J PROD INNOV MANAG 2016;33(5):513–526

DOI: 10.1111/jpim.12314

From Experience: The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity

Robert G. Cooper* and Anita F. Sommer*

Agile development methodologies have been widely employed in the software industry, where they have been found to yield positive results. But can these new methods, with their new tools such as sprints, scrums, burndown charts, and backlogs, really be integrated with the traditional and popular Stage-Gate approach and then applied to physical products? Initial but limited evidence suggests yes: Larger IT firms have already integrated Agile and Stage-Gate and gained the benefits of both approaches; and most recently, a handful of manufacturing firms have employed this Agile–Stage-Gate hybrid model for physical new products. And if recent evidence can be trusted, this new approach promises to be the most significant change to our thinking about how new-product development should be done since the introduction of today's popular gating systems 30 years ago. The benefits of this hybrid model are a faster and more adaptive response to changing customer needs, better integration of voice-of-customer, better team communication, improved development productivity, and faster to market. A case example from a toy company, LEGO, is provided as an illustration. But there are negatives as well, and additionally, manufacturers must make modifications to Agile in order to adopt it successfully. Although initial results appear promising, much research is needed to explore this new Agile–Stage-Gate hybrid model, and many research challenges remain.

Introduction

Conventional wisdom is that Agile development methods apply only to software projects. But new evidence reveals that this is not true—that Agile methods can be integrated with traditional gating approaches to yield an Agile–Stage-Gate® hybrid model that can be adapted to manufactured new products.[1] Further, research into very recent industry experience suggests that this hybrid model, which incorporates Agile development methods, has significant potential benefits for manufacturers of physical products, from food and toys to heavy industrial equipment, yielding surprisingly and dramatically positive results.

Agile is a set of methodologies created for the development of new software products; it is based on the Agile Manifesto crafted by IT industry leaders in 2001—a set of rules and guidelines for how to best develop new software code (Beck et al., 2001). But there has been some skepticism as to whether Agile can be generally used for most hardware or physical product development. Development of IT products, after all, is clearly quite different than new-product development in the manufacturing world. So will Agile work in this different manufacturing context?

The evidence is limited, but early results from a few lead users in the manufacturing world suggest "yes"—that Agile can be combined with traditional stage-and-gate or plan-based models and does indeed work very well for manufactured goods. If the recent evidence can be trusted, this new approach promises to be the *most significant change* to our thinking about how new-product development should be done since the introduction of today's popular gating systems 30 years ago! Some of the reported benefits of this new Agile–Stage-Gate hybrid model include:

- Much more responsive to changing customer needs (this is critical when facing fluid markets where things change quickly).
- Builds in voice-of-customer (VoC) in a much more proactive and effective manner than traditional methods.
- Deals with the resourcing issue (for example, dedicated team members) more directly.
- Reduces cycle time and is more productive.

Address correspondence to: Robert G. Cooper, McMaster University, Ontario, Canada. E-mail: robertcooper@cogeco.ca. Tel: 905 845 7273.

*Both authors contributed equally to this article.

[1]Stage-Gate® is a legally registered trademark of R.G. Cooper (and Associates Inc.) in the EU and Canada; and of Stage-Gate International in the USA.

Questions to Investigate

One reason for this paper is to encourage academics to have a hard look at this new approach, which is quickly gaining momentum in the business world (Cooper, 2014). Sadly, with one exception, there has been little or no academic research done in this new area of hybrid Agile–Stage-Gate methods for physical new products. Ironically, the business community, particularly in Europe, and not we academics, seems to be leading the way here.

Some big questions and key research issues:

1. The first issue and research question is: Can Agile be combined with traditional gating processes and work well and symbiotically (or are the two approaches mutually exclusive or just incompatible)?

Many major firms already employ traditional gating approaches for new-product development, and have gained benefits over the years. Understandably, management might be skeptical, and thus reluctant, to throw out the tried-and-proven system for a relatively new and untried approach. Early research evidence suggests that this is true—that management skepticism regarding Agile is an issue (Barlow et al., 2011). But if the two systems work well together and to advantage, then acceptance by management is facilitated: One can keep the existing and familiar Stage-Gate system, and simply build in the new Agile approaches.

But more: Can a hybrid model yield *more benefits* than either model employed on its own? Traditional gating approaches bring a disciplined approach to new-product development by breaking the process into discrete stages from idea to launch (Cooper, 2011). Each stage incorporates proven best practices, such as undertaking VoC, concept testing, doing effective front-end homework, putting a cross-functional project team in place, and so on. And defined deliverables requirements give project teams clear objectives in terms of what is required to move forward to the next stage. Stage-and-gate systems also build in gates, or Go/Kill decision-points, where resources are committed for the next stage of the project and weak projects are weeded out. Gates also enable an incremental investment approach, thereby mitigating risk. So Stage-Gate is very much a *macro-model* (or top-down model), designed to *help select the right projects* (do the right projects), and once selected, to *map out the key stages*, best practice activities, and roles and responsibilities as part of the project (doing the project right).

By contrast, Agile is a *project management method* that brings agility, adaptability, and speed to development projects: It includes *micro-planning tools* for creating software code and get to a working end-product quickly. Agile is designed specifically for managing and supporting *product developers* (technical people) as a way to develop their working software once a development project has been "approved." In practice, Agile development typically consists of a number of very short sprints, where each sprint or iteration, undertaken by a dedicated (full-time) project team, produces an executable IT code or software that actually works, and that can be demonstrated to stakeholders.

So both models work, and they do deliver benefits. But do they *work together*, for example a hybrid model that yields better results than either model can deliver alone—the best of both systems? Or, do they yield the worst of both? Or perhaps they are just incompatible or even mutually exclusive.

2. The second issue and key research question is: Can such a hybrid approach, which incorporates Agile methods, work effectively in the manufacturing world which uses a gating process—for physical product development? And does it deliver positive results there?

Agile, notably the Scrum version of Agile, brings a number of new methods and procedures to software

BIOGRAPHICAL SKETCHES

Dr. Robert G. Cooper is Professor Emeritus at the School of Business, McMaster University, Canada and ISBM Distinguished Research Fellow at the Smeal College of Business Administration, Pennsylvania State University, USA. Cooper is also a Fellow of the PDMA. He is the co-founder of the Product Development Institute Inc. and Stage-Gate International. Dr. Cooper has spent more than 40 years studying the practices and pitfalls of over 3000 new-product projects in hundreds of companies. He pioneered the research into best practices and success drivers in product innovation, which led to ground-breaking discoveries including the Stage-Gate® idea-to-launch system. A noted lecturer and consultant to industry, he has written 13 books and over 120 academic articles on the management of new products. He lives in Oakville, Ontario and Sarasota, Florida. robertcooper@cogeco.ca

Dr. Anita F. Sommer is a senior consultant in continuous improvement in The LEGO Group, and visiting research associate at the Engineering Design Centre, University of Cambridge. Her research focuses on the improvement of product development management, especially on the usage of Agile/Stage-Gate hybrid frameworks, and virtual experimentation and simulation. Since 2001, she has consulted for industrial manufacturing companies, improving development, management, and manufacturing processes. She holds a PhD in product development management (2014) and a master's in Industrial Engineering (2009) from Aalborg University, Denmark. She lives in Billund, Denmark. anita.friis.sommer@lego.com.

