel Israel. *Naked Conversations: How Blogs Are Changing the Way Businesses Talk with Customers*. Hoboken, NJ: Wiley, 2006.

Smith, Preston G., and Guy M. Merritt. *Proactive Risk Management: Controlling Uncertainty in Product Development*. New York: Productivity Press, 2002.

Smith, Preston G., and Donald G. Reinertsen. *Developing Products in Half the Time* (Second Edition). New York: Wiley, 1998.

Sobek, II, Durward Kenneth. "Principles That Shape Product Development Systems: A Toyota-Chrysler Comparison." Ph.D. dissertation, University of Michigan, 1997.

Sobek, II, Durward K., Allen C. Ward, and Jeffrey K. Liker. "Toyota's Principles of Set-Based Concurrent Engineering." *Sloan Management Review* 40(2): 67–83 (Winter 1999).

Stevens, Wayne G., Glen J. Myers, and Larry L. Constantine, "Structured Design," *IBM Systems Journal* 13(2): 115–139 (May 1974).

Teasley, Stephanie, Lisa Covi, M. S. Krishnan, and Judith S. Olson. "How Does Radical Collocation Help a Team Succeed?" Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Philadelphia, pp. 339–346, 2000.

Terwiesch, Christian, Christoph H. Loch, and Arnoud De Meyer. "Exchanging Preliminary Information in Concurrent Engineering: Alternative Coordination Strategies." *Organization Science* 13(4): 402–419 (July/August 2002).

Thomke, Stefan H. "The Role of Flexibility in the Development of New Products: An Empirical Study." *Research Policy* 26(1): 105–119 (March 1997).

Thomke, Stefan H. *Experimentation Matters: Unlocking the Potential of New Technologies for Innovation*. Boston: Harvard Business School Press, 2003.

Thomke, Stefan H. "Capturing the Real Value of Innovation Tools." *Sloan Management Review* 47(2): 24–32 (Winter 2006).

Thomke, Stefan, and David E. Bell. "Sequential Testing in Product Development." *Management Science* 47(2): 308–323 (February 2001).

Thomke, Stefan, and Donald Reinertsen. "Agile Product Development: Managing Development Flexibility in Uncertain Environments." *California Management Review* 41(1): 8–30 (Fall 1998).

Turner, Richard, and Barry Boehm. "People Factors in Software Management: Lessons from Comparing Agile and Plan-Driven Methods." *Crosstalk: The Journal of Defense Software Engineering* 16(12): 4–8 (December 2003).

Ulrich, Karl T., and Steven D. Eppinger. *Product Design and Development* (Third Edition). New York: McGraw-Hill, 2003.

van Putten, Alexander B., and Ian C. Macmillan. "Making Real Options Really Work." *Harvard Business Review* 82(12): 134–141 (December 2004).

von Hippel, Eric. *The Sources of Innovation*. New York: Oxford University Press, 1988.

von Hippel, Eric, Stefan Thomke, and Mary Sonnack. "Creating Breakthroughs at 3M." *Harvard Business Review* 77(5): 47–57 (September–October 1999).

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster." *Sloan Management Review* 36(3): 43–61 (Spring 1995).

Weick, Karl E., and Kathleen M. Sutcliffe. *Managing the Unexpected*. San Francisco: Jossey-Bass, 2001.

Wheelwright, Steven C., and Kim B. Clark. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. New York: Free Press, 1992.

Williams, Laurie, and Robert Kessler. *Pair Programming Illuminated*. Boston: Addison-Wesley, 2003.

Customer Council

Many people have contributed generously to this book, for which I am most grateful. Without them, the book would not have been possible. However, I would like to acknowledge in particular a "customer council" of lead users of flexibility techniques, the people who have played a key role in reviewing chapters, suggesting improvements, offering examples, and providing sources of associated material. I am especially grateful to these individuals on the customer council, each of whom has made an important contribution to the book:

Name	*Affiliation*	*Country*
Bob Becker	Product Development Advantage Group	United States
Chuck Blevins	LifeScan, Inc. (Johnson & Johnson)	United States
Jim Callahan, PMP	C-Cor, Incorporated	United States
Alan Chachich	Breakthrough NPD	United States
Frankie Chan	T&K Industrial Company Limited (China)	Hong Kong
Mike Clem	Cordis Accelerated Medical Ventures (Johnson & Johnson)	United States
Mike Cohn	Mountain Goat Software	United States
Mike Dowson	Draeger Safety	United Kingdom
Ricco Estanislao	Johnson & Johnson Worldwide Emerging Markets Innovation Center	China

John Farnbach	Silver Streak Partners LLC	United States
Gregory D. Githens, PMP, NPDP	Catalyst Management Consulting, LLC	United States
Mike Griffiths	Independent Consultant	Canada
David Gunderson	Micro Power Electronics, Inc.	United States
Greg Krisher	GE Water & Process Technologies, Analytical Instruments	United States
Jeff Oltmann, PMP	Synergy Professional Services, LLC	United States
Barry Papoff	Harris Corporation	Canada
David Petrie	California State University, Fullerton	United States
Roman Pichler	Pichler Consulting Ltd	United Kingdom
Katherine Radeka	Whittier Consulting Group, Inc.	United States
Radhika Ramnath	Cadence Design Systems	India
John Shambroom	Shambroom Associates, LLC	United States
Kulbhushan Sharma	Quark Media House India Pvt. Ltd.	India
Thomas Sigemyr	IVF	Sweden
Reardon Smith NPDP, CMC	Business Vectors Inc.	United States
James Joseph Snyder	Medtronic, Inc.	United States
Gary J. Summers, Ph.D.	Skillful NPD	United States
John Tepper	Alpha Med-Surge, Inc.	United States

Index