HARVARD | BUSINESS | SCHOOL

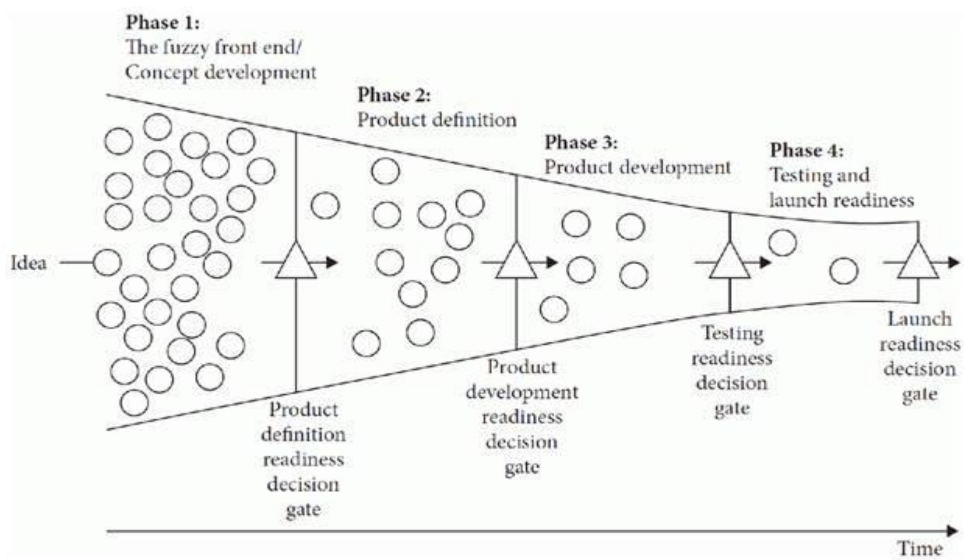
9-617-024
OCTOBER 3, 2016

MARCO IANSITI
MICHAEL TOFFEL
KERRY HERMAN
JULIA KELLEY

Product Development Fundamentals

The management of product development can be viewed as a process depicted as a narrowing of options and characterized by capacity, bottlenecks, and structure.[1] A product development process shares similarities with a manufacturing process, but also has significant differences. Most notably, product development typically entails individual projects as opposed to repetitive processes. Product development projects can be visualized using a product development funnel diagram, as depicted in **Figure A**. The funnel shows that each product development project entails a gradual narrowing of uncertainty, as one starts with a broad range of design options and ends with a concrete product being distributed to users.[2] The process features a series of screens and milestones.

Figure A The Product Development Funnel



Source: Maximilian von Zedtwitz, Sascha Friesike, and Oliver Gassman, "Managing R&D and New Product Development," in *The Oxford Handbook of Innovation Management*, edited by Mark Dodgson, David M. Gann, and Nelson Phillips (Oxford: Oxford University Press, 2015), via Oxford Handbooks Online, accessed September 2016.

Professors Marco Iansiti and Michael Toffel and Associate Director Kerry Herman and Associate Case Researcher Julia Kelley (Case Research & Writing Group) prepared this note as the basis for class discussion.

Product Development Team Structure

Many companies tend to structure their product development activities in one of four ways (see **Figure B**).[3]

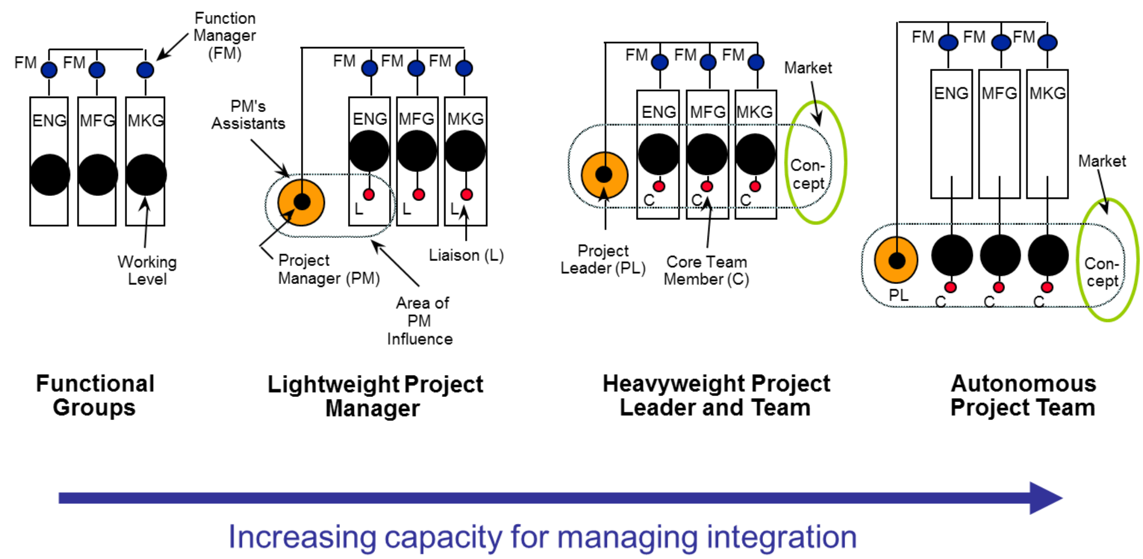
Functional Team Structure In this structure, product development projects are conducted as a series of tasks conducted by several distinct functions (e.g., engineering, manufacturing, marketing). Each function is responsible for a distinct set of tasks and has its own functional manager. Most communication about a project occurs within each functional group while it executes its tasks, and project coordination is executed by the functional managers.[4]