Table 1. Where Agile and Plan-Driven Gating Models Fit

Characteristics of Projector Setting	Agile Home Ground	Plan-Driven (Gating System) Home Ground
Criticality of project	Low	Extreme
Developers' experience	Senior (experienced)	More junior
Product requirements	Change often during project	Stable requirements & specs
Project team size	Small	Large
Company culture	Culture that responds to change	Culture that demands order

Source: Boehm & Turner (2004)

development teams. (Although there are many versions of Agile, the Scrum version appears to have the most applicability to physical products). Some attributes of Agile-Scrum include: the use of short time-boxed sprints, daily scrum meetings, burndown charts, sprint backlog lists, and scrum masters (more on these important elements later). In the IT world, development of the product is broken down into a series of time-boxed sprints or iterations; each sprint lasts about 2–4 weeks. And at the end of every short sprint, the project team must produce a working and *potentially releasable version* of the new software product. Note that each sprint is planned in real time, literally in the few days before each sprint begins—there is no "master plan" or critical path plan for the entire Development stage of the project. And a dedicated team, which resides in a dedicated project room, is also a requirement of Agile.

But can these methods and tools *really be used to advantage in physical product development*? For example, if one is developing a new pharmaceutical, a new polymer, a new tractor, or even a new beer, it's hard to imagine how the project team might have a "potentially releasable version" of the new product ready every 2–4 weeks! And dedicated team members working in a project room together, full-time on this one project, is also difficult to picture for most firms, especially when one considers that more than 90% of such physical-product development projects are under-resourced (according to the Cooper, 2013).

So can this new method—Agile–Stage-Gate—really work for physical products, and if so, how? What modifications have been made (or are needed) by manufacturing firms to integrate Agile into their traditional gating processes? And what results are achieved?

The Available Evidence

Issue 1: Are the Two Models Compatible and Symbiotic?

The IT world provides most of the evidence that the two models, namely plan-driven (the stage-and-gate model; and the waterfall model, IT's equivalent to the gating approach) and Agile (plan-on-the-fly), are compatible. Boehm and Turner (2004) discuss the contrasts between plan-driven software development and Agile approaches at length in their book, complete with successful case examples from the IT industry where the two models are integrated. The authors propose a means of *balancing agility and plan-driven approaches* to suit the needs of a given project: The balance depends on looking at the risks of swinging too far in either the plan-driven or the Agile directions. One of the conclusions drawn in the book is that future projects will *need both agility and discipline*, which can be implemented by containing the Agile development methodology within the gate model. They note the types of projects best suited to each approach—Table 1—and since no project is 100% one extreme or the other on all five dimensions, a balanced approach appears to be best.

The argument for adopting an Agile–Stage-Gate hybrid is that both models bring benefits; that there are circumstances where one model works better than the other; and finally, that senior management has proven somewhat skeptical of Agile, and thus is reluctant to discard the tried-and-proven plan-driven model (Barlow et al., 2011).

An investigation based on case studies of software projects, where Agile methods were combined with Stage-Gate, provide successful illustrations of the hybrid model in practice in the IT world (Karlstrom and Runeson, 2005, 2006). The researchers note that "...software development projects are not isolated activities. They usually exist as sub-projects in an environment composed of hardware development, marketing, production planning, etc., which all must be managed and coordinated concurrently." The authors go on: "[stage-gate methods] give support not only for the communication within the project, but also for decision-makers sponsoring the project or acquiring the outcome of the project."

The researchers conclude that "Agile methods give the stage-gate model powerful tools for micro-

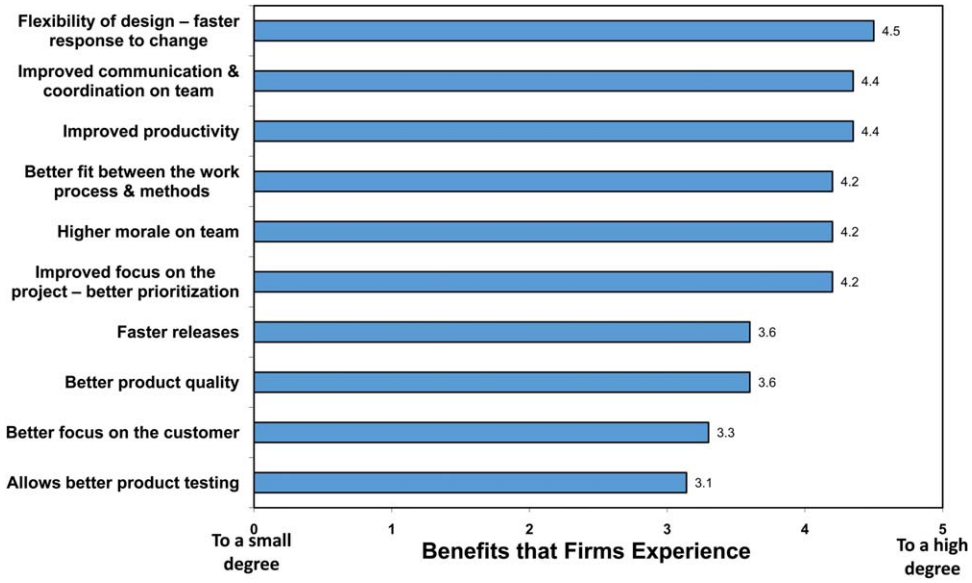

Figure 1. Payoffs from Implementing Agile–Stage-Gate (Mean Scores: 10 Respondents Across 5 Firms)

planning, day-to-day work control, and progress reporting" (Karlstrom and Runeson, 2005). The face-to-face meetings provide more powerful communications than written documents, and the fast and continuous feedback from customers on product features make for a better product and a more efficient project. In turn, the Stage-Gate model provides Agile a means to coordinate with other development teams and communicate with functions such as Marketing, Production, and senior management (e.g., at the gates) and to plan ahead within the project further than the Agile 2–4 week sprints.

