Agile-Stage-Gate:
New Idea-to-Launch Method for Manufactured New Products Is Faster, More Responsive
Robert G. Cooper & Anita F. Sommer

New evidence reveals that Agile methods, until now used primarily for IT developments, can be integrated with traditional gating approaches to yield significant potential benefits for manufacturers of B2B physical products. Indeed, this new Agile-Stage-Gate hybrid approach represents a significant change to our thinking about how new-product development should be done since the introduction of today's popular gating systems thirty years ago!

The article shows how Agile emerged in the IT industry and early attempts to integrate it with gating models, also in the IT world. The article moves on to the recent use of this hybrid model by manufacturers, and the results achieved by early adopters when implementing this new system in industries from food to heavy equipment. In terms of implementation, the details of the new Agile-Stage-Gate system are presented, including the “Power of Nine” – the three key artefacts (such as sprints and scrums); three important tools (such as sprint backlogs and burndown charts), and the three vital roles (such as the product owner and the scrum master) needed to make it work.

Agile from the IT world cannot be directly integrated into Stage-Gate for physical products without some important modifications, however. These needed adjustments – such as redefining a “done sprint” and how to present versions of the product or “protocepts” for continuous customer feedback – are outlined, complete with a case study from an equipment manufacturer. Additionally, the article identifies and deals with ten important issues and apparent inconsistencies that arise when implementing this new system for B2B products.


© Cooper & Sommer & IMM
Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive

Robert G. Cooper, Dr. a,b,⁎, Anita F Sommer, Dr. c

⁎ IBM Distinguished Research Fellow at Institute for the Study of Business Markets at Penn State University's Smeal College of Business Administration, USA
a Professor Emeritus at DeGroote School of Business, McMaster University, Canada
b ISBM Distinguished Research Fellow at Institute for the Study of Business Markets at Penn State University's Smeal College of Business Administration, USA

A R T I C L E   I N F O

Article info
Received 9 February 2016
Received in revised form 4 June 2016
Accepted 12 July 2016
Available online xxxx

Keywords:
Agile
Stage-Gate
Hybrid
New-product process
Gating

A B S T R A C T

New evidence reveals that Agile methods, until now used primarily for IT developments, can be integrated with traditional gating approaches to yield significant potential benefits for manufacturers of B2B physical products. Indeed, this new Agile-Stage-Gate hybrid approach represents a significant change to our thinking about how new-product development should be done since the introduction of today’s popular gating systems thirty years ago! The article shows how Agile emerged in the IT industry and early attempts to integrate it with gating models, also in the IT world. The article moves on to the recent use of this hybrid model by manufacturers, and the results achieved by early adopters when implementing this new system in industries from food to heavy equipment. In terms of implementation, the details of the new Agile-Stage-Gate system are presented, including the “Power of Nine” – the three key artefacts (such as sprints and scrums); three important tools (such as sprint backlogs and burn-down charts), and the three vital roles (such as the product owner and the scrum master) needed to make it work. Agile from the IT world cannot be directly integrated into Stage-Gate for physical products without some important modifications, however. These needed adjustments – such as redefining a “dona sprint” and how to present versions of the product or “protocepts” for continuous customer feedback – are outlined, complete with a case study from an equipment manufacturer. Additionally, the article identifies and deals with ten important issues and apparent inconsistencies that arise when implementing this new system for B2B products.

© 2016 Elsevier Inc. All rights reserved.

1. Agile – Not just for IT development projects

Agile development methods have traditionally applied only to software new-product development projects. But new evidence reveals that Agile methods can be integrated with traditional and familiar gating approaches to yield an Agile-Stage-Gate® hybrid model. Moreover, research into very recent industry experience suggests that this new hybrid model has significant potential benefits for manufacturers of physical products, from heavy industrial equipment to food and toys, yielding surprisingly and dramatically positive results. This new approach has the potential 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 thirty years ago!

Many R&D managers, familiar with Agile from their internal IT departments, are skeptical as to whether Agile can be used with gating approaches and/or for hardware or physical product development (Barlow et al., 2011). Development of software products, after all, is clearly quite different than new-product development in the manufacturing world. So can Agile work in a manufacturing context? The evidence is limited, but early results from lead users in the manufacturing world suggest that Agile methods can be combined with traditional stage-and-gate or plan-based models and does indeed work very well for manufactured products, even heavy B2B goods. Some of the reported benefits of this new Agile-Stage-Gate hybrid model include:

- Builds in voice-of-customer (VoC) continuously – early, often and cheaply – to get the product right.
- Responds quickly to changing customer needs, which is critical when facing fluid markets where things change quickly; and also to changing technical options and solutions.
- Copes with uncertainty and ambiguity, often characteristic of more innovative developments.
- Deals with the resourcing issue (for example, via dedicated team members) more directly.

http://dx.doi.org/10.1016/j.indmarman.2016.10.006
0019-8501/© 2016 Elsevier Inc. All rights reserved.

Please cite this article as: Cooper, R.G., & Sommer, A.F., Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive, Industrial Marketing Management (2016), http://dx.doi.org/10.1016/j.indmarman.2016.10.006
• Improves within-team communication and team motivation, reduces cycle time, and is more productive.

2. Gating approaches and Agile methods

2.1. How gating methods work

Traditional gating methods to product development have been around since the mid-1980s and are typified by the Stage-Gate® model; they have been widely adopted by manufacturers in both B2B and B2C sectors (Cooper, Edgett, & KleinSchmidt, 2002, 2005; Griffin, 1997). Stage-Gate breaks the idea-to-launch process into a series of five or six discrete stages or phases, beginning with “Idea Generation” and moving to “Product Launch” and beyond (see Fig. 1). Each stage costs more than the preceding stage, so that the process is very much an incremental commitment one, much like buying a series of options on a property – thus risk is mitigated. Each stage contains a set of known success-drivers and tasks, such as voice-of-customer studies, fact-based product definition, and robust front-end homework, so that best practices are built into every development project by design. And preceding each stage is a gate, where Go/Kill and investment decisions are made – are we doing the right project, and are we doing this project right? Here poor projects are culled out, so that resources are funneled to the best projects as their potential becomes clearer. Stage-Gate is also cross-functional (that is, involves people from marketing, sales, and operations alongside technical personnel). Gating models in general are considered to be plan-based approaches.

