Research-Technology Management

ISSN: 0895-6308 (Print) 1930-0166 (Online) Journal homepage: http://www.tandfonline.com/loi/urtm20

Agile–Stage-Gate Hybrids

Robert G. Cooper

To cite this article: Robert G. Cooper (2016) Agile–Stage-Gate Hybrids, Research-Technology Management, 59:1, 21-29

To link to this article: http://dx.doi.org/10.1080/08956308.2016.1117317

Agile–Stage-Gate Hybrids

The Next Stage for Product Development

Blending Agile and Stage-Gate methods can provide flexibility, speed, and improved communication in new-product development.

Robert G. Cooper

OVERVIEW: Leading firms are now beginning to integrate elements of the Agile IT product development method into their traditional gating processes to develop physical products. The trend began first in the IT industry, where Agile and Stage-Gate methods were found to complement each other, and only recently has been seen in manufacturing firms. The benefits of the hybrid model include much faster product releases, better response to changing customer requirements, and improved team communication and morale. But some modifications to the Agile model are required for physical products. Two large-company best-practice examples are provided to illustrate how to run a hybrid model.

KEYWORDS: Agile, New product development processes, Hybrid models

Some months ago, I facilitated a heated meeting between software and hardware developers in a large US instrument firm whose products included both hardware and software components. The question was, for the development of the software component, can or should Agile development methods—developed for software projects—and Stage-Gate[1]—developed for hardware projects—be used together or only separately? But more: can or should hardware developers employ aspects of Agile, for example, the sprints and scrums that are central to the Agile-Scrum method? In other words, are the two approaches complementary or mutually exclusive? Can Agile be integrated with a traditional stage-and-gate model? And can the resulting hybrid model also be used for the development of physical products?

Agile was created in response to the particular problems facing software developers; in this context, its relevance in instances where a firm's products include both hardware and software and the two development efforts must be integrated, is clear. In these cases, a hybrid Agile–Stage-Gate approach can both respond to the specific needs of each component of the product and help integrate the two efforts.

Moreover, Agile methods promised to improve speed to market and increase development productivity, something that all hardware developers strive for. As they face increasingly fluid markets, where nothing is stable for long, manufacturers have also begun looking at development methods that are more adaptive, allowing for faster response to changing customer requirements. Some manufacturers, struggling with these challenges, found Agile quite attractive. But Agile alone isn't sufficient to support new product development for manufacturers. As a result, some manufacturers of products from food to machinery are turning to hybrid development processes that integrate Agile with Stage-Gate, even when no IT development is involved. And some of these early adopters are finding that the benefits of adopting a hybrid Agile–Stage-Gate approach can be significant.

The Evolution of Agile

Agile software development is a group of software development methodologies based on iterative and incremental process in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams (Beck et al. 2001). When Agile emerged in the late 1990s and early 2000s, its methods were seen as the solution to many problems in IT development that traditional waterfall or gating development processes could not deal with (Reagan 2012).

[1] Stage-Gate is the registered trademark of R. G. Cooper and Associates, Inc., in Europe and Canada and of Stage-Gate International in the United States.

Robert G. Cooper is president of the Product Development Institute. He is ISBM Distinguished Research Fellow at Penn State University's Smeal College of Business Administration and a Crawford Fellow of the Product Development and Management Association. A thought leader in the field of product innovation management and developer of the Stage-Gate new product development system, he has won two IRI Maurice Holland awards and has published 7 books and more than 120 articles. He received his PhD in business administration from the University of Western Ontario and Bachelor's and Master's degrees in chemical engineering from McGill University. robertcooper@cogeco.ca

DOI: 10.1080/08956308.2016.1117317

Stage-Gate is a macroplanning process and Agile is a microplanning project management methodology.

These traditional processes tend to focus on a big, long-term goal—a final product and its major features. But requirements tend to change rapidly in IT projects; the features and criteria defined when the project was initially planned often were no longer valid by the end of a 12- to 18-month development cycle. And, as Reagan (2012) puts it, "it's hard to alter course when you're being swept down a large waterfall . . . Too much up-front planning means too much change management downstream" (Slide 2). Committing early to features and schedule means that compromises will be needed late in the game; early commitments to large features, long schedules, long feedback loops, and the replanning inherent to traditional product development processes create inefficiencies and slow the development cycle.