Issue 2: Does the Agile–Stage-Gate Hybrid Model Work for Physical Products? Does It Deliver Positive Results?

In manufacturing companies, there is far less evidence that Agile–Stage-Gate hybrid models do work well—there's simply a dearth of research here. One very recent Danish study focused on five manufacturing firms that had traditionally used stage-and-gate systems successfully for many years, and had just introduced Agile to the process (Sommer, Hedegaard, Dukovska-Popovska, and Steger-Jensen, 2014, 2015). The results in all five firms are positive: First, the new hybrid system did indeed work quite well from an operational standpoint, and as evidence, all five firms continue to employ and even expand the use of the new hybrid model. Second, anecdotal evidence and comments during interviews revealed a number of positives as well. Sadly, only a few of the firms had, a priori, installed quantitative metrics that could measure the impact of the new hybrid model, and none of the metrics they used was comparable across firms. Nonetheless the firms' few quantitative metrics revealed:

- improved efficiencies,
- reduced work effort per project—by about 25%, and
- reduced rework—by about 20%.

The main qualitative and common impacts observed by the firms included increased visibility of the process; improved team ownership, independence, motivation and morale; and improved communication both within the team and better coordination across the organization. But from an outside researcher's standpoint, these in-company metrics missed some of the important gauges of performance.

The researchers thus created their own metrics consistent across all five firms, and quantitatively gauged both benefits and challenges of the new model; results are shown in Figures 1 and 2 (Sommer, 2015). Note the strength of responses, albeit from a small sample (10 user respondents, 2 per firm), for the five top benefits: The prime payoff is a faster response to changing product or customer requirements; that is, a much more adaptive and flexible process. Improved team communication is a key benefit, which is consistent with experience in the IT world. Other major payoffs include improved project productivity, a better fit

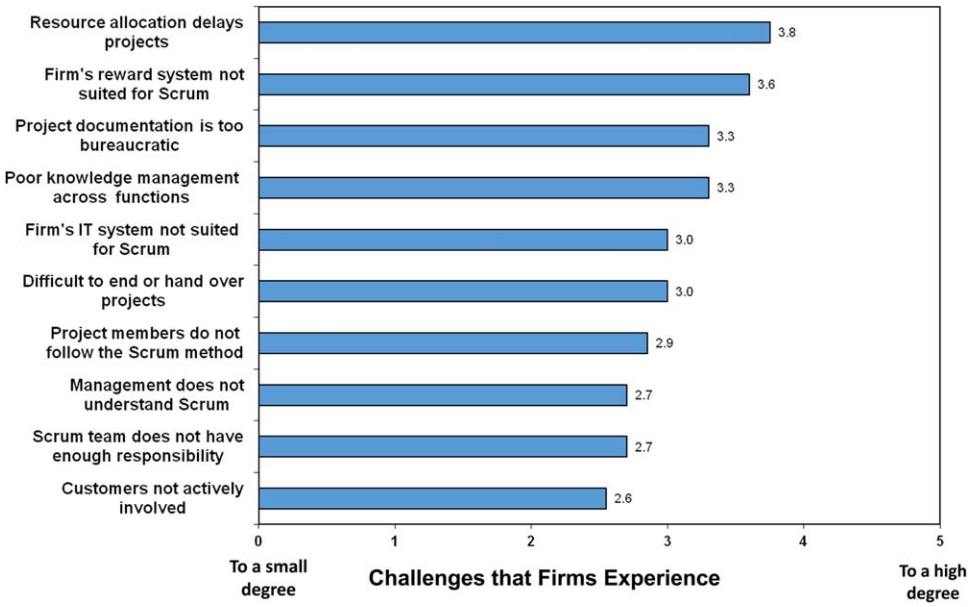

Figure 2. Challenges Faced in Implementing Agile–Stage-Gate (Mean Scores: 10 Respondents Across 5 Firms)

between work process and methods, improved team morale and ownership, and improved focus on the project (better time prioritization).

A few Agile–Stage-Gate case studies were also studied in major global firms, one from the United States (Cooper, 2016). Again, this very limited sample revealed exceptionally positive results, with many of the same benefits shown in Figure 1 being cited by these early adopters: more adaptive (faster response to changing product requirements), better team communication, higher team morale, and most important, faster to market. There are challenges of course in implementing any new system. But the negatives are much more subdued. Figure 2 shows the major challenges as gauged in the five-firm Danish study, and again is consistent with the case studies also cited above (Sommer, 2015).

The evidence regarding the suitability and performance of this new Agile–Stage-Gate hybrid model for manufacturing firms is clearly very limited, but when combined with previous evidence from the IT world, strongly suggests that Agile-Scrum methods can indeed *be built into more traditional gating systems*—they are quite compatible and indeed symbiotic. Further, the evidence suggests that in the case of physical products, Agile–Stage-Gate *appears to give positive results*, and in some cases, exceptional results. Much research is needed, however, notably studies focused on *performance results achieved* that employ some "harder metrics," such as productivity improvements, increase in speed or time-to-market, and success rates of projects entering development. But first, consider some of the details of how Agile works, and hints about why it works.

Some Background and Details on Agile

Agile emerged in the late 1990s in software development; it was developed by IT project managers who needed smarter ways to execute projects. The trend was global, and more than 14 different Agile methodologies or frameworks were proposed and implemented. Finally, in 2001, the Agile masters met in a cabin in Utah to discuss their methodologies and find a common ground. From this meeting came the Agile Manifesto, which today still binds the Agile methodologies together with a joint set of values (Beck et al., 2001). The Agile Manifesto emphasizes:

- individuals and interactions over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation, and
- responding to change instead of following a plan.

These are the core values of Agile. They differ from traditional project management values in that they explicitly challenge the need for documentation and the use of methods for the sake of the methods, rather than being customer centered, even when customers change their minds. However, even though the Agile Manifesto has clearly stated values, it provides

little understanding of why Agile methodologies are so successful, especially in combination with the Stage-Gate model. Clearly, more research is needed!

The Components of Agile

Agile methods consist of many diverse elements. (In practice, not all elements are implemented, depending on the specific Agile version employed.) In the outline that follows, the focus is on the *Scrum version* of Agile, which is the choice of those implementing Agile–Stage-Gate hybrid models for physical products. In 1986, Takeuchi and Nonaka called for a "rugby approach" to managing new product development, where the team meets for scrummage (much like a huddle in North American football) to decide their next moves (Takeuchi and Nonaka, 1986). This is the reason why one of the most popular Agile frameworks was called "Scrum."