While gating methods have proved very effective in most development applications, especially in B2B industries, there have been criticisms as well (Becker, 2006; Cooper, 2014; Lenfle & Loch, 2010). Such models are seen as being too linear, too rigid and too planned to deal with today’s fast-paced and often quickly-changing world. In the IT sector, traditional plan-based methods, such as the waterfall process, 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 are 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”. 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.

2.2. Agile in the IT world

Agile methods were 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. The Agile Manifesto, created by thought-leaders in IT in 2001, 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) – see Fig. 2. Agile methods emphasize individuals over processes, working software over complete documentation, collaboration over contracts, and flexibility over planning.

Agile development is designed specifically to help product developers rapidly create working software with continuous 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, Agile breaks the development phase of the project into a series of very short time-boxed iterations or sprints, each typically about 2–4 weeks long. The goal of each sprint is to deliver working software code that can be demonstrated to stakeholders – that is, software that is potentially releasable to the market (although it usually takes several sprints to product a market-ready product).

The dedicated development team, which is co-located in one room, meets every morning for their daily stand-up or daily scrum to discuss issues and what they will accomplish that day. Sprint planning meetings at the beginning of each sprint decide what can be realistically accomplished in the next 2–4 week sprint. Customer feedback and required product changes are introduced at the end of each sprint for action in the next sprint. Thus the Agile process is very fast and responsive – it is not plan-based, but more “plan on the fly”, sprint by sprint. Note that Agile is used principally for the “technical phases” of the project, namely the equivalent of the Development (Alpha) and Testing (Beta) stages of the typical gating process in Fig. 1.

2.3. 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 achieve a competitive 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 study, namely Ericsson, ABB and Vodafone, all employed Stage-Gate systems, so they simply built in Agile methods (the XP version2) into their existing model from the development approval gate onward.

The study found that the integration did indeed work – the two models were compatible – and that this hybrid approach yielded several major payoffs:

• Better internal team communication, leading to the team feeling more in control.
• Better and more visually intuitive progress metrics for management, for example, the burndown chart (shown later in Fig. 8).
• More efficient planning, based on early customer feedback on the really important product features; and avoiding inflexible, fixed plans that lead to delays on important features and “requirements cramming” at the end of development.
• Improved customer feedback, via continuous feedback from customers.
• Improved morale on the project team: 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 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 & Runeson, 2005, 49). They also note that “software development projects are not isolated activities (Karlstrom and Runeson, 2006, 204). Such projects 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

---

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.

2 There are an estimated 26 different versions of Agile.
not only for the communication within the project, but also for decision-makers sponsoring the project or acquiring the outcome of the project. 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.

The differences between plan-driven software development (based on gated or waterfall models) and Agile approaches for IT projects are aptly summarized by Boehm and Turner (2004). For instance, plan-driven approaches emphasize verification (working products that reflect the original requirements) and validation (the final product satisfies its intended mission), while Agile emphasizes simple designs based on YAGNI (you aren’t going to need it), a concept of not designing anything that is not needed currently – because, as things change, it may never be needed.

The differences emerge from the different intents of the two systems: Stage-Gate is a comprehensive idea-to-launch system and a
macroplanning process, whereas Agile is a microplanning project management methodology (Table 1). The authors propose integrating agility and plan-driven approaches – both are useful – to suit the needs of a given project. This integration depends on looking at the risks of swinging too far in either the plan-driven or the Agile directions. Case study illustrations show the combined Agile-Stage-Gate model successfully in use for IT projects.

Agile has received significant attention since the creation of the Manifesto and does appear to offer 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.

3. Applying Agile–Stage-Gate to manufactured B2B products

B2B manufacturers today experience increasing external pressure and internal complexity, thus creating a need to act faster and be more flexible during product development. This strains the traditional product development model, where R&D people respond to requests to develop products based on requirements and fixed specifications often gleaned from limited market insights. This traditional approach is no longer feasible due to the pace of change; thus, a handful of new-product managers, especially in B2B, now recognize the need for a new approach and mindset for product development. The Agile-Stage-Gate hybrid model is the solution some firms have elected for product development, accommodating rapid design cycles (including changes in customer needs) and providing early and fast customer validation.

3.1. Dramatic results

Agile-Stage-Gate has indeed begun to be successfully adopted by developers of physical or manufactured products in the B2B world (Cooper, 2014; Ovessen & Sommer, 2015). In lead-user manufacturing firms, Agile was first adopted either by internal 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, Hedegaard, Dukovska-Popovska, & Steger-Jensen, 2015).

A study of five major Danish manufacturing firms that implemented Agile-Stage-Gate hybrid models revealed very positive results (Fig. 3: Sommer et al., 2015). The companies, in a range of industries from B2B heavy equipment to professional products to one strictly consumer goods firm, reported many of the same results found in the IT world, namely:

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

Fig. 3 shows performance results quantitatively: note how very strong the positive responses are. Similar results were reported in case studies of other firms (Cooper, 2016; Cooper & Sommer, 2016).

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 Agile and the company's reward system, and a sense that the system was still too bureaucratic (Fig. 4). But these negatives are quite subdued when compared to the enthusiastic positive feedback in Fig. 3 above.

Other challenges for manufacturers adopting Agile practices have been identified, including a lack of scalability, a proliferation of meetings, and a lack of management buy-in due to the differences from the

---

**Table 1**

Where Agile and plan-driven gating models fit.

(Adapted from: Boehm & Turner, 2004)

<table>
<thead>
<tr>
<th>Characteristics of project or setting</th>
<th>Agile home ground</th>
<th>Plan-driven (gating system) home ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criticality of project</td>
<td>Low</td>
<td>Extreme</td>
</tr>
<tr>
<td>Developers experience</td>
<td>Senior (experienced)</td>
<td>More junior</td>
</tr>
<tr>
<td>Product requirements</td>
<td>Change during project</td>
<td>Stable requirements &amp; specs</td>
</tr>
<tr>
<td>Project team size</td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>Company culture</td>
<td>Culture that responds to change</td>
<td>Culture that demands order</td>
</tr>
</tbody>
</table>

---

**Fig. 3.** Payoffs from implementing Agile-Stage-Gate in five manufacturing firms (mean scores across firms).

(Adapted from Cooper & Sommer, 2016)
familiar gating systems. Management resistance may also be attributed to some common misconceptions: implementing Agile, for instance, does not necessarily mean abandoning Stage-Gate; Agile can be added to Stage-Gate, creating a hybrid that incorporates positive features of both (Sommer et al., 2015).