Lightweight Team Structure The lightweight team structure is similar to the functional team structure but includes a "lightweight" project manager who is primarily responsible for logistical tasks such as scheduling meetings, tracking progress, and enabling communication. Functional groups act with little oversight from the project manager, in some cases dedicating a member to liaise with the project manager to discuss the group's progress.[5]

Heavyweight Team Structure This structure, like those described above, also relies on functional teams wherein team members report to functional managers. A distinctive feature of this structure is the use of "heavyweight" project managers who, in contrast to lightweight project managers, deeply influence the product development team's work and shape critical project-related decisions on key features and other priorities. Heavyweight project managers have significant responsibility and influence, and are often senior managers of their organizations.[6]

Autonomous (or Dedicated) Team Structure In this structure, team members are pulled out of their functional departments and reassigned to work as a self-contained product development team, often for the entire duration of the product development effort. Similar to the heavyweight team structure, autonomous project teams are overseen by heavyweight project managers.[7]

Figure B Alternative Structures for Product Development Teams



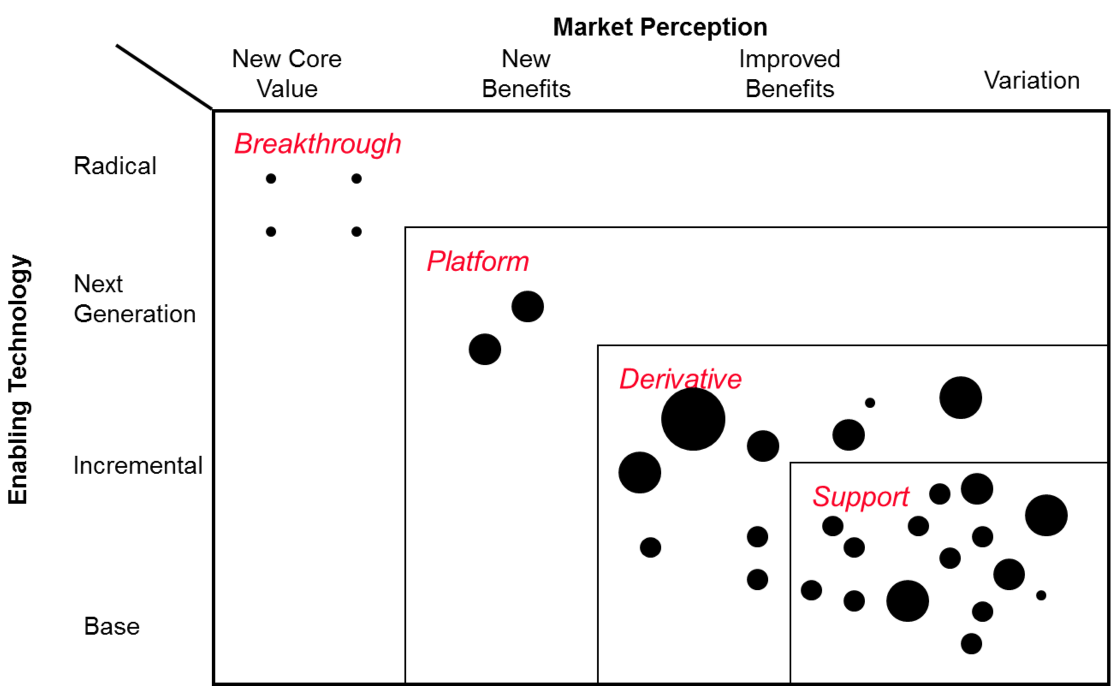
Source: Casewriters, adapted from Steven C. Wheelwright and Kim B. Clark, *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality* (New York: The Free Press, 1992), p. 191.