Agile was introduced in the IT world to deal with these issues through adaptive planning, evolutionary delivery, a time-boxed iterative approach, and flexible response to change. Beck and colleagues (2001) coined the term Agile in their "Manifesto for Agile Software Development," which called for emphasis on individuals over processes, working software over complete documentation, collaboration over contracts, and flexibility over planning; they elaborated a set of 12 supporting principles, among them an insistence that (1) working software be delivered quickly and iterated frequently (in cycles of weeks rather than months), and that (2) working software be the principal measure of progress (Beck et al. 2001).

Agile vs. Stage-Gate

Boehm and Turner (2004) aptly summarize the differences between plan-driven software development (based on gated or waterfall models) and Agile approaches: gate models, they explain, are generally "plan-driven models," whereas Agile is more "plan and build on the fly."

The differences emerge from the two systems' different intents—Stage-Gate is a comprehensive idea-to-launch system and a macroplanning process, and Agile is a microplanning project management methodology (Table 1). Stage-Gate is cross-functional (that is, involving people from marketing, sales, and operations alongside technical personnel) and it has multiple stages spanning the entire idea-to-launch chain, from idea generation through the business case and market launch (Cooper 2011, 83–116). It is also a guide to action, building in specific best practices at each stage—doing voice of the customer work, building a robust business case, designing an effective launch, and so on. In this way, it's more like a football playbook than a project management approach. The decisions in Stage-Gate follow an investment decision model; a go decision at a gate commits the resources for the next stage, so that resources are funneled to the best projects as their potential emerges. Stage-Gate thus provides guidance for what projects to do and then what to do within each project.

By contrast, Agile development is designed specifically to help product developers rapidly create working software with continual validation from the customer. Once a development project has been approved and its initial requirements mapped out, Agile provides a focus on execution—that is, writing lines of code. In practice, the Agile development stage typically consists of a number of short development cycles, known as sprints, with each sprint undertaken by a dedicated project team. The outcome of each sprint should be a working product (executable code) that can be demonstrated to stakeholders (customers, for example). An iteration may not produce enough functionality to warrant a market release, but the goal is to have a potentially available release at the end of each iteration. A sprint iteration typically lasts two to four weeks; multiple iterations are usually required to bring a product or major new features to the point of market release. In this way, product requirements, which are not totally known at the start, are revealed and validated through iteration, and requirements that are initially thought to be important but turn out not to be are weeded out.

Agile–Stage-Gate expert Peter Fürst, managing partner of consulting firm Five Is Innovation Management, offered another conceptualization of the differences between the two systems in a private conversation: "In project management, there are three variables: scope of work, budget, and time. In traditional methods, scope of work is fixed (the product requirements), and budget and time are flexible. But in a time-boxed system, for each sprint, time and budgets are fixed, and scope of work flexible" (Figure 1).

TABLE 1. Characteristics of Stage-Gate vs. Agile

	Stage-Gate	Agile
Type	Macroplanning	Microplanning, project management
Scope	Idea to launch	Development and testing, can be expanded to pre-development
Organization	Cross-functional team (R&D, marketing, sales, operations)	Technical team (software developers, engineers)
Decision model	Investment model—go/kill decisions involve a senior governance group	Tactical model—decisions about actions for next sprint made largely by self-managed team