3.2. How Agile-Stage-Gate works for manufacturers

Although there are reported to be at least 26 different versions of Agile, the Scrum method seems to be the most popular Agile variant among the handful of firms employing Agile for new manufactured products (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 & Nonaka, 1986). This new commercial product development method, labeled the rugby approach, promised increased speed and flexibility: The whole idea-to-launch 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, somewhat like a huddle in North American football; in new-product development, a scrum is a meeting of the project team to plan its next moves – that is, to decide how to move the ball forward (Schwaber and Beedle, 2002; ScrumInc, 2013).

Agile-Stage-Gate works much the same for B2B manufacturers as it does in the IT world – but there are some important differences too. For manufacturers, Agile-Stage-Gate is also based on a series of short time-boxed sprints, each lasting about 2–4 weeks, sometimes longer. At the end of each sprint, the project team produces “something physical” that can be demonstrated to stakeholders, often to customers. Each sprint is planned in real time – on the fly – and thus the process is highly responsive and adaptive. And the rapid, iterative, and incremental releases of concepts, designs and rapid prototypes provides fast and timely feedback – this gets the product right and improves the odds of market success. The voice of the customer thus becomes a dynamic driver throughout the project, as customers also better define their own needs through their active involvement (for example, via frequent user tests or the customer co-designing). The project team itself is dedicated to the one project (not spread too thinly) and is co-located in one team room, thereby ensuring faster developments and better communication. And the entire process is very visual, with novel visual metrics that display plans, progress, and results in real time.

Gates and stages remain very much part of the process, with gates being the vital Go/Kill decision points – culling out the weak projects and providing focus in the development pipeline. Stages provide a high-level overview of the main phases of the project, and also insights into the activities required within each stage and the expected deliverables at the end of each stage.

Although Agile-Stage-Gate appears most popular use for the technical phases of the project – specifically the Development and Testing stages of the typical gating system in Fig. 1 – as firms gain confidence, they then use Agile-Stage-Gate for more than just these two technical stages. The method can be employed in the pre-development stages, to develop the concept and assess feasibility. In these early phases, open knowledge gaps become analogous to desired software or product features, and Agile-Scrum then works in the normal way, with each sprint aimed at resolving a particular gap or set of gaps.

4. The details of Agile-Stage-Gate for manufactured products

Agile methods are employed by the project team within each stage in Stage-Gate in Fig. 1, as they plan, manage and execute their work. There are nine distinct elements essential for Agile execution: three artefacts, three roles and three tools, namely the “Power of Nine”.

4.1. The three artefacts

The artefacts generate the framework for fast, incremental product releases. The first artefact is the “sprint”: this is a time-boxed work effort lasting between two and four weeks, where a small part of the project is executed. The aim is to have a product increment completed at the end of each sprint, ready to present to the customer and/or to other relevant stakeholders at a sprint review. The sprint review presents the sprint results to customers and management: In practice, not all sprints produce a product increment or prototype; some may yield a design, a concept or a rapid prototype or even the results of a VoC study – “something tangible”.

![Fig. 4. Challenges faced in implementing Agile-Stage-Gate in five manufacturing firms (mean scores across firms) | (Cooper & Sommer, 2016).](image-url)
Some B2B manufacturers employ sprints that are longer than the normal 2–4 weeks, particularly where their products are complex. This longer sprint allows the team a little more time to create something physical and more useful, that they can demonstrate to stakeholders. In other firms, the customer feedback is lengthened to more than the very quick sprint review at the end of the sprint; rather the entire sprint – perhaps the full 2–3 weeks – is devoted to getting customer feedback to versions of the product.

The second artefact is the “daily scrum” or daily stand-up. This is a short meeting, where the project team meets at the start of each day for 10–15 min to review updates. Each team member in turn states what they accomplished yesterday, what they expect to do today, and what problems they may have. After the meeting, the flagged problems are dealt with through continued within-team collaboration. The benefits of this daily stand-up is both instant knowledge-sharing and increased team motivation.

The final artefact is the “retrospect meeting” at the end of the sprint. The purpose of this meeting is not to review the sprint’s results (this is done in the sprint review), but for the team members to improve how the team functions. In this retrospective meeting, the team members challenge their own performance, highlight their successes in internal cooperation, and try to improve their ways of working together. This meeting thus helps to institutionalize continuous improvement: solving key issues that arise evolving to a high-performance team.

An illustration: Siemens (Motors Division) use the retrospect meeting at the end of each sprint to self-evaluate, and to improve team cooperation and utilization of Agile principles. Their starfish chart helps to structure the meeting to cover the topics in Fig. 5. Using Post-Its, each team member notes their concerns and suggestions on the chart; a dialog ensues on how to self-improve for the next sprint. This retrospective meeting is facilitated by the scrum master who ensures adherence to the starfish model and sees that all team members participate equally.

4.2. Three distinct roles

The most important role in Agile is the individual member of the development or project team. In Agile, the team is empowered to take responsibility for the project’s execution: it is the team, and not management, that defines, selects and allocates activities or tasks among the team members. Based on the agreed target outcome of the sprint, the team’s job is to define the necessary activities to achieve the sprint goals, to decide on who does what, and then to undertake the tasks during the sprint.

Agile also has a process manager, called the “scrum master”, a servant-leader and facilitator of the project team whose role is to support the team as they undertake each sprint. The scrum master facilitates the daily scrums, as well as the sprint planning and post-sprint meetings; he or she ensures that the team adheres to the Agile methods and that they use the artefacts and tools properly. The scrum master is also responsible for removing impediments to the team, so that they are able to execute quickly and without roadblocks.

The third distinct role is the “product owner”, who is responsible for the product backlog (the product requirement): the owner is not a project team member per se, but works closely with the development team to ensure that the right product requirements are built into the sprints. The product owner also has some of the same responsibilities as those of a project leader in traditional product development: the product owner focuses on stakeholder management in order to ensure management support, input, and resources to the project, and also manages customer involvement. Note that not all Agile-Stage-Gate user-firms adopt all three roles, and in some cases, remain with one or more of the familiar roles and responsibilities of project leaders, project managers and team members.