Research shows that particular project structures are better suited to different situations.[8] Functional and lightweight structures are often most effective for executing large numbers of relatively small, simple projects. Heavyweight and autonomous structures are often best for managing small numbers of large, complex projects, and often much more effective in situations where project uncertainty is particularly high, as is most often the case in venture-backed startups and other highly innovative projects.[9]

Types of Development Projects

A new product development project can be classified into one of four primary types: support, derivative, breakthrough, or platform, depending on the project's market perception and enabling technology (see **Figure C**).[10] Generally speaking, the more the projects involves new technology or new market concepts, the more integration they require, which motivates the use of heavyweight and dedicated teams. **Figure C** shows a full project portfolio for a mid-sized business unit of a particular medical device company. Each bubble represents a product development project; the size of each bubble is proportional to the resources currently devoted to it.

Figure C Defining and Mapping Different Types of Development Projects



Source: Casewriters.

As summarized in **Figure C**, projects to develop new "breakthrough" products depend on *radical,* or entirely new, technologies; these products are often perceived in the market as moving towards a fundamentally *new core value*, or something that has not been done before.[11] Projects to develop new "platform" products use *next generation* technology, which may require significant changes to manufacturing processes or product dimensions; platform products typically have *new benefits,* or new features that are better tailored to customers' needs.[12] Projects to develop new "derivative" products generally rely on *incremental,* or relatively small, modifications that require "little or no manufacturing process change."[13] Derivative projects typically are perceived as having *improved benefits*, such as updated product features.[14] "Support" projects are aimed at implementing *base*, or simple, product improvements, such as quality fixes or packaging changes. In most cases, as shown in **Figure C**, companies develop several products at once, often a combination of support, derivative, breakthrough, and platform projects.

Figure C illustrates the focus of the business unit's product development organization by revealing that most projects in its portfolio are incremental (support or derivative). It also shows that the four breakthrough projects appear to be receiving very little resource investment. This information is useful to senior management, as it can be used to analyze and optimize resource allocation. If the company with the portfolio depicted in **Figure C** is trying to drive innovation, it might want to reduce the number of and investment in support and derivative projects and increase investment in breakthrough and platform projects.[a]

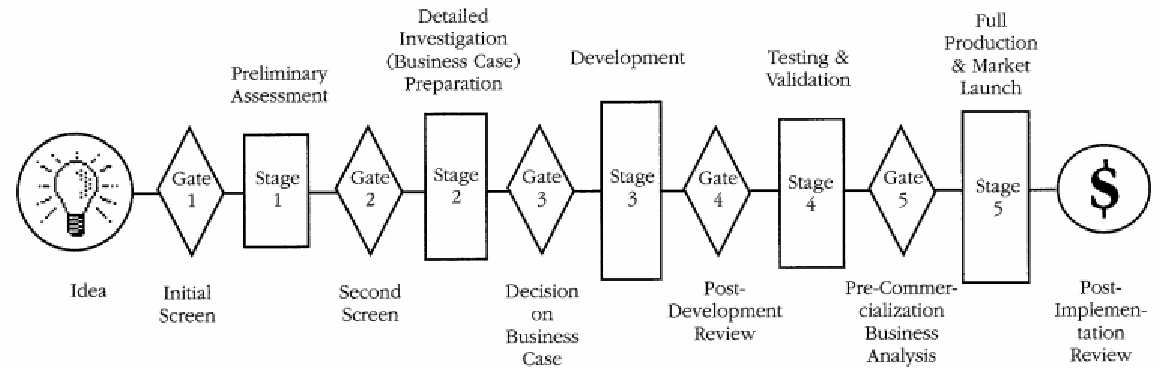
The Product Development Process

In traditional industry settings, many firms benefit from the structure of formal approaches to product development, which are designed to minimize uncertainty and reduce iterations. Such techniques include the stage-gate process and the critical-path method. In industries that are subject to rapid change or high uncertainty, it may be more beneficial to use integrated, iterative approaches, such as agile methodologies.