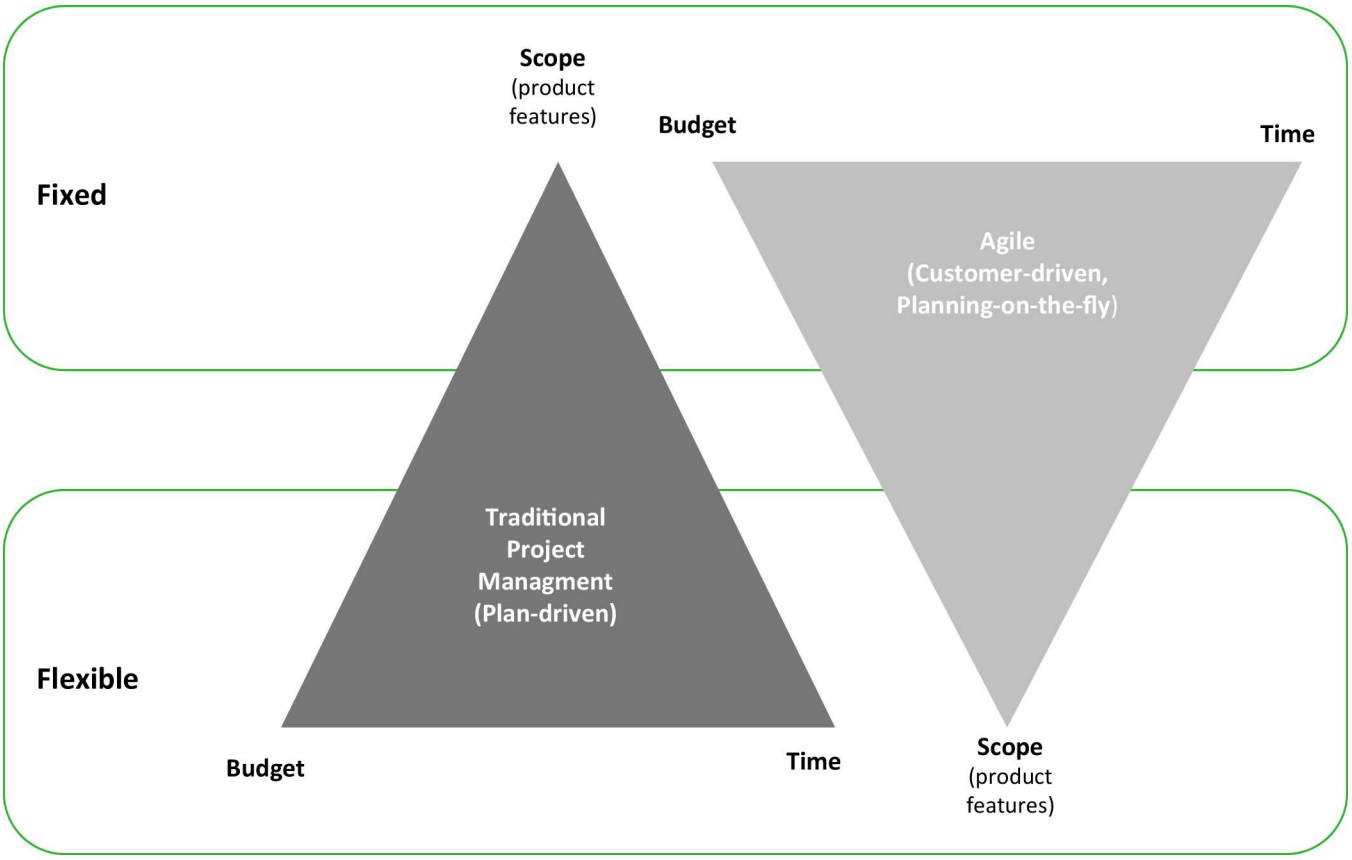

FIGURE 1. Fixed vs. flexible elements in Stage-Gate and Agile (figure courtesy P. Fürst, Five Is Innovation Management)

To sum up, Agile is a microplanning or project management tool designed to engage a development team, including the customer, in getting to a working end product quickly. Agile is used mostly during the development and testing stages of a new-product project—that is, for two stages out of the five or six included in the typical Stage-Gate process. And it is principally used by the technical people doing the actual development work.

Discounting all the hype—Agile has received significant attention since the emergence of the Manifesto—Agile does appear to offer some important benefits for software companies. In their study of its implementation in IT contexts, Begel and Nagappan (2007) identified three primary benefits: improved communication and coordination, quicker product releases, and faster responses to changed customer requirements or technical challenges. With these important benefits, not surprisingly Agile began to be adopted and embraced by much of the software development industry.

Blending Agile and Stage-Gate

As Agile took root in the software industry, a few larger IT firms that had formal development systems already in place began to build it into their existing gating processes, thus creating hybrid models. Their experience suggests that Agile and Stage-Gate can be used together to advantage. For instance, Karlstrom and Runeson (2005, 2006) studied three large, European high-technology firms where Stage-Gate and Agile were integrated for IT projects. The three firms that took part in this Swedish study—Ericsson, ABB, and Vodafone—all already had Stage-Gate systems; they simply built Agile methods (the XP version) into their existing processes from the development-approval gate onward. The researchers found first, that the integration did work—the two models were indeed compatible—and second, that this hybrid approach yielded several major payoffs:

- *Better internal team communication*, leading to the team feeling more in control, and, incidentally, to better and more visually intuitive progress metrics for management, for example, the burndown chart.
- *More efficient planning*, based on early customer feedback on the really important product features, avoiding inflexible, fixed plans that lead to delays on important features, and "requirements cramming" at the end of development.
- *Improved customer feedback*, as Agile processes seek continuous feedback from customers, making the technical project manager a good candidate for the role of customer representative.
- *Clearer resolution of documentation issues,* as priorities are resolved between documentation and code.
- *Improved attitudes,* as developers are more motivated by the improved communication and sense of control.