4.3. The three tools

Agile-Stage-Gate features three important visual tools to help manage and monitor projects. The first is the “product backlog”, which is the Agile equivalent of the traditional product definition or requirements specifications. Unlike a product definition, however, the product backlog does not contain detailed specifications (such as desired materials or performance requirements), but rather customer requirements, needs, wants and preferences (Fig. 6) – providing freedom for the project team to experiment with the product’s design, but within the guidelines in the backlog. (In some firms, the product backlog is broader, and also includes knowledge gaps to be filled and key tasks required for the next stage of the project, thus more a “project backlog” than product backlog per se). Items in the backlog are prioritized, the optimal sequence determined, and the most important tackled first. This is a very dynamic document and is continuously updated and re-prioritized as new information and learnings are integrated. At the beginning of each sprint, the sprint planning meeting takes the top priority items from the product backlog and translates them into well-defined activities or tasks and creates the “sprint backlog” (Fig. 7).

Firms in some industries employ user stories in their product backlog. Rather than requirements for the product, user stories describe how users will engage with the final product and how it fulfills the user’s needs in practice. User stories are assigned story points to indicate how much work is required – expected resource requirements – to create the feature or performance characteristic needed to accommodate that user story. Scales to gauge story points vary widely by company, but a simple measure is T-shirt sizes (S, M, L, XL).

The second tool is the “sprint board” or kanban board. The board contains the sprint backlog, the list of activities or tasks to be done during the current sprint. In practice, these are placed as Post-Its on the physical board or as activity cards on a virtual board. There are many ways to set up the sprint board, but the most common format is to use three columns: “to-do”, “doing” and “done”. During the sprint, the activities are initiated by moving a task from “to-do” into the “doing” column, and when finalized, moved into “done”. This creates a continuous flow of activity cards from left to right across the board during the sprint. If the sprint has been well planned, there will be just enough activities for execution for that sprint.

To keep track of the activity flow during a sprint, a “burn-down chart” is used. It displays the number of days in the sprint versus activities finished (see Fig. 8). Ideally, this chart should be a straight line (the dashed line in Fig. 8), ending with zero remaining activities on the final day of the sprint. In practice, deviations occur, which are captured in the burn-down chart on a daily basis – the solid line in Fig. 8. The development team thus can readily see if they are under- or over-burnt.
over-performing. The goal here is to have a constant flow and the ability to plan this flow well. Initially, a team struggles to plan the sprint accurately – they usually overestimate what they can accomplish during a sprint but over time their ability to plan improves, and so does their performance as gauged by the burn-down chart.

4.4. An integrated system

While Agile-Stage-Gate system consists of a number of elements – the nine artefacts, roles and tools – the whole is much greater than sum of its parts. The key to making it work is the change in mindset.

- Example is for a lamp-post system to visually project advertisements onto a city sidewalk
- The product owner (or project leader) and team prioritize tasks or activities for the upcoming sprint
- Additional tasks can be added to this sprint backlog
- Based on a real case; disguised

Fig. 6. The Product Backlog is similar to the “product definition” except is more dynamic (can change) and broader. It lists all the things that need to be done (work items) over the entire stage, prioritized.

The Product Backlog is a list of prioritized items on which work needs to be done. It includes customer needs, wants and requirements, and desired product features; and sometimes knowledge gaps to be filled, and/or needed project tasks.

Each new item that is added is prioritized and added to the backlog. Items may be re-prioritized at any time, and even removed from the backlog.

The Product Backlog provides the basis for the sprint planning meeting and the resulting sprint backlog for each upcoming sprint.

Fig. 7. The flexible sprint backlog developed at the sprint planning meeting, and showing tasks to be completed in the next sprint. (Adapted from Fürst, 2016)
required by both management and project teams, and the application of the entire system in an integrated fashion. While the elements of the system are useful, experience suggests that they do not bring significant performance improvement unless they are employed together and with the appropriate change in the “way of working”.

5. Customizing Agile-Stage-Gate for manufactured products

Agile has value for B2B product development, but Agile-Scrum methods from the IT world cannot be directly implemented for manufactured products without some modifications. Here are some examples:

5.1. Defining a done sprint

The concept of a “done sprint” is critical, given the tightly time-boxed nature of the process. One important point of difference between Agile for IT products and Agile-Stage-Gate for physical products is the definition of a sprint 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 sub-projects – for example, writing several thousand lines of code and producing a few screens – which can each be completed in a single sprint. A “done sprint” is a working product (executable software, a completed feature, potentially releasable) and can be demonstrated to stakeholders (management and customers). Thus, each increment or sprint yields a working, albeit feature-limited, product.

By contrast, the development of a new medical device, machine, or polymer cannot be easily incrementalized. If your product is an engine or a scanning device, you cannot build part of the engine or part of the scanner and demonstrate it working within a few weeks, nor will it be releasable to the market. Thus, the notion of short time-boxed sprints and the IT definition of “done” do not apply so neatly to B2B physical products (Cooper, 2014, 2016).

B2B firms have made the adjustment, however; here are some examples of “done sprints” in B2B manufacturing firms:

- A US-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” (Cooper, 2016).

- Siemens industrial motors: A “done sprint” is when the development team decides the sprint is done. However, the sprint results must be approved by the product owner to become “done done”, which marks the actual close-out of a sprint.

- 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” (Cooper, 2016).

- LEGO Education (professional B2B products): A “done sprint” is when a ten-element checklist (the “definition of done”) is completed by all team members, Quality Assurance and the product owner, declaring that all product acceptance criteria have been met, and that all open actions have been completed, handed over, or closed (Cooper & Sommer, 2016). Sample checklist elements at LEGO include:
  - development documentation created;
  - all open actions completed, handed over to others responsible, or closed; and
  - the Project Handover document updated with relevant product life cycle information.

When Agile-Scrum is applied to earlier stages of the project, for example, the concept and business case stages in Fig. 1, 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 technical feasibility analysis would count as a “done sprint”.

5.2. Something that can be demonstrated

Newer versions of Stage-Gate built in “spirals” or iterations a decade ago in order to make the traditional 1990s gating model more adaptive and responsive: these were typically a series of “build-test-feedback-and-revise” iterations, usually several months apart, to confirm the product’s design with the customer and to get the product right, in spite of fluid market conditions and changing customer requirements (Cooper, 2011; Cooper & Edgett, 2005).