Stage-Gate Process

In a stage-gate process, a project goes through various stages (or phases), each of which includes a particular set of tasks and is separated from the next sequential stage by a "gate" (see **Figure D**). Each gate represents the necessary requirements or hurdles that a project has to meet before proceeding to the next stage.[15] Each gate has a set of inputs (the deliverables brought to the gate), exit criteria (goals the product must achieve by this point in the process), and an output (the decision to move forward or not).[16] Each stage requires final managerial review and approval before funding is released for the next phase.[17]

Figure D Stage-Gate System Overview



Source: Robert G. Cooper, "Stage-Gate Systems: A New Tool for Managing New Products," *Business Horizons* 33, no. 3 (May-June 1990): 46, via ScienceDirect, accessed August 2016.

[a] For a more detailed analysis, see Steven C. Wheelwright and Kim B. Clark, *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality* (New York: The Free Press, 1992).

Critical-Path

The critical-path method (CPM) of project management clearly separates the activities of planning—stating the steps necessary to complete a project—and scheduling—developing a specific project plan based on the project's timeframe and costs.[18] Just as "in construction work, forms must be built before concrete can be poured" and "in advertising, artwork must be made before layouts can be done," for research and development and for product planning, "specs must be determined before drawings can be made."[19]

Using the CPM often entails creating a chart in which each project step is represented by an arrow, which then points to another step that cannot be undertaken until the first task is completed. In some cases, two or more tasks might need to be completed before another particular activity can begin. The points at which arrows intersect are referred to as events, and the beginning and the end of a project are known as the origin and the terminus, respectively.[20] Alternatively, CPM charts sometimes represent different tasks with circles instead of arrows, and these circles are then linked by arrows to illustrate the relationship between various activities.[21]

In both cases, the "critical path" refers to the model's longest sequence of activities, as this sequence determines how long the entire project will take to complete; only by reducing the length of the critical path can the total project duration be shortened.[22] The CPM involves the use of mathematical formulas to determine the optimal times to begin and end each task, which allows product development managers to better assess a project's progress over time. These formulas also take into account the most efficient allocation of resources to complete a project.[23]

Agile

Several researchers who questioned the traditional "linear" framing of product development began to emphasize more integrated, iterative approaches (examples include the "rugby" approach.[24]) Research on the management of product development in rapidly changing environments of high uncertainty, such as hardware and software projects, concluded that the best-performing projects were highly iterative, featuring rapid experimentation and a flexible, adaptive approach to innovation.[25]

Such approaches correlated with superior speed and productivity, therefore suiting the needs of internet companies. Iterative, flexible approaches were adopted across the board for the development of internet services in many organizations, including Amazon, Google, and Yahoo. Flexible approaches to the management of product development matched the evolution of engineering processes, revolutionized by emerging methodologies such as extreme programming, described below.

In 2001, a group of software developers published the "Agile Manifesto," which featured 12 core principles for an agile development process. (See **Exhibit 1**.) The culmination of years of developing and practicing theories for product and software development, the Agile Manifesto called for developers to value "Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan."[26]

Agile is an iterative, collaborative process that emphasizes users' needs. In order to avoid creating a finished product that no longer fits customer requirements, agile seeks to generate useful pieces of a product early in the development process. As one expert summarized, agile practices "emphasize fast and early delivery of pieces of software incrementally in shorter timescales."[27]

Unlike the waterfall model (a sequential process in which a project flows through different phases, each of which is separated by a review) and other software development strategies that emphasize planning before coding, all types of agile methodology focus on always having working code that can be adjusted as needed throughout the development process. Agile practices focus on face-to-face communication rather than frequent documentation, thereby accelerating a project's completion. As a result, they allow for product guidelines to adapt as needed throughout the process, making them better suited to changing business environments and customer needs.[28]

Potential drawbacks to agile include the difficulty of completing longer and more complicated projects without a clear plan, and the risk of wasting time and resources while trying to be flexible. A reliance on face-to-face communication over formal documentation can result in a communication breakdown, particularly for complex projects that are completed over a long period of time.[29]

Agile product development features several specific methodologies, including:

Scrum Scrum is a simple process framework—a set of practices followed in order with as little overhead as possible—for agile development. Using Scrum, development team members and the team leader (referred to as the Scrum Master) work in sprints, or "short, sustainable bursts of energy."[30] Sprints are interspersed with frequent team meetings to discuss sprint results and the project's overall progress. Using iterative and incremental practices, Scrum is often used to manage complex software and product development. Proponents believe that Scrum reduces lead time relative to classic waterfall processes, and allows the organization to adapt to changing requirements and business goals.[31]