There are, of course, also some challenges: teams communicate better internally, but the dedication of full-time teams to the project may lead to more isolation from other

The sprint approach has been enabled by the fact that in some fields, hardware development is becoming more like software development.

parts of the organization; long-range planning tends to be ignored in favor of a focus on the current sprint; and conflicts and resistance may remain, particularly among managers who must give up some control during the Agile portions of the development process.

Overall, though, the researchers conclude, "Agile methods give the stage-gate model powerful tools for microplanning, day-to-day work control, and progress reporting" (Karlstrom and Runeson 2005, 49). The daily face-to-face meetings called for by Agile methods 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. Conversely, they 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 . . . [Stage-Gate] gives support not only for the communication within the project, but also for decision-makers sponsoring the project or acquiring the outcome of the project" (Karlstrom and Runeson 2006, 204). Thus, Agile offers greater efficiency and focus, and Stage-Gate provides a means to coordinate with other development teams and communicate with functions such as marketing and senior management.

Scrum and Stage-Gate: Applying Hybrid Development Processes to Physical Products

Recently, Agile has begun to attract serious interest from developers of physical products (Cooper 2014; Ovesen and Sommer 2012). In manufacturing firms, Agile was first adopted by IT departments or by R&D groups in which software development was a key part of hardware projects (for example, telecommunications systems). The results of these initial projects encouraged R&D groups working on hardware development to experiment with Agile, and to modify the method to fit their needs (Sommer et al. 2015).

To some extent, the sprint approach has been enabled by the fact that in some fields (such as electronics and electro-mechanical systems), hardware development is becoming more like software development, with shorter, faster iterations in the development stage. Newer techniques and tools, such as computer simulation and 3D printing, mean that traditionally long-lead development-stage tasks (for instance, securing cast components or electronic boards) can now be compressed or even eliminated. This means hardware development in these areas can look more like software development, with multiple quick iterations and multiple, working prototypes.

There have been some challenges for manufacturers adopting Agile practices, among them a lack of scalability, a proliferation of meetings, and a lack of management buy-in due to the differences from the familiar gating systems. Management resistance may also be attributed to some common misconceptions: implementing the Scrum version of Agile, for instance, does not necessarily mean abandoning Stage-Gate; Scrum can be added to Stage-Gate, creating a hybrid that incorporates positive features of both (Sommer et al. 2015).

In fact, the Scrum method seems to be the most popular Agile variant among the handful of firms employing Agile for physical product development (Sommer et al. 2015). Scrum was first identified in 1986 as "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach" (Takeuchi and Nonaka 1986, 1995). Takeuchi and Nonaka described a new approach to commercial product development that would increase speed and flexibility, which they called the rugby approach. The whole process is performed by one cross-functional team working across multiple overlapping phases, during which the team tries to go the distance as a unit, passing the ball back and forth, similar to the way in which a rugby team moves the ball down the field. In rugby, a scrum is the manner of restarting the game after a minor infraction; in new-product development, a scrum is a meeting of the project team to to plan its next moves—that is, to decide how to move the ball forward (Schwaber and Beedle 2002; ScrumInc 2013).

As with other Agile methodologies, Scrum is employed mainly in the development and testing phases of a product-development project. The project has been approved by this point in the gating process, but the development stage is not definitively planned in advance; instead, it is broken into small increments—iterations or sprints—each with its own sprint plan. Sprints are time-boxed, limited to very short timeframes, typically from one to four weeks. Each sprint is preceded by a planning meeting at which three questions must be answered (ScrumInc 2013):

- What does the customer value most (based on feedback from customers in the previous sprint)?
- What can be delivered in the upcoming sprint?
- What work is needed to achieve this deliverable?

At the meeting, the team identifies the tasks to be accomplished during the sprint and makes a commitment to the sprint goal. Thus, the goals and work plan for the sprints are very much in the control of the project team, which is self-managed, just as in Agile-Scrum in the IT world.

Each sprint is followed by a retrospective meeting at which progress is reviewed and lessons for the next

sprint are identified, including feedback from the customer. At this point, the method may diverge from its practice in the IT world. In the case of software development, the outcome of each sprint is a completed, useable, and potentially releasable product increment. For physical product development, however, the definition of a "done" deliverable is very different—creating a potentially releasable, working product every two weeks is not usually feasible. Thus, the definition of success for a sprint and the way tasks are allocated to sprints may be different in the hardware context.