Here, each iteration builds a product version somewhere between a concept (usually drawings and words) and a ready-to-trial prototype – we label this a “protocpt”. Unlike in Agile in the IT world, the result of a sprint may not be a working product, but is still something that can be demonstrated – to test a market-facing
hypothesis (to seek customer feedback), and coincidentally to seek technical proof of concept as well as management buy-in, as in Fig. 9. These product versions, or protocepts, can be computer-generated 3D drawings, virtual prototypes, crude models, rapid prototypes (3D-printed), 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 and which management can see (Cooper, 2014, 22).

6. An Agile-Stage-Gate case study: A global manufacturer of B2B valves and sensors

6.1. The company

ThermoValves (disguised name) is a business unit within a major corporation that provides heating solutions ultimately targeted at the residential construction sector (HVAC). The company is a global manufacturer with headquarters in Europe, annual sales of $7 billion, and an R&D spend of 4% of sales.

The development of heating solutions is highly innovative, entailing both hardware and software development using novel technologies. The product development process has for many decades been successfully managed using a traditional Stage-Gate model with clear stages, milestones, gates and deliverables. In recent years, the Agile-Scrum framework had been used to develop the software components within ThermoValves’ heating solutions; but these two approaches had never been integrated.

6.2. The pilot implementation

In mid-2015, the head of R&D at ThermoValves decided to invest in a pilot project to try an Agile-Stage-Gate hybrid model for development of physical products. He realized that market needs were changing faster and faster, partly due to increased global competition, thus necessitating frequent and continuous confirmation of market needs to ensure market success – even after the Development stage was well underway. Thus, he decided to experiment, applying Agile-Scrum to the business’ gating process for physical product developments as well as for software. A pilot project was selected, and sufficient funding provided, for example, training on Scrum, and to cover the extra costs of the frequent market validations. So successful was this pilot project that the company decided to try this novel approach on several other projects, and early results are positive.

6.3. Agile-Stage-Gate – How it works in this firm

Agile-Stage-Gate at ThermoValves allows for early and frequent customer validations of physical and virtual product designs. The main change from the previous method is that design specifications are no longer fixed up front, but are continuously adapted through the design iterations: no longer is there a pre-Development design-freeze! (This is similar to fast-track construction techniques used in the construction industry – a project delivery strategy to start construction before the design is complete).

While ThermoValves’ Stage-Gate system remains in place and unchanged at the leadership team level, a compromise version of Agile-Scrum is used at the operational (project team) level. The new method is deployed from the earliest stage of the project throughout the entire new-product process for all stages of their Stage-Gate model. Two-week sprints are employed to execute both design developments and customer validations (user tests).

As the project progresses through gates and milestones, some design choices are locked-in as manufacturing decisions are made. That is, due to long lead times in manufacturing, some hardware choices must be frozen earlier than software choices; the latter tend to stay open for longer, allowing more iterations with customers on software and after the physical design is decided.

Project teams have so far been dedicated and co-located, with their own project rooms, and use visual, physical scrum boards. Stand-up meetings or scrums are conducted daily within the teams at their scrum boards, facilitated by the scrum master. ThermoValves chose to not deploy a “pure” product backlog, but rather a flexible list of product requirements in prioritized order (Different words are used). This has worked well, since it better suited a development pipeline consisting of both

Please cite this article as: Cooper, R.G., & Sommer, A.F., Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive, Industrial Marketing Management (2016), http://dx.doi.org/10.1016/j.indmarman.2016.10.006
Agile and traditional projects. And rather than a product owner, ThermoValves decided to keep the existing role of the project leaders as owner in order to avoid role confusion with roles in other projects.

Customer reviews are conducted regularly, but not as part of the sprint review at the end of each sprint. Rather, conducting user tests is deemed a significant task as part of some sprints, with test results merely presented at the sprint review. This too has worked successfully and is perceived as a core benefit from this modified Agile-Stage-Gate system, permitting more in-depth tests and design iterations.

The gates in the system remain the same as in the traditional Stage-Gate model: the regular Go/Kill decision points. The main difference is the acceptance of a higher level of ambiguity regarding the final product. But Go/Kill decisions now are thought to be more fact-based, the result of frequent user tests which provide better insight into customer acceptance and ultimate product success.

6.4. Benefits and challenges at ThermoValves

The pilot project lived up to expectations in terms of being more adaptive to market needs via customer validations: The product design indeed underwent significant alterations to accommodate customer inputs. The project team leader indicated that market success would have more difficult to achieve without these product modifications made during the early product design phases, and that these changes were made possible by using the Agile-Stage-Gate system.

The team leader also realized some unexpected benefits when interacting with internal stakeholders (management) who normally provide inputs to the product specifications. These stakeholders were relieved not to have to provide and approve the entire product specification at the outset, but were able to adapt and learn with the development team as they went through the design iterations.

The greatest challenge came from developers who were frustrated by not having a frozen product specification. Some disliked having to iterate and involve people outside the team, rather than being able to lock themselves in a room for six months and come out with a final product!

6.5. Learnings and advice from ThermoValves

The resounding advice from the business’ leaders at ThermoValves is simple: “just try it!” They recommend that one begin with a selected pilot project that receives the right level of management attention and resources. Other recommendations include adapting the model to suit the context of the organization (some modifications to the system may be necessary, as happened at ThermoValves) and to provide for open dialog throughout implementation to deal quickly with issues and challenges.

7. Resolving difficult issues when implementing Agile-Stage-Gate

Questions arise at Agile-Stage-Gate conferences that identify thorny issues and apparent inconsistencies in merging the Agile and Stage-Gate approaches for manufactured products. These inconsistencies and issues are partly the reason for management’s initial skepticism. Some issues are based on misinformation or a lack of understanding of the new system; others are more profound and require adroit actions. In this section we deal with ten of these implementation issues and solutions.

7.1. Either or both systems?

Agile and Stage-Gate have been found to be quite compatible for B2B firms, and are not substitutes for each other nor mutually exclusive. Thus it’s not a matter of “either/or”: The hybrid model integrates the best of both systems. In practice, Agile is typically implemented into an already-profitable Stage-Gate system, as at ThermoValves. Thus Agile is not a solution to a poorly designed or badly implemented gating system; rather Agile supports an already functional system by providing faster response to change, increased visibility, and increased flexibility.