In practice, nine Agile-Scrum elements are relevant to Agile–Stage-Gate hybrids—three roles, three artefacts and three tools:

- Artefacts:
  1. Time-boxed sprints
  2. Daily scrums
  3. Retrospective (review) meetings
- Tools:
  4. Product backlogs and sprint backlogs
  5. The scrum board
  6. Burndown charts
- Roles:
  7. Product owner
  8. Scrum master
  9. Development team.

The Agile roles give new sense of ownership, increased motivation, and enhanced communication and knowledge sharing. The Agile artefacts create the *project heartbeat*, and incorporate change management and learning cycles. Finally, the novel tools provide project visibility, showing what needs to be done, what is being done, and progress according to the sprint plan.

Artefacts: Sprints, Daily Scrums, and Retrospective Reviews

The Agile software development project is divided into a series of *time-boxed short sprints* of about 2–4 weeks in length. At the beginning of each sprint, the development team holds a *sprint planning session*: They define what they will achieve in that sprint (sprint goals), develop the sprint backlog (a list of product features or product increments to be developed in the next 2–4 weeks), and map out how they will achieve their goal. In short, they develop or replenish their "to do" queue.

Note that sprints are time-boxed—one cannot exceed the specified time—so there is some time pressure here. Once the sprint gets underway, daily stand-up meetings, called *daily scrums*, are held, where the entire team gets together for 15 minutes at the beginning of each day; each team member outlines:

1. What did they do since the previous daily scrum (in the previous 24 hours).
2. What they will do in the next 24 hours.
3. Which problems they have.

Thus, the entire team receives a quick update on the project status, and team members help each other to achieve their common goal.

The result of each sprint is a tangible result, namely *working software code* that is a potentially releasable product (although it usually takes several sprints to create a real product release). When the sprint ends or is "done," the team demonstrates their working code to the customer (or to an internal stakeholder who is acting as the customer), and seeks feedback and validation; if feedback is negative, a change in requirements, and even a change in direction, may be necessary. Thus, constant customer input and validation is built into the system.

Finally, at the end of the sprint or iteration (after the 2–4 weeks), the project team holds a *retrospective meeting* to evaluate the sprint results, set goals for the next sprint, and recommend needed improvements for the team. This internal, personal and ongoing self-assessment is unique to Agile (traditional approaches normally focus only on the result delivered). Then, the next sprint starts with a new sprint planning session and the entire cycle starts again. And so the team moves quickly down the rugby field, sprint by sprint, scrum by scrum, until a goal is scored.

The very core of being agile is adapting to change. In Agile, change is introduced actively into the project, including changes to the product specifications, but in a controlled way and without disrupting the pace or rhythm of the project team. Change introduction is limited to the beginning of each new sprint, so that work in the subsequent 2–4 week period can be done without disruption. This iterative introduction of appropriate changes thus creates maximum customer

value (the process is highly adaptive and responsive to changing customer requirements), yet at the same time, the work is done in controlled and well-planned time-boxed sprints (highly productive periods, where product specifications are kept fixed).

These sprints thus generate a "project heartbeat" for working productively and creatively (Wells, 2009). A *steady* heartbeat has a fixed length iteration, as short as can be. A *strong* heartbeat produces working software ready for deployment. A *responsive* heartbeat creates a new plan for each iteration, based on feedback from the previous iteration and changing customer needs, not just based on what was left unfinished. This measurable, predictable, sustainable, and constant pace or rhythm helps the team plan and meet its commitments (Wells, 2009).

This analogy of a *project heartbeat* suits the process well (see Figure 3): Each short sprint, with a defined plan, permits work to proceed without interference or change, but also without planning activities too far into the future (plans that would invariably get changed). Changes to product requirements are introduced to the project much like oxygen into the body; and the heart is the Agile methodology itself, pumping at a regular pace and keeping the project alive and revitalized. So rhythm and a steady pace or heartbeat are keys to the team's success, much like *takt time* and the steady rhythm in the world of Lean Manufacturing.

Visual Project Management

Three *visual tools* enable effective prioritization, visual management, and adaption to change. The main tool is the *product backlog*, which is a prioritized list of features that the final product should have. This product backlog is prioritized by the product owner (who represents management and the customer) together with customers and internal stakeholders. The initial *product backlog* contains only the known and well-understood product requirements at the project's beginning, but it is dynamic and evolves in successive sprints as insights into customers' needs grow and the product evolves (ScrumInc, 2013).

Each sprint also has a backlog: The *sprint backlog* is the set of product backlog items selected by the project team for the current sprint, plus a plan for delivering the product increment and achieving the sprint goal. The highest priority features are included in the sprint's backlog, up to the limit of what can be reasonably accomplished within the 2–4 week sprint.

The main advantage of these backlogs is the ability to adapt to change while providing a visual plan for product development. As noted above, the *sprint backlog* may only be changed at the beginning of a sprint, but short-duration sprints still allow great flexibility within the project.

When the sprint is begun, the backlog is broken down into a set of activities by the development team. These activities are shown on a s*crum board* (either

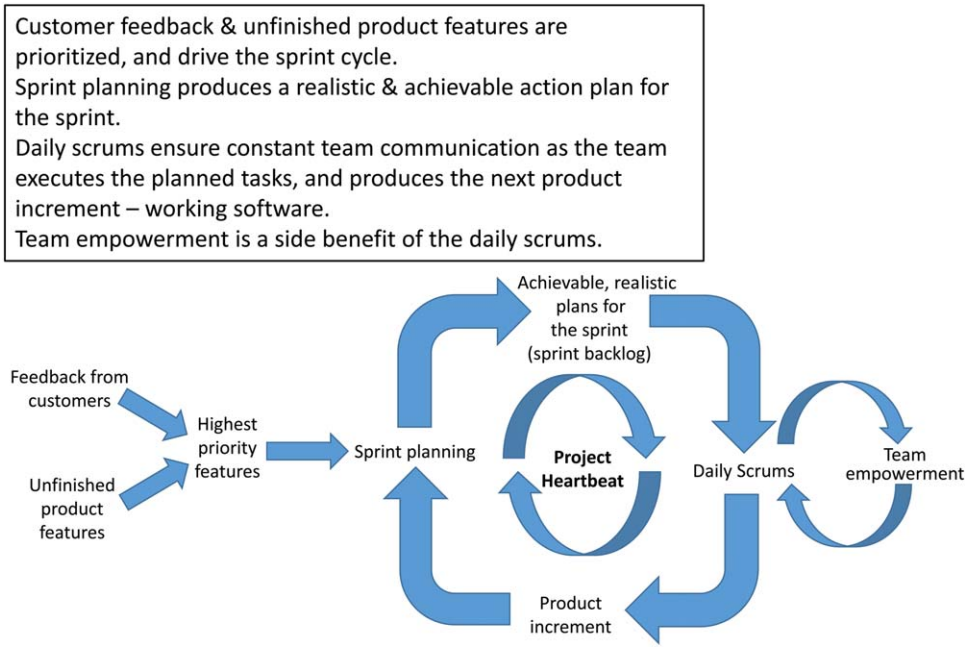

Figure 3. The Heartbeat of an Agile Project (adapted from Wells, 2009)