There are also some important differences from a typical Stage-Gate process. First, Scrum–Stage-Gate project teams must be dedicated—that is, working only on this one project—and physically collocated in a dedicated project room (Sommer, Dukovska-Popovska, and Steger-Jensen 2014). The scrum project room is equipped with at least one large white board (called the "scrum board"), used for visually displaying sprint details and key project status information. The team begins each day with the daily scrum, a 15-minute event at which the team synchronizes activities and creates a plan for the next 24 hours. Each sprint works from the sprint backlog, a list of priority features, product increments, and tasks to be completed in the current sprint (items defined at the sprint planning meeting). Progress is monitored via a burndown chart, a two-dimensional graph with the sprint time-period on the x-axis and remaining sprint task times on the y-axis. Behind-schedule tasks are immediately visible on the burndown chart, providing an ongoing focus on executing tasks according to plan. The scrum master, who is a servant-leader for the development team, ensures that the team adheres to Scrum theory, practices, and rules (ScrumInc 2014). He or she also facilities the daily scrums.

A study of five major Danish manufacturing firms that implemented Agile–Stage-Gate hybrid models revealed positive results for this Scrum hybrid (Sommer et al. 2015). The companies, in a range of industries from consumer products to B2B heavy equipment, reported many of the same results found in the IT world, namely:

- Design flexibility (a faster response to change),
- Improved productivity, communication, and coordination among project team members,
- Improved focus on the project leading to better prioritization, and
- Higher morale among team members.

The Danish study also revealed some negatives, namely delays due to the difficulty of finding dedicated team members, difficulties in linking project teams to the rest of the organization, mismatches between the requirements of Scrum and the company's reward system, and a sense that the system was still too bureaucratic.

In some firms, Scrum–Stage-Gate is used for more than just the two technical stages. It can also be employed in the predevelopment stages, to develop the concept and assess feasibility. In these early phases, open knowledge gaps become analogous to desired software features on the burndown chart, and Scrum then works in the normal way, with each sprint aimed at resolving a particular gap or set of gaps.

Case: Scrum–Stage-Gate in the Heavy Equipment Sector

A global Swedish manufacturer (automotive industry, B2B) adopted a hybrid Agile–Stage-Gate approach when faced with the challenge of accelerating the development of vital mechanical, electronic, and software subsystems. The company had employed for years a traditional gating system in which considerable effort was spent in the front-end work to avoid entering full-scale development with many knowledge gaps. But it was difficult to define tangible, distinct tasks for these front-end phases, and so project teams ended up focused on the technical side, namely on designs and drawings. As a result, there were many knowledge gaps, in Voice of the Customer data, market requirements, and technical concept capabilities. Thus teams were rushing into the development stage without knowing how the concept would perform technically or if it would meet customer requirements.

Scrum was introduced to increase the speed of development and make the front-end work crisper. Four-week sprints were defined, scheduled consistent with calendar months to make planning and time reporting easy. Each project was assigned a visualization room with the scrum board on the wall on one side and a number of alternative designs on the other side. Each team held scrum meetings in front of the board twice a week.

The clear focus and tight follow-up created by the Scrum approach ignited a strong drive on the project teams: peer pressure within the teams was considerable, with team members pushing each other to deliver on the sprint list. The burndown curve, updated after each scrum meeting, provided the team with an indicator of progress toward the sprint goals. Teams also learned to be more realistic in work planning after a few sprints.

The concept of time-boxing was also introduced to improve efficiency in some tasks—for example, concept evaluation. The time limit, expressed as a task requirement to "make the best possible use of 10 hours to evaluate concept X," helped the team avoid over-engineering. Agreed-on definitions of "done," which included results documented as a single-page report, formatted for posting, reviewed by a colleague, and checked into the document repository, also helped teams know when to move

The definition of success for a sprint and the way tasks are allocated to sprints may be different in the hardware context.

forward. Demonstration meetings with the major stakeholders outside the project team were held after every sprint, and sprints were closed with retrospective meetings at which outcomes were reviewed and next steps determined.