Extreme Programming Extreme Programming (XP) has been described as a "no nonsense, code first approach to software delivery,"[32] and emphasizes four key activities: coding, testing, listening and designing. It challenges traditional waterfall methods by bringing testing to the forefront of the delivery process. XP helped popularize automated testing, test-driven development, and other software engineering practices.[33]

Feature-Driven Development While most agile techniques work best for small projects, Feature-Driven Development (FDD) was designed for larger teams. It aims to encourage creativity and innovation by containing just enough process to ensure scalability. FDD's five-step process includes: 1) developing an overall model, 2) building a features list, 3) planning by feature, 4) designing by feature, and 5) building by feature.[34]

Since the 2000s, the rapid spread of software in products—and the advent of a broad range of powerful computer-based design tools—allowed traditional companies to improve iteration times, the time necessary to complete one experimental test or prototype. This led to the application of more flexible and agile approaches to product development in hardware and traditional industries.[35]

Elements from traditional and flexible approaches can be combined in order to achieve a more efficient product development strategy, particularly in manufacturing-based industries. For example, the stage-gate framework can be used along with agile techniques—specifically, elements from the Scrum method, such as sprints—during various phases, particularly the development and testing stages.[36] A hybrid model might work best for manufacturing projects in which "there is a high uncertainty and a great need for experimentation and failing fast."[37]

Product Development Metrics

A product development process can be analyzed using several metrics.

Lead Time The most common metric used to assess a product development process is the *product development lead time* or, alternatively, *time to market.* This is defined as the time elapsed between the project's start and completion; it measures the length of time required for a project to traverse through the product development funnel (refer to **Figure A**). A shorter time to market is generally perceived to be more desirable, as it provides for greater responsiveness. However, projects using agile approaches can be modified after they start, as their teams can more easily adjust to changing customer requirements during subsequent project iterations compared to those following a less flexible approach. The project's time to market is, in essence, the throughput time of a project going through a product development process.

Capacity As with manufacturing processes, product development processes also have a maximum output rate, usually defined as the number of products that a product development organization introduced to the market per unit of time. A product development capacity metric is more of a general guideline than a fixed rate because organizations tend to simultaneously develop several products that vary in complexity and uncertainty.

Productivity Product development productivity is typically defined as the number of products developed per unit of effort (or cost). Product development productivity typically refers to the number of products developed per R&D resource (or person-hour) of effort.

Many factors influence product development performance.[b] For example, research indicates that product development organizations perform better when they use processes that match key project dimensions such as complexity and uncertainty. Projects that face more complexity and uncertainty will be more effectively managed by heavyweight and autonomous organizations, as well as by using more flexible and agile processes. By contrast, projects that face less complexity and uncertainty are generally more effectively managed through functional and lightweight organizations, as well as by using more formal approaches to product development.

Differences in development performance across organizations can be quite significant. Productivity metrics, for example, are quite sensitive to process and the quality of resources, and can vary by more than a factor of two across organizations. Additionally, productivity is quite sensitive to the number of projects to which each team member is assigned. Generally speaking, an individual's productivity declines as he or she is assigned to a greater number of projects, as additional projects necessitate that additional time be spent on "non-value-added tasks" such as "coordinating, remembering, or tracking down information."[38] Using Queuing Theory, for variable processes such as product development, high utilization of available capacity significantly increases the amount of time that projects are on hold. If managers increase capacity utilization from 80% to 90%, for example, the waiting time doubles; waiting time doubles again if managers increase capacity from 90% to 95%.[39]

[b] See, for example, Kim B. Clark and Takahiro Fujimoto, Product Development Performance: Strategy, Organization, and Management in the World Auto Industry (Boston: Harvard Business School Press, 1991), Marco Iansiti, Technology Integration: Making Critical Choices in a Dynamic World (Boston: Harvard Business School Press, 1998), and Gary P. Pisano, The Development Factory: Unlocking the Potential of Process Innovation (Boston: Harvard Business School Press, 1997).

Exhibit 1 Principles Behind the Agile Manifesto Published in 2001

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas, "Manifesto for Agile Software Development," 2001, http://agilemanifesto.org/, accessed August 2016.

Endnotes

[1] Steven C. Wheelwright and Kim B. Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality, (New York: The Free Press, 1992).

[2] Maximilian von Zedtwitz, Sascha Friesike, and Oliver Gassman, "Managing R&D and New Product Development," in *The Oxford Handbook of Innovation Management*, edited by Mark Dodgson, David M. Gann, and Nelson Phillips (Oxford: Oxford University Press, 2015), via Oxford Handbooks Online, accessed September 2016.