7.2. For which projects?

Agile-Stage-Gate is designed to handle more dynamic development projects facing fluid markets and changing customer needs and requirements; information uncertainty and ambiguity is also well-handled by the new system. Thus not every development project may be a candidate for Agile-Stage-Gate. Moreover, the requirement of dedicated resources also precludes many projects. Thus, most firms limit the application of Agile-Stage-Gate to their most important, larger, more innovative or riskier development projects. For example at 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, 2014). Similarly, in the case of the U.S. remote control manufacturer cited above, 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% of the projects in their development pipeline (Cooper, 2016).

7.3. For which stages?

Agile is most often implemented first in the technical stages of Stage-Gate, for example for the Development and Testing stages in Fig. 1. Usually it is the technical people that first hear about Agile: Many companies report that positive results achieved from using Agile in their software development projects initially inspired technical people to try the system for physical products. Additionally the fact that Agile iterations result in working or demonstrable versions of the product well suits the technical phases of a new-product project. Finally the fact that technical people are more likely to be dedicated to a single project makes implementation easier here from a resourcing standpoint.

Beyond the obvious technical stages, Agile-Stage-Gate has also been found to work well in other phases of the project, as at ThermoValves. Indeed, early adopters report that this new approach should be applied across the entire project in order to achieve maximum benefit, including the earlier stages, Ideation, Concept and Business Case in Fig. 1, and even for the Launch stage; for example, GEMBA Innovation in Denmark utilizes both ideation and concept sprints in their version of Agile-Stage-Gate (Vedsmann et al., 2016). Other functional areas, such as marketers or manufacturing engineers, may find it more difficult initially to adapt to the Agile way of working and to commit dedicated effort to the project; but given solid training, an effective change management effort, and proper project prioritization and resource allocation, the benefits of using Agile-Stage-Gate across the entire project soon become evident. Dedicated time is clearly an issue for some departments: For example, marketers, who always seem to over-multi-task, might commit to a dedicated effort for a two-week sprint on a project (that is, spend 100% of their time on the one project, perhaps doing a market analysis or voice-of-customer study).

7.4. Project team composition

In some firms using traditional gating methods, the project is handed off to a “commercialization team” partway through the project. But in so doing, one loses momentum, knowledge, accountability and ownership or passion for the project! By contrast, in Agile-Stage-Gate, the core team remains intact from beginning to end of the project: that is, the team remains on the field from beginning to end of game.

As the project progresses, new people may join the project as needed, for example manufacturing and sales people may be added towards the commercialization phases. Work intensity by functional
area will obviously shift during the project – product designers or developers have a more intense role during the middle phases of the project – but in order to ensure continuity, ownership and accountability, the team members remain “on the team” throughout the project and up to the post-launch review.

Team members in Agile-Stage-Gate have a desired profile: They are experts in their functional area but also have a broad set of general capabilities; for example, a product designer (one’s expertise) who is also able to provide support in product testing. Working together, such team members function as a true team: they are able to “play each other’s position on the field” rather than each team member just doing their own finite piece of the project alone. In practice however, people with such profiles are much sought after, and a team of such experts with generalist capabilities may be hard to find. But this ideal team member profile remains the goal in order to foster a true team approach.

7.5. The role of gates

A commonly-asked question by people new to Agile-Stage-Gate is: “Are there still gates in the system?” One might expect that the notion of a gate is “quite rigid” and thus contradicts the Agile mindset of “being flexible”. But not so: Agile is very consistent with tough Go/Kill decisions on projects – gates with teeth. The flexibility of the Agile system lies with the product deliverable; that is, there is flexibility about what is delivered, not in the Go/Kill decision structure!

The main difference and the flexibility in Agile-Stage-Gate versus traditional gating systems is in recognizing and dealing with the reality that many projects, especially more innovative ones, have high levels of ambiguity and uncertainty at their beginning – that it is often impossible to know everything up front. Thus, uncertainties about product deliverables is managed by accepting and even supporting product and project scope changes, as long as these changes do not affect the overall project plan, budget and project’s financial attractiveness. If they do, the project may be flagged for re-evaluation and possibly terminated (with a new project based on the new prerequisites often initiated). Recall from Section 4.3 above that the product backlog (product definition) is very dynamic; indeed the product may be only 20% defined at the point of project approval (the often-cited range is 20–90% defined at project approval, but 40–60% defined is more typical). Thus the project team is allowed to learn and adapt during the project, but within limits.

Gates continue to play a vital role in Agile-Stage-Gate, however, as at ThermoValves. They allow senior management to periodically review the project, to kill weak projects and reallocate resources to better initiatives, and most important, to ensure that the necessary resources are committed to so that the project can move forward. Note that gates are not just a quality check-point, but they are also a resource commitment decision. In this way, the project team secures the needed personnel, time and funding to continue their work in an accelerated fashion. Note that insufficient resources and project team members spread too thinly is one of the major causes of long times-to-market (Cooper, 2013).

Gates also allow senior management to track the progress and on-time performance of the project: when to deliver the product on the “long-term horizon” scale thus remains defined and a key part of Agile-Stage-Gate. While each sprint has its own “plan” and timeframe and thus is flexible, the longer term timeline remains relatively stable. For example, in one process controls firm, the Agile project leader was asked by senior management: “When will this product be done – when will it be on the market?” The answer the executive received was: “It will be done when it’s done!” This answer is clearly unacceptable and inconsistent with Agile-Stage-Gate. Indeed projects within this new system are managed with timelines or Gantt charts complete with milestones that clearly define the progress of the project over a pre-determined longer term horizon. Admittedly these timelines are fairly high level, and the details may change over time, but the longer term plan and timeline remain intact.

7.6. Hardware versus software gates?

Another common gate question is: “Should there be different gates for different deliverables, for example, have separate software and hardware gates?” Most users find that having multiple gates for different facets of the same project are troublesome and not beneficial: Such structures only add complexity and ambiguity to the project. And consider the possible and ridiculous consequence of a new product involving both hardware and software, where the software gatekeepers kill the software part of the project at their gate meeting, when a few weeks earlier the hardware gatekeepers gave it a positive decision. Both parts of the project depend on each other, hence the need for a common gate structure. The only time that a separate gate structure may be warranted is where the two developments are not totally inter-dependent.