Lars Cederblad, Senior Partner at management consulting firm Level 21, which supported the company in developing its new approach, described his experience with the hybrid process, and its results, in a private conversation with me:

> I was acting as scrum master and independent change agent during the first 15 months of the project. The results really exceeded our expectations, with a speed increase of around 30 percent. With that comes more motivated staff and higher employee satisfaction. We also showed that Scrum is excellent for closing knowledge gaps, the focus of the front-end phases of a project.

Four years after implementation, most of the business's projects now follow the Scrum method within the gating system, and with the same positive effects. The burndown curves from all projects are now reviewed at the senior-level Project Pulse meeting, allowing management to identify potential problems and act before they occur.

Why Agile–Stage-Gate Hybrids Work for Physical Products

The benefits of Stage-Gate have been well researched and documented: discipline, the staged structure, the go/kill decision-points that cull out bad projects, clear expectations (in the form of defined deliverables) for project teams, and built-in best practices, to name a few. The benefits of the Scrum version of Agile are less well known to hardware developers, but the admittedly limited experience with Agile–Stage-Gate hybrid development models, much of it European, suggests that manufacturers can benefit greatly from this new approach.

That's because the hybrid model balances the benefits and challenges of the two different approaches, creating a number of important advantages. The hybrid Agile–Stage-Gate model, specifically using the Scrum version of Agile:

- *Gets the product right.* The hybrid method requires the project team to develop something physical or visual, early and cheaply (the sprints), and quickly get it in front of customers for feedback. As Steve Jobs, never a proponent of traditional market research, famously said, "People don't know what they want until you show it to them" (Isaacson 2011, 567), especially in the case of more radically innovative products. An Agile–Stage-Gate approach addresses this challenge well: the method shows customers something they can see, all the way through the project, beginning even before the development stage commences. The system is also highly adaptive. If product requirements change, the design can be modified early when the cost of change is lower, similar to the strategic pivot in the Lean Startup method (Ries 2011). Finally, building something physical early and often means that solutions to technical issues can be worked through as early proof-of-concept prototypes emerge.
- *Accommodates uncertainty.* In traditional stage-and-gate methods, the problem is identified and defined by conducting investigations before development begins. These early stages, or "homework phases," require the project team to undertake market, technical, and business assessments in order to define the product and financially justify the project. Thus, the requirements for the solution are largely defined even before the product enters development. But not every project is so definable. When there is much uncertainty—for example, in the case of a highly innovative or bold initiative—and where no amount of voice-of-customer work or technical assessment can get all the answers, then the problem can only be understood through experimentation. This means trial and error: building and testing possible solutions, which Agile sprints and iterations allow. Thus, in an Agile or hybrid approach, requirements are not defined before development but are established as part of the solution-finding process.
- *Accelerates development.* Time-boxed sprints, and even time-boxed tasks within sprints, bring a sense of urgency to the development project. In Scrum, all events are time-boxed events, so every event has a maximum duration; once a sprint begins, its duration is fixed and cannot be lengthened (ScrumInc 2013). Thus project teams commit to certain deliverables at the beginning of each sprint and then are under pressure to deliver within the agreed timeframe. This forces teams to focus on the essentials and deliver results, rather than focusing on a large, finalized list of requirements or features.
- *Focuses teams.* Agile–Stage-Gate project teams are dedicated to the one project to ensure adequate resources to get the work done on the compressed sprint timeline. The notion of dedicated project teams is not new: 24.1 percent of top-performing businesses already use focused teams, but only 11.4 percent of average firms do (Cooper 2013, 26). Scrum won't work optimally without a dedicated team—and this one step alone increases speed dramatically by making sure the project is adequately resourced and supported. Most traditional project teams are woefully underresourced, the result being that projects move painfully slowly. Frequently,

The hybrid model balances the benefits and challenges of the two different approaches.

even the project leader is spread across multiple projects, and so lacks focus and dedication. By ensuring solid resourcing, Agile–Stage-Gate helps drive new products to market much more quickly.

- *Improves Within-Team Communication.* Dedicated teams, a dedicated space where the entire team resides, and daily, face-to-face scrums all contribute to improved communication. Every study of Agile (whether for IT or physical products) reports this benefit. This leads to more effective, cross-functional teams with good internal cooperation and communication—a factor frequently cited as a key to both increased speed to market and higher success rates in new product development (Cooper 2013, 25).

The benefits of building Agile-Scrum into a traditional Stage-Gate system are many, as shown by the evidence, including enthusiastic comments from users in the manufacturing world. But what adjustments must be made when applying Scrum to the development of physical products?