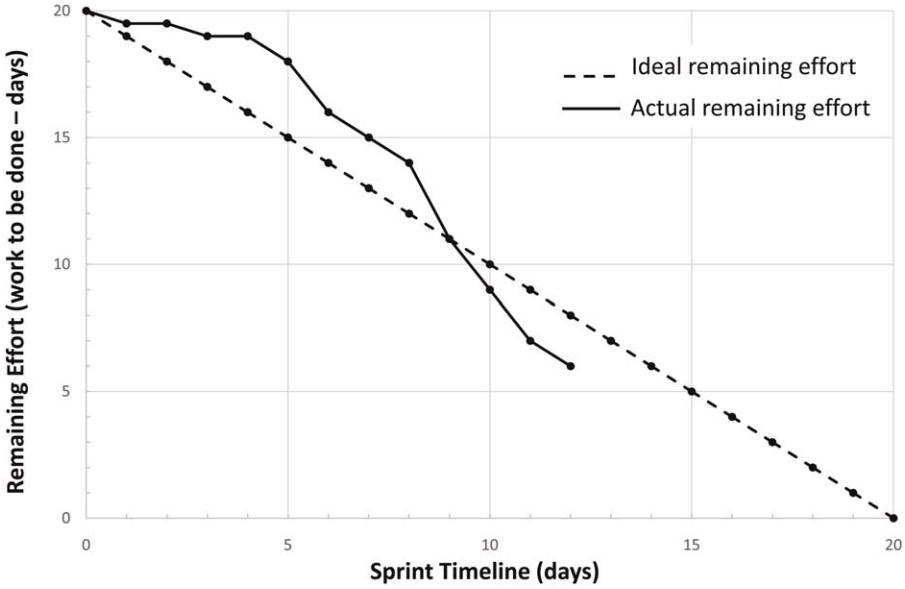

Figure 4. Burndown Chart for a Single 4-Week Sprint, Showing the Work-Effort (Days) Yet to Be Done in the Sprint (*Y*-axis) Versus the Number of Work-Days Available in the Sprint (X-Axis)

physically or virtually), which displays the status of activities (either to-do, doing, done, on-hold, or postponed), and who is currently undertaking each activity. In this way, *product owners* (or any executive) gain a complete overview of all teams' current status on activities within each sprint.

The scrum board also displays the *burndown chart*, which compares expected progress against actual progress for tasks in the current sprint. As illustrated in Figure 4, the *burndown chart* is an X-Y graphical display of work to be executed in the time remaining in the sprint. The vertical or the Y-axis shows the work that remains to be completed in that sprint (person-days needed), and the X-axis, the work-days still available in the sprint. As long as the number of person-days of work remaining-to-be-done is below the base or "ideal" dashed straight line, the sprint is on schedule. But when activities are not finished on time, then the work remaining (the solid line) starts to drift above the ideal baseline, indicating that the sprint is running behind schedule. Thus the team can instantly recognize if activities are behind schedule or not finished at a regular pace, and take steps to increase productivity or flow.

Agile Roles

In traditional product development, a number of different team leadership models and roles are common. For example, some firms use team leaders, who are entrepreneurial and really lead the project team, liaising with management, much like the captain of a football team. There is also the project manager role, a more technical role, dealing with timelines, budgets, and team meetings—the types of things that the PMO or project management office is concerned with. And some firms use both for the same, often larger project—project leaders and project managers!

Agile introduces somewhat different and new roles into new-product development: the product owner and the scrum master. The product owner is responsible for the project, for managing project stakeholders, and for keeping the project on track. Thus he or she is much like the project leader in the traditional gating model. In Agile, however, the product owner is not responsible for the project team's daily activities. This responsibility now falls on the process master for the project, called the scrum master. The scrum master's role is to ensure that the development team follows the prescribed Agile methods, using the tools appropriately, and also to remove any obstacles hindering the team.

This role division—product owner and scrum master—changes communication paths within the project, enabling increased knowledge sharing. Since the project leader or manager is no longer the focal point and daily decision-maker, the team thus must share knowledge and delegate tasks by themselves in order to

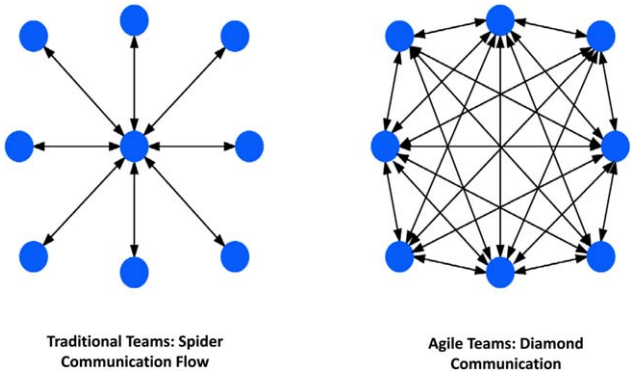

Figure 5. The Diamond Pattern of Communication in Agile Teams Fosters Much Better Within-Team Communication During the Project

make day-to-day decisions. This could be a recipe for anarchy if it were not for the scrum master, who keeps the team focused on executing activities and also on sharing knowledge (e.g., via daily scrum meetings). This shift in decision-making empowers the team, and gives an increased sense of project ownership and engagement in the project.

The improved communication and knowledge sharing is illustrated by the spider versus diamond diagrams in Figure 5. This figure highlights how having a core project manager limits the amount of communication between project members, and how the removal of this role and the resulting creation of many more communication paths enables increased knowledge sharing and team learning. The belief by Agile proponents is that when the responsibilities are appropriately divided among the product owner, scrum master, and development team, the project has a better chance of success than other, traditionally organized projects.

Integrating Agile into Stage-Gate for Physical Products

So far, Agile and Scrum have been largely described for the world of software development, where Agile has been widely adopted. The potential benefits for physical product development are considerable, however, as noted above from the limited research undertaken. But so are the challenges—it is not simply a matter of applying IT's Agile method directly to the manufacturing world.

When Agile and Stage-Gate are integrated, and then utilized for manufactured or physical product development (hardware), Agile is normally applied as the *project management method* within the stages in Stage-Gate. That is, the stages remain, and Agile is applied *within some stages*. The Scrum method is the particular Agile method that seems most appropriate for hardware development, and indeed, is the method used for *all the hardware case studies* uncovered so far in industry.