7.7. Plan-based versus plan-on-the-fly and fixed or flexible budgets and times?

An apparent contradiction within Agile-Stage-Gate is typified by the question: “How can a project be approved for Development if you don’t know the product definition and consequently don’t have a solid development plan (Gantt chart) – and without a Development plan, is there no development cost estimate? Surely an executive cannot be expected to approve a development project when the development cost is variable!” This question, in one form or another, is the result of trying to reconcile planned-based approaches (traditional gating) with plan-on-the-fly methods (Agile).

The “product backlog” partly fills the role of the traditional “product definition”, says Peter Fuerst, new product expert at Five I’s Innovation in Germany (Fürst, 2016). Note that the product backlog is not locked-in early in the project – it varies depending on uncertainty. And the product backlog (product definition) evolves over time as the project progresses. Nonetheless a higher level albeit somewhat tentative product backlog is required at the outset, and provides sufficient definition for planning and project approval.

The project plan presents a similar dilemma: One goes from a product backlog to a “schedule” of tasks for the entire project (both are created at the beginning of Development, based on best estimates at the time). The schedule is similar to a traditional product development plan (a Gantt chart); but it must necessarily be very tentative and high level at the Go to Development gate in Fig. 1, as noted in Section 7.5 above.

From this schedule, a budget or estimated development cost is determined. And it too will be somewhat tentative, an estimate. Given the high probability of change, the tasks within stages and thus the project’s budget are certainly variable during the early phases of the project. But as uncertainty decreases over time, the schedule or timeline and the budget both become increasingly fixed or stable. A rule of thumb is that one can lock-in the final budget by the end of the Development stage in Fig. 1.

7.8. The need for up-front homework and early-stage voice-of-customer work

A worrisome potential consequence of Agile-Stage-Gate is the mistaken belief that the front-end homework and voice-of-customer work is no longer needed – that one can move into a significant project into the development stage without the traditional due diligence in place. The argument is that “things change”, that Agile-Stage-Gate allows for change, and that attempts to rigorously define the product and the plan in advance – only to see them change later – are a waste of time.

An illustration of how things might go wrong (words of the scrum master): “The company (a major construction equipment manufacturer)
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: VoC and market requirements, and technical concept capabilities. Thus project teams were rushing into the development stage without knowing how the concept would perform technically or if it would meet the customer requirements” (Cooper, 2016).

When Agile-Stage-Gate was introduced in this firm, the project team members no longer had to spend as much time struggling, often in vain, to do due diligence and VoC work before project approval. Taken to an extreme, and being design engineers who did not feel comfortable doing VoC work, the new system might provide a convenient excuse not to do any of the needed and valuable homework – that is, to promote ill-conceived short-cuts and intellectual laziness. The temptation to omit front-end homework was certainly an issue here: were it not for the tough-minded scrum master, who ensured that his project teams still did the due diligence within their abilities, no doubt vital actions would have been skipped.

Indeed there are often questions about whether market analysis and VoC in the early stages of Agile-Stage-Gate are still needed in order to understand the customer. For example, Steve Jobs, never a proponent of traditional early-stage market research, famously said, “People don’t know what they want until you show it to them” (Isaacson, 2011, 527); the implication was that early VoC studies which ask people what they want or need can be replaced by product iterations and customer tests much later in the project. Thus, one view is that these VoC tasks might be considered unnecessary, since users are involved in sprint reviews, and the project team learns as the project moves forward. And even early-stage technical assessment may no longer be warranted, say some, simply because the product definition is likely to change dramatically as the project moves along; so what’s the point in doing a very detailed technical assessment on a product that never will be.

These arguments, while intuitively appealing, are quite wrong: Agile actually encourages multiple approaches to understanding customer and user value and to getting the product right. Thus, the new system still requires proper market analysis and VoC in the early stages in order to provide the necessary foundation for the project and to steer the project in the right direction. And VoC is the basis for developing the vital “product backlog” (see Sections 4.3, 7.5 and 7.7), which in turn leads to the “schedule” (project plan) and development cost estimates, all essential items to securing project approval. Agile-Stage-Gate also requires early-stage technical assessment, even though the product definition may change, so that developers can be appraised of technical options and technical risks before charging into the Development stage.

One benefit of Agile-Stage-Gate is that the homework – both technical and marketing – need not be excessive, thus avoiding the frequent complaint in traditional systems of “paralysis by analysis”. That is, the information gleaned pre-Development does not have to be perfect. The initial information available and assumptions at the beginning of the Development stage in Fig. 1, the result of early stage homework, will be verified many times via the sprint iterations in the stages that follow – by building versions of the product, testing them with users or customers, securing regular feedback from lead customers in the market, and also testing for technical proof of concept.

7.9. New roles and old roles

Agile introduces some new roles and role terminology. But some users of Agile-Stage-Gate stay with traditional roles and terms. For example, some user-firms, such as ThermoValves, do not use the Agile term “product owner” as highlighted in Section 4.2 above. Rather, they have a “project leader”, responsible for leading the project. Larger and more complex projects also might have an on-team “process manager”, whose role is similar to that of the traditional project manager.

Although the two terms are similar, the project leader is quite different from a project or process manager: A project manager is typically responsible for developing timelines, ensuring the team keeps to the project plan, practices sound project management methods, and brings the team together for necessary meetings – somewhat of a “mechanical” but necessary job. The project leader, by contrast, is a person whose job it is to drive the project forward to its ultimate goal, ensure stakeholder commitment, generate a clear value proposition, and generally champion the project. The project leader is often likened to the entrepreneur in a start-up business, with the project being his or her start-up. Characteristics of effective project team leaders include: credibility, enthusiasm, entrepreneurship, people skills, project management skills and project knowledge.3

In Agile-Stage-Gate, both project leader and project or process manager are valuable and necessary roles, and are preferably delegated to two different people: a product owner or project leader, and a scrum master. The project manager or scrum master is generally able to support more than one project at a time, depending on the project size and complexity. For smaller projects, sometimes the team leader fills both roles – leader and project manager.