Defining a Done Sprint

Clearly, Agile, and particularly Scrum, has value for product development, but Scrum methods cannot be directly implemented for hardware without some modification. One key point of difference is in the definition of sprints and what constitutes a done sprint. Software development is almost infinitely divisible; an IT development consisting of multiple product features can be broken down into multiple, small subprojects, which can each be completed in a single sprint. A done sprint is a working product (executable software) that meets the goal of the subproject and can be demonstrated to stakeholders (customers). Thus, each increment—each sprint—yields a working, albeit feature-limited, product.

By contrast, the development of a new machine, food item, or polymer cannot be easily incrementalized. If your product is beer or a diesel engine, you cannot build part of the beer or part of the engine and demonstrate it working; it certainly won't be releasable to the market. Moreover, it is usually not possible to have anything that actually functions ready and available within a few weeks. Thus the notion of short time-boxed sprints and the IT definition of "done" do not apply so neatly to hardware (Cooper 2014).

A solution may be found in newer versions of Stage-Gate that build in spirals or iterations as a way to make the traditional 1990s gating model more adaptive and responsive to fluid market conditions and changing customer requirements (Cooper 2011; Cooper and Edgett 2005). In these models, each iteration builds a product version somewhere between a concept (or virtual product) and a ready-to-trial prototype. Unlike in pure Agile, the result of a sprint may not be a working product but is something that can be shown to the customer to seek feedback—to test a market-facing hypothesis and to seek proof of concept. These product versions, or "protocepts," can be computer-generated 3D drawings, virtual prototypes, crude models, working models, or early prototypes. The result of a done sprint, in this context, may not be a working product, but it is something physical that the customer can respond to (Cooper 2014, 22).

Unlike in pure Agile, the result of a sprint may not be a working product but is something that can be shown to the customer to seek feedback.

If Scrum is applied to earlier stages of the project, for example, the concept and feasibility stages, then the definition of done is relaxed even further, to include anything tangible that can be reviewed by an expert. For example, the results of a market study or voice-of-customer work could count as a done deliverable.

There is strong evidence that this spiral, iterative development approach is feasible and works for hardware products: 44.8 percent of top-performing businesses practice these build-test-feedback-revise iterations with customers, compared to only 26.3 percent of firms on average (Cooper 2012, 599).

Case: Scrum–Stage-Gate in the US Consumer Electronics Sector

One large US manufacturer of electromechanical control devices for homes has increasingly moved into remote control devices, for example to control the household thermostat, lighting, and even the front door, sometimes via smartphone connectivity. Thus, an ever-larger percentage of each new-product project entails software development. To no one's surprise, the perennial conflict been the hardware and software developers arose: Stage-Gate or Agile?

In response, Richard Peterson, the company's vice president for new product development, introduced the concept of Agile within Stage-Gate, integrating the two concepts to improve development efforts across all groups and all project types. As he told me, "We developed a modified Agile approach that requires a rigorous Stage-Gate process, and continual end-to-end assessment." The firm now uses Agile sprints and scrums for both physical and IT development within Stage-Gate phases. Agile is employed in particular in the development and testing stages of the Stage-Gate process. A scrum master oversees daily scrums, about 20 minutes in length; the firm also builds design reviews into some scrums and even brings in peers and outsiders for a peer review.

Sprints are about two weeks in length. For this firm's remote control interconnected devices, it is usually not possible to produce a potentially releasable product every

two weeks, but the project team must show something physical at the end of a sprint that was defined at the start; this is the result of completed tasks in that sprint—and not just a slide deck. The result of a sprint could be, for example, a set of completed design drawings or a prototype or an early working model of the product.

In this firm's system, project teams have dedicated team members for each project. Because dedicated teams are not feasible for every project, the firm uses this Agile–Stage-Gate approach only for the larger, major revenue-generating projects—about 20 percent of the projects in their development pipeline.

The company has been using this hybrid process on all major new-product initiatives for over two years. The process has worked well, according to senior management, and has driven down cycle times. Also, there is much better communication within development teams, and a heightened sense of community.

A few challenges have arisen. Project leaders and teams tended to become so focused on the sprints—the next few weeks and their objective for that sprint—that the team lost sight of the ultimate goal. Senior management now meets with hybrid teams periodically (more frequently than just at gates) to ensure that the longer-term view is considered and the ultimate goal is clear. The problem is now resolved. Additionally, senior leaders were initially somewhat skeptical of the new Scrum system. Thus, they were not required to "speak Agile," and the firm did not change the new-product development language used in the business. Moreover, the gates remained as they had been in the firm's gating system: deliverables from the previous stage were checked, and a go/kill decision was made to move to the next stage. The changes took place at the project team level—multiple sprints were employed within the development and testing phases, and program managers (project leaders) were subjected to much pressure to learn how to facilitate the Agile process and to become scrum masters.