Next, most manufacturing firms utilize the Agile–Stage-Gate Scrum method *for only the very technical (development) phases* of the project—for example, for the Development and Testing stages in a typical stage-and-gate model. There is no reason why Agile–Stage-Gate could not be used for earlier stages (e.g., for the concept, feasibility, or building-the-business-case stages) or even for the launch stage. Indeed, some more advanced firms do use Agile for the early stages as well as Development and Testing; but there are more challenges here, and more adjustments are required for these earlier stages (e.g., defining a "done sprint" and securing dedicated resources in the concept stage or business case stage).

Here's an example of how Agile-Scrum was integrated into Stage-Gate in a fairly advanced manufacturing firm in Denmark, namely the LEGO Group, and the results achieved:

LEGO Education is a business unit in the LEGO group, responsible for development of software and physical products for educational purposes for children in pre-school and higher. In 2011, LEGO Education initiated a highly innovative new-product project, *Story Starter*—an educational solution whose purpose was to create confident writers and readers in elementary schools. Due to the introduction of digital components into the product, management tried a totally new-to-LEGO hybrid development method, which turned out to have significant positive advantages to the project, and ultimately resulted in the launch of this highly successful and innovative new product just 12 months later!

LEGO management saw this *Story Starter* initiative as a radically new business opportunity; but the journey was explorative and the process complex, involving customer trials from early in the project. Initially, the project was managed using LEGO's tried-and-proven but traditional stage-and-gate model. This model, however, did not fit well with the design iterations required by intensive customer interaction; but the team had no alternative, and so had to work with the traditional gating model available to them.

By pure chance, the team was offered a unique alternative that both sped up the customer iterations and improved overall project productivity. It was

decided to incorporate a digital documentation tool in order to document children's learning experiences. The Digital Solutions group at LEGO were already using Agile methods for software development; thus when project team members from Digital Solutions joined the *Story Starter* team, they brought Agile with them.

After seeing the advantages of Agile for software development, the *Story Starter* team decided to try this new approach for *managing the entire project*. Management did not want to lose the strategic benefits of their stage-and-gate model, however, so they chose to keep it while implementing Agile within the gating system. As a consequence of this merger, an Agile–Stage-Gate hybrid emerged which was applicable for software as well as physical components.

The *Story Starter* development team began using this new Agile–Stage-Gate hybrid model complete with sprints, 15-minute daily stand-up meetings (scrums), visual scrum boards, daily activity logs (the requirement for burn-down charts), a prioritized project backlog, and sprint planning meetings. The Agile method was incorporated into the project during the development phase and continued into the implementation phase in LEGO's gating model. Throughout the development phase of the project, teachers from more than 50 schools participated in product trials. Sprint reviews were conducted using internal representatives for customers, and pilot studies involving the primary customers (schools) and tests with children were conducted regularly. The number of active team members changed during the project depending on needed expertise, but remained between five and ten throughout; however, they were not dedicated to this one project.

After the implementation of this new Agile–Stage-Gate hybrid model, the *Story Starter* project experienced a remarkable acceleration, and within a short period of time, it was doing much better than previously undertaken projects. The most dramatic change for the team was the immediate *increase in productivity*. In hindsight, team members indicated that the productivity increase was caused mainly by much better team communication and a decrease in misunderstandings. Also, team members experienced improved work-flow each day, partly due to feeling secure in the knowledge that their current problems would be readily solved at the next morning's scrum meeting. Finally, Agile–Stage-Gate helped the team to remain in excellent communication, despite frequent travels and unforeseen activities and tasks.

Following the successful piloting of Agile–Stage-Gate with *Story Starter*, a second and large new-product project, *More to Math*, was similarly developed using the new hybrid model, and also proved to be a winner (*More to Math* is an elementary-level teaching solution for mathematical problem solving).

This new and experimental Agile–Stage-Gate method, initially piloted because an alternative project management approach was needed for radical and highly uncertain projects, has proved so effective that an Agile–Stage-Gate standard is currently being developed for widespread use in the business. LEGO Education's new-product releases for 2016 are all being developed using this novel hybrid approach.

Adjustments Needed for Physical Products

Additional adjustments to the Agile-Scrum model have been implemented by some hardware product firms as they experiment with Agile–Stage-Gate hybrid models.

Defining a "Done Sprint"

The concept of a done sprint is very important in Agile, as it is something tangible and concrete by which to measure progress, one of the key principles in the Agile Manifesto. With Agile and software development, each sprint or iteration produces a *working product* (executable software) that can be demonstrated to stakeholders (customers), as noted above. Furthermore the working product at the end of each sprint is potentially releasable to the market. When contrasted to hardware development, the difference is that software development—writing thousands of lines of code—is *infinitely divisible*; but hardware development is not usually *not divisible*. That is, one usually cannot build *part of the product* that actually works; thus it is normally not possible to have anything that actually functions developed within a few weeks. For example if your product is a lab instrument or a diesel engine, one cannot build "part of the instrument" or "part of the engine" and demonstrate it working!

If the end of a sprint does not produce a working and potentially releasable product, then does the concept of a done sprint still apply? An important adjustment that manufacturers thus must make is *to redefine what is meant by a done sprint*. Here, an iteration or spiral *does not build a working product*, but a *product version* somewhere between a "product concept" (usually words and pictures) through to a "ready-to-trial

prototype." Think of this in-between product version as a "protocept"—something to show the customer to seek feedback, something to test a market-facing hypothesis. These product versions or *protocepts* can be computer-generated 3D drawings, virtual prototypes, crude models, working models, rapid prototypes, or early prototypes; that is, something physical that the customer can respond to. Such *protocepts* are used not only to seek customer feedback and validation often, early, and cheaply, but they also reduce technical uncertainties, since they can be used to demonstrate preliminary technical "proof of concept" early in the Development phase—an envisioned technical solution. Note that definitions of a done sprint tend to vary by company; here are some examples:

- A United States-based remote-control systems manufacturer: A done sprint is "something physical, the result of completed tasks in that sprint (and not just a PowerPoint show). This could be, for example, a set of completed design drawings, or a rapid prototype, or an early working model of the product."
- A Swedish construction equipment manufacturer: A done sprint is "results of work done, documented as an A4 (12 × 17 inch) templated report, easy to read and easy to post on a wall, reviewed by an expert colleague, and checked into the document repository."
- LEGO: A done sprint is when a ten-element checklist (the "Definition of Done") is completed by all team members, as well as by Quality Assurance and by the product owner, declaring that all product acceptance criteria have been met, and that all open actions have been completed, handed over, or closed.