[3] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

[4] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

[5] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

[6] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

[7] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

[8] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

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

[10] Steven C. Wheelwright and Kim B. Clark, "Creating Project Plans to Focus Product Development," *Harvard Business Review* 73 (March 1992): 70-82, via EBSCOhost, accessed September 2016.

[11] Wheelwright and Clark, "Creating Project Plans to Focus Product Development."

[12] Wheelwright and Clark, "Creating Project Plans to Focus Product Development."

[13] Wheelwright and Clark, "Creating Project Plans to Focus Product Development."

[14] Wheelwright and Clark, "Creating Project Plans to Focus Product Development."

[15] Robert G. Cooper, "Stage-Gate Systems: A New Tool for Managing New Products," *Business Horizons* 33, no. 3 (May-June 1990): 44-54, via ScienceDirect, accessed August 2016.

[16] Cooper, "Stage-Gate Systems: A New Tool for Managing New Products."

[17] Robert G. Cooper, "Third-Generation New Product Processes," *Journal of Product Innovation Management* 11, no. 1 (January 1994): 3-14, via ScienceDirect, accessed August 2016.

[18] James E. Kelley, Jr. and Morgan R. Walker, "Critical-Path Planning and Scheduling," 1959, Eastern Joint Computer Conference, The IEEE Computer Society, https://www.computer.org/csdl/proceedings/afips/1959/5055/00/50550160.pdf, accessed September 2016.

[19] Kelley and Walker, "Critical-Path Planning and Scheduling."

[20] Kelley and Walker, "Critical-Path Planning and Scheduling."

[21] F. K. Levy, G. L. Thompson, and J. D. West, "The ABCs of the Critical Path Method," *Harvard Business Review* 41, no. 5 (September 1963): 98-108.

[22] Levy, Thompson, and West, "The ABCs of the Critical Path Method."

[23] Kelley and Walker, "Critical-Path Planning and Scheduling."

[24] Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game," *Harvard Business Review* 64 (January 1986): 137-146, via EBSCOhost, accessed September 2016.

[25] Iansiti and MacCormack, "Developing Products on Internet Time."

[26] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas, "Manifesto for Agile Software Development," 2001, http://agilemanifesto.org/, accessed August 2016.

[27] Subhas Misra, "Agile Software Development Practices: Evolution, Principles, and Criticisms," *The International Journal of Quality & Reliability Management* 29, no. 9 (2012): 972-980, via ProQuest, accessed August 2016.

[28] Jim Highsmith and Alistair Cockburn, "Agile Software Development: The Business of Innovation," *Computer* 34, no. 9 (September 2001): 120-122, via IEEE Xplore Digital Library, accessed September 2016.

[29] Jordan B. Barlow, Justin Scott Giboney, Mark Jeffrey Keith, David W. Wilson, Ryan M. Scheutzler, Paul Benjamin Lowry, and Anthony Vance, "Overview and Guidance on Agile Development in Large Organizations," *Communications of the Association for Information Systems* 29 (2011): 25-44, via AIS Electronic Library, accessed September 2016.

[30] Chris Sims and Hillary Louise Johnson, "Scrum: A Breathtakingly Brief and Agile Introduction," Agile Learning Labs, http://www.agilelearninglabs.com/resources/scrum-introduction/, accessed September 2016.

[31] "What is Agile? What is Scrum?" cPrime, n.d., https://www.cprime.com/resources/what-is-agile-what-is-scrum/, accessed September 2016.

[32] Agile in a Nutshell, http://www.agilenutshell.com/xp, accessed September 2016.

[33] Agile in a Nutshell, http://www.agilenutshell.com/xp, accessed September 2016.

[34] Scott Ambler, "Agile Modeling and Feature-Driven Development," Dr. Dobb's, February 24, 2003, http://www.drdobbs.com/agile-modeling-and-feature-driven-develo/184415799, accessed September 2016.

[35] Stefan Thomke and Donald Reinertsen, "Six Myths of Product Development," *Harvard Business Review* 90 (May 2012): 3-11, via EBSCOhost, accessed September 2016.

[36] Robert G. Cooper and Anita F. Sommer, "From Experience: The Agile-Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity," *Journal of Product Innovation Management* 33 (2016): 513-526, via Wiley Online Library, accessed September 2016.

[37] Cooper and Sommer, "From Experience: The Agile-Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity."

[38] Wheelwright and Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality.

[39] Thomke and Reinertsen, "Six Myths of Product Development."