Wrap Up

For physical product developers, an Agile–Stage-Gate hybrid product development model is feasible and may yield positive results. Sprints can be employed for maximum speed, consistent with the IT Agile-Scrum model. But sprints are usually restricted to the development and testing stages. And the result of each sprint may have to be redefined somewhat to include something physical, the result of a completed task. Additionally, spirals—a series of build-test-feedback-revise iterations—make the system more adaptive; these spirals fit well into the Agile sprinting concept, in which at the completion of sprints, some version of the product—a protocept—can be demonstrated to stakeholders (customers and management). Finally, dedicated teams, which are a must for this system to work well, help accelerate the project even more.

The early evidence, albeit quite limited, is encouraging. Lead users of this new hybrid system are enthusiastic. In all the cases I've cited, the companies have expanded their use of the hybrid model, which speaks to the results it has delivered. Indeed, integrating Agile-Scrum methods into Stage-Gate to yield this new Agile–Stage-Gate hybrid model may be the most exciting and significant change to the new-product process since the introduction of gating systems more than 30 years ago.

References

Begel, A., and Nagappan, N. 2007. Usage and perceptions of agile software development in an industrial context: An exploratory study. In *ESEM '07: First International Symposium on Empirical Software Engineering and Measurement*, 255–264. Washington, DC: IEEE.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. et al. 2001. Manifesto for Agile Software Development. http://agilemanifesto.org/

Boehm, B., and Turner, R. 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 edition*. New York, NY: Basic Books.

Cooper, R. G. 2012. The Stage-Gate system for product innovation in B2B firms. In *Handbook of Business-to-Business Marketing*, ed. G. L. Lillien and R. Grewat 596–624. Northampton, MA: Edward Elgar Publishing.

Cooper, R. G. 2013. New products: What separates the winners from the losers and what drives success. In *The PDMA Handbook of New Product Development*, 3rd Edition, ed. K. B. Kahn 3–34. Hoboken, NJ: John Wiley & Sons.

Cooper, R. G. 2014. What's next? After Stage-Gate. *Research-Technology Management* 57(1): 20–31.

Cooper, R. G., and Edgett, S. J. 2005. *Lean, Rapid and Profitable New Product Development*. Product Development Institute.

Isaacson, W. 2011. *Steve Jobs: The Exclusive Biography*. New York: Simon & Schuster.

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

Karlstrom, D., and Runeson, P. 2006. Integrating Agile software development into Stage-Gate managed product development. *Empirical Software Engineering* 11:203–225.

Ovesen, N., and Sommer, A. F. 2015. Scrum in the traditional development organization: Adapting to the legacy. In *Modeling and Management of Engineering Processes, Proceedings of the 3rd International Conference 2013*, 87–99. Berlin: Springer-Verlag.

Reagan, B. 2012. Going Agile: CA Technologies, Clarity PPM Division's transformative journey. Digital Celerity, San Francisco, CA, September 22. http://www.slideshare.net/DCsteve/going-agile-with-ca-clarity-ppm-agile-vision

Ries, E. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation To Create Radically Successful Businesses*. New York: Crown Publishing.

Schwaber, K., and Beedle, M. 2002. *Agile Software Development With Scrum*. New York: Prentice-Hall.

ScrumInc. 2013. *The Scrum Guide*. July. http://www.scrumguides.org/scrum-guide.html

Sommer, A. F. Dukovska-Popovska, I., and Steger-Jensen, K. 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: International Conference, APMS 2014, Proceedings, Part I*, 184–191. Berlin: Springer-Verlag.

Sommer, A. F., Hedegaard, C., Dukovska-Popovska, I., and Steger-Jensen, K. 2015. Improved product development performance through Agile/Stage-Gate hybrids: The next-generation Stage-Gate process? *Research-Technology Management* 58(1): 34–44.

Takeuchi, H., and Nonaka, I. 1986. The new new product development game. *Harvard Business Review* 64(1): 137–146. https://hbr.org/1986/01/the-new-new-product-development-game

Takeuchi, H. and Nonaka, I. 1995. *The Knowledge Creating Company*. New York: Oxford University Press.