Resource Allocation

Another adjustment that may be necessary is the requirement for a *dedicated, co-located project team*. Dedicated teams are certainly not the norm in physical product development; in fact quite the reverse is true! For one thing, there are often "waiting times" in such projects—for example, waiting for field trial results, or waiting for equipment delivery; the result is that, in the meantime, project team members can then work on other projects (these waiting times are usually not part of a typical IT project). Additionally, there are so many projects underway in most manufacturing firms that finding dedicated team members is almost impossible.

In an Agile IT project, however, the development team is a small group of people, largely technical (code-writers) working in a single project room, with every day devoted to the one same project! But physical product development demands the use of a *cross-functional team*, which means people from Marketing, Operations, and even Sales working on the project. Securing 100% dedicated resources from these functions, and collocating them in the same room with the technical team members may be a bit of a challenge for many firms. Compromises are often made, such as having the Marketing people dedicated for a single sprint only, for example, while the VoC work is being done; or even limiting the Agile facet of the project to only the Development and Testing stages of the project, which is mostly technical people anyway. Another solution is that employed by some firms, such as Corning: Only the major and critical projects go through the new Agile system—about 20% of projects—and have dedicated teams. But the majority of Corning's new-product development projects still have team members spread across multiple projects (Cooper, 2013).

There is no question that having a dedicated project team is a major benefit of the Scrum method, and this fact alone may explain many of the gains made in speed to market. It would be indeed unfortunate to substantially relax this "dedicated team" requirement of Scrum in the case of physical product development.

Integrating Traditional-Model Planning into Sprint Planning

Two often-asked questions by manufacturers which use Stage-Gate (and are thinking about integrating Agile) include:

1. Isn't there an apparent contradiction between the longer-term plan that is required and approved in the gating model (a plan-based model) versus the short-term plans developed "on the fly" at the beginning of each sprint in Agile?
2. And how can one ensure that the many backlogs developed for sprints are consistent with the product definition approved at the gate?

Consider the simpler case of a manufacturing firm using Agile–Stage-Gate for just the Development and Testing stages. The project team delivers a business case to the "Go to Development" gate in order to seek approval of the project for the next stage, namely Development. In the traditional gating model, a business case includes a financial and business analysis, a

detailed product definition, and a "go-forward" or development plan for the entire next stage, namely Development, complete with expected development costs and times. Senior management reviews the business case and approves the project for development (including approval of the development plan), and commits major resources for development.

In Agile–Stage-Gate for hardware products, however, the business case must be somewhat modified, for example, as follows:

- The financial and business analysis, the same as in the traditional gating model.
- The product definition—but this time with *fixed and variable elements*. That is, not all product requirements are known or fixed at the point of project approval. Indeed, many product requirements are still unknown and fluid—for example, in Hewlett Packard's Agile system, the physical product is only 40-to-70% defined on entering the Development phase (MacCormack, Crandall, Henderson, Toft, 2012).
- The development plan; but instead of the detailed and fixed development plan, a *higher level, tentative plan*, namely the *development backlog* is delivered. This backlog for the next stage is more than just a *sprint backlog;* it is for the entire Development stage. And besides outlining the product and its elements or features, this *development backlog* also lists *the tasks to be done* and the *information gaps to be filled* in the next stage. Note that this plan is somewhat tentative—the development backlog could change once the sprints are underway—and it is not very granular (quite high level). Thus expected costs and times for Development can change from those approved at the Go to Development gate.

Once the project team starts the first sprint, their sprint planning meeting *identifies those development backlog* items *that they can realistically* accomplish during that first sprint; then they create their *sprint backlog* (again product elements, tasks, and information gaps to be filled). They then move into the sprint, tracking *progress on the sprint burndown chart*. And so the rhythm starts, as they continue through the Development stage, sprint by sprint, planning each sprint on the fly and in real time. Thus both the product and the development plan evolve over time.

When to Use Agile–Stage-Gate

One thorny question concerns whether or not Agile–Stage-Gate is suitable for both *radical* and for *incremental* new-product developments? In theory, it is suitable for both, but Agile–Stage-Gate *adds the most value* when there is high uncertainty and a great need for experimentation and failing fast (seeking VoC inputs early). That is, for more radical new-product projects, Agile–Stage-Gate offers the additional benefits of managing high uncertainty through incremental product versions (protocepts), quick learning cycles, and frequent customer involvement. For example, one large U.S. manufacturer of residential remote-controlled devices reserves its Agile–Stage-Gate system for about 20% of development projects, namely only the larger, major revenue-generating, higher risk projects. Another firm, a noted global manufacturer of process controls, has created *three product development models*. Besides their traditional stage-and-gate system, they are currently deploying two more: Agile–Stage-Gate and also Iterative Stage-Gate (not quite Agile with scrums and sprints; the iterations or spirals last about 8–12 weeks), but this firm so far *allows the project leader and team to choose the model* most suited to their project. Finally, Hewlett-Packard's various product development models have guidelines to help the project team elect the right one, with the Agile version recommended as best for projects with rapidly changing markets and needs and lower maturity technologies with more unknowns (see Table 2).

Many Unanswered Questions and Research Challenges

The prospect of a new-and-proven approach to new-product development is exciting. And early research evidence suggests that this Agile–Stage-Gate approach may have dramatic and positive impacts on new-product development. While many fads have come and gone over the years since Stage-Gate was first introduced, few had much empirical evidence to support them. This new approach does!

Even more exciting are the research prospects and opportunities for us academics, due to the many and provocative research questions that remain unanswered:

1. Does this new Agile–Stage-Gate hybrid method really deliver positive results in terms of higher success rates, better new-product development productivity, faster to market, and meeting sales and profit objectives? Note that much of the research to date has been anecdotal and case-study based; what

Table 2. Two of Hewlett Packard's Models for Physical New-Product Development, and When Each Is Most Appropriate

Project Type	Agile	Efficient (Traditional)
The situation	Exciting & rapidly growing but changing market	Mature market Well known market, customers, and needs Well-known technology
Customers	Early majority in "adoption curve"	Late majority & laggards
Customer needs	Known, but changing rapidly	Well-known & stable over time
Technology maturity	Maturing quickly; following a well-defined trajectory	Proven and stable, predictable
The approach	Agile approach: Evolutionary process based on frequent design-build-test iterations, milestone releases, & beta versions with actual customers; continually re-prioritize features	Traditional but efficient phase-gate process; well-defined staged, gated process with clear entry/exit criteria, explicit tasks and deliverables, & rigorous checkpoint review meetings; monitor to plan
Objectives	Rapidly evolve the product design to meet changing customer needs & technical choices	Improve only those features valued by customers Reduce product costs
Product specifications	Established in general up front; updated as feedback on performance & changing customer needs.	Established in detail up front; based on well-understood customer requirements & technical solutions
Product definition on entering development	40-70% of final design parameters defined	>90% of final design parameters defined

Source: Adapted from: MacCormack et al. (2012).

is needed are larger sample studies with more quantitative and operational performance metrics.

2. If so, *why* are the results better? What specifically is so unique about Agile–Stage-Gate that delivers positive performance results? Reasons why Agile–Stage-Gate hybrid models work so well are summarized by Cooper (2016), but these reasons are largely opinion-based or hypotheses and remain to be empirically validated. For example, a skeptic might argue that "faster to market" is the direct result of having a dedicated project team, and not so much about the Scrum method per se; and indeed, a dedicated team using a traditional gating model would achieve the same fast-to-market results. Similar questions could be posed about all the hypothesized reasons why the hybrid model appears to work so well.
3. What are some of the challenges and weaknesses of this new hybrid approach? Can it all be so positive? Sommer identified some deficiencies (in Figure 2, but not previously published), but obviously a larger sample and more quantitative measurement is needed. And perhaps senior managements' views should also be sought. For example, in one U.S. firm, senior management expressed concerns that project teams, so heavily focused on the next 2–3 week sprint, had lost sight of the ultimate goal, thus requiring senior management intervention (Cooper, 2016).

   Whenever a new method is introduced, often the proponents become strong advocates, and with religious zeal, deny or refute any weaknesses or faults in the new approach. Such is the case with some Agile adherents. Indeed, one of the attendees at the Agile Manifesto 10-year reunion in 2011 stated that the Agile movement was much like a teenager: "very self-conscious," "accepting few criticisms," and "at times cocky and arrogant." So we need to hear more about the "elephants in the room" (Kruchten, 2011), and what firms are doing to overcome these fail-points.
4. What additional adjustments must be made to the Agile method to integrate it with Stage-Gate and then to apply the hybrid model to physical products? For example:

   - How does one define a done sprint? This is an important concept in Scrum, but the definition is not readily obvious in Agile–Stage-Gate, especially in the fuzzy front end of projects. For example; the definitions of a done sprint outlined

above would not work so well in the concept or build-business-case stage of a project.

- How does one integrate the planning process of a traditional gating model with the sprint planning model of Agile (plan on the fly). For example, when one approves a project for development, one normally approves the resources needed, and thus approves the full Development Plan. But this plan may not be fully crafted prior to the start of Development in the Agile model. Some suggestions were offered above—for example, creating a *development backlog* for the entire Development stage, or roughing out a high-level tentative development plan—but even here there are some apparent inconsistencies and challenges. More research and probing is needed.
- What about the spirals or iterations? What does one show the customer for validation at the end of each 2-4 week sprint? Is it an MVP (a minimally viable product), which can be sold commercially (as in the Lean Start Up method; Ries, 2011); or it is more a representation or model of the product, somewhere between concept and full prototype (a *protocept*)? What approach works best here, and when?

5. What should be in a backlog? What does a backlog look like in the hybrid model? In Agile for IT development, backlogs contain the features of the software product to be completed in that sprint. But some manufacturing firms were struggling with this narrow definition, and were including "knowledge gaps to be filled" and "tasks to be done" in their backlog lists (Cooper, 2016).
6. And what about roles and responsibilities? Agile brings some new terms and introduces new roles such as *scrum master* and *product owner*. How do these new roles fit in with traditional concepts of program manager, project manager and project leader?

These and other research questions provide a rich and exciting landscape full of research potential, and with potentially high pay-offs for industry.

References

Barlow, J., J. Giboney, M. Keith, D. Wilson, R. Schuetzler, P. Lowry, and A. Vance, A. 2011. Overview and guidance on Agile development in large organizations. *Communications of the Association for Information Systems* 29 (1): 25–44.

Beck, K., M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. 2001. Manifesto for Agile software development. Agile Alliance. Available at http://agilemanifesto.org/.

Boehm, B., and R. Turner. 2004. *Balancing agility and discipline: A guide for the perplexed*. Boston, MA: Addison-Wesley.

Cooper, R. G. 2011. *Winning at new products: Creating value through innovation* (4th ed.). New York: Basic Books.

Cooper, R. G. 2013. New products—What separates the winners from the losers and what drives success. In *PDMA handbook of new product development* (3rd ed.), ed. K. B. Kahn, 25–33. Hoboken, NJ: Wiley.

Cooper, R. G. 2014. What's next? After Stage-Gate. *Research-Technology Management* 157 (1): 20–31.

Cooper, R. G. 2016. Agile-Stage-Gate hybrids: The next stage for product development. *Research-Technology Management* 159 (1): 21–29.

Karlstrom, D. and P. Runeson. 2005. Combining Agile methods with Stage-Gate project management. *IEEE Software* 22 (3):43–49.

Karlstrom, D. and P. Runeson. 2006. Integrating Agile software development into Stage-Gate managed product development. *Empirical Software Engineering* 11: 203–25.

Kruchten, P. 2011. Agile's teenage crisis. *InfoQueue*. Available at: http://www.infoq.com/articles/agile-teenage-crisis.

MacCormack, A., W. Crandall, P. Henderson, and P. Toft. 2012. Do you need a new-product development strategy? *Research-Technology Management* 155 (1): 34–43.

Ries, E. 2011. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Publishing Group.

ScrumInc. 2013. *The Scrum Guide*. Available at: http://www.scrum-guides.org/scrum-guide.html.

Sommer, A. F., C. Hedegaard, I. Dukovska-Popovska, and K. Steger-Jensen. 2014. Agile product development governance—On governing the emerging Scrum/Stage-Gate hybrids. In Advances in production management systems. Innovative and knowledge-based production management in a global-local world, ed. B. Grabot, B. Vallespir, S. Gomes, A. Bouras, D. Kiritsis, 184–91. Berlin:Springer-Verlag.

Sommer, A. F., C. Hedegaard, I. Dukovska-Popovska, and K. Steger-Jensen. 2015. Improved product development performance through Agile/Stage-Gate hybrids–The next-generation Stage-Gate process? *Research-Technology Management* 158 (1): 1–10.

Sommer, A. F. 2015. Unpublished research results from research reported in Sommer et al., 2015.

Takeuchi H., and I. Nonaka. 1986. The new product development game. *Harvard Business Review* 64 (1): 137–46.

Wells, D. 2009. A project heartbeat. The Agile Process. Available at: http://www.agile-process.org/heartbeat.html.