7.10. Managing the development pipeline of Agile projects

With teams progressing so quickly, and so much happening fast, the potential for chaos or anarchy exists; so how does a senior manager stay on top of the entire development pipeline of projects? One very positive facet of Agile-Stage-Gate is its very visual nature: The system generates excellent visibility of the progression of tasks within and across projects. The visual boards contain the details of exactly what activities or tasks are to be conducted during the sprint or iteration, which activities are underway and by whom, and which tasks are done. At any time, the portfolio owner can enter a project room and instantly see the progress within the sprint, and the status of the product backlog and prioritized requirements – and can do this for all projects in the development pipeline.

Even more dramatic is where a virtual software tool is used to create the visual boards. Such software not only enables companies to manage individual projects, but permits one to view the full project portfolio instantaneously and in real-time. With such a setup, managers can run analytics across the portfolio of all projects, for example:

- to reveal differences in the flow of activities across projects, showing all activities that are delayed by more than a week, thus pointing to the need for immediate action;
- to show the types of tasks undertaken by which people and departments, information useful for employee development; and even
- to explore real-time resource capacity and usage (if using story points or similar weightings of probable resource consumption by tasks) in order to improve capacity planning in the portfolio.

As a result, in Agile-Stage-Gate, the portfolio owner becomes more proactive, and takes action with the project leader when something troublesome is spotted or where an intervention might be needed. Traditionally, the portfolio owner responded reactively, this reaction based on periodic updates from project managers or monthly project reviews. Such a reactive information flow can delay the necessary intervention by weeks. By contrast, with the enhanced visibility inherent in Agile-Stage-Gate, problems are immediately pinpointed and interventions take place right away. Sometimes interventions occur before the problem even materializes: for example, intervening

---

3 Based on a content-analysis review of articles appearing over a 20-year period in the Journal of Product Innovation Management, that probed for the characteristics of effective project team leaders

Please cite this article as: Cooper, R.G., & Sommer, A.F., Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive, Industrial Marketing Management (2016), http://dx.doi.org/10.1016/j.indmarman.2016.10.006
when a trend indicates problems are inevitable, such as when project velocity decreases or core activities are delayed.

8. Why the hybrid Agile-Stage-Gate system works

The benefits of Stage-Gate have been well-researched and and its widespread use documented. Less well-known to B2B manufacturers is the benefits of Agile. The limited experience with Agile-Stage-Gate hybrid development models suggests that manufacturers can indeed benefit greatly from this new hybrid approach. Here are some conclusions about why and how:

8.1. Deals with uncertainty and validates assumptions for very innovative projects

Most firms’ new product processes emphasize extensive front-end homework to define the product and to justify the development project, before Development gets underway. Indeed, robust up-front homework and VoC work early in the project are consistently cited as keys to new-product success (Cooper, 2013). But not all projects are quite so definable. Indeed, in some highly uncertain projects – those in new markets and using new technologies – no amount of VoC work, technical assessment or market analysis will deal with the uncertainties and validate all the assumptions prior to the Development stage. Understanding what the customer values and what will work technically only comes about through experimentation.

The rapid sprint-iterations in Agile-Stage-Gate encourage experimentation and testing – build something, test it with the customer and in the lab, and then revise one's thinking. The product may be only 40% defined on entering the Development stage, as noted in Hewlett Packard’s model (Maccormack, Crandall, Henderson, & Toft, 2012), but evolves and solidifies via these iterations. In this way, key assumptions are validated and major uncertainties dealt with, but in real time and as the project moves along. Thus, understanding product requirements and envisioning a technical solution does not occur before Development, but in Agile-Stage-Gate is done as part of the Development and Testing stages of the project – learning on the fly.

At the same time, it’s a learning process for the customer too. Needs are often difficult for customers to articulate, especially at the outset of a development project and in the case of more innovative products and solutions. But seeing and critiquing prototyped products along the way helps customers to understand and define their own needs.

8.2. Adaptive – Deals with changing requirements

When customers’ needs change, or a new product requirement becomes evident partway through Development, traditional gating models, with fixed product definitions, fail to respond easily and quickly: they are simply not very adaptive. For example, once product specifications are “frozen”, any design change request is viewed quite negatively.

By contrast, by building very early product versions or prototypes via the sprints – a model, computer generated graphics, or a rapid prototype – Agile-Stage-Gate is more adaptive: if product requirements change, then needed modifications to the product’s design can be made earlier during the Development stage when the cost of change is lower, much like a strategic pivot in the Lean Start-up method (Ries, 2011). For example, a study of two major B2B European manufacturers revealed that on average, between 3 and 4.5 versions of the product – from early-model through to prototype – were presented to validate the design with customers through the Development and Testing stages (while product ideation-and-design contractors, such as IDEO, iterated on average 15 times with the customer per project!) (Sandmeier, Morrison, & Gassmann, 2010).

8.3. Focuses teams, accelerates development, improves communication

Agile-Stage-Gate project teams are dedicated to the one project to ensure adequate resources to get the work done. In traditional new-product development, only 11.4% of average firms have focused (dedicated) project teams (Cooper, 2013), and only for some projects. But Agile-Scrum places such emphasis on this dedicated team facet that teams really are dedicated for every major project. This one step alone increases development speed dramatically and may also improve quality of execution of key tasks.

Additionally, time boxed-sprints, and even time-boxed tasks within sprints, bring a sense of urgency to the development project. In Agile-Scrum, all events are time-boxed, such that every event has a maximum duration: That is, once a sprint begins, its duration is fixed and cannot be lengthened (ScrumInc, 2013). Project teams commit to certain deliverables at the beginning of each sprint, and then are expected to deliver within the time frame agreed. And it’s vital to have exactly the right time available (including a buffer) to undertake the required tasks. (In practice, the project team estimates the times for all tasks at the beginning of a sprint, allowing six hours per person per day available, which assumes 75% efficiency).

The key here is to have a steady flow, and to avoid doing things in haste or in a last-minute panic (the negative term “feature cramping” is used in the IT world). Additionally, the notion of a steady, strong and responsive heartbeat creates a rhythm for the project team, and keeps moving the project along at a sure and steady pace – momentum is maintained.

Finally, dedicated teams (not spread across other work or other projects), a dedicated team room where the entire team resides, and face-to-face daily scrums all contribute to much improved team communication. Every study of Agile (whether for IT or physical products) reports this benefit.

References


Furth, P. (2016). Based on private interviews with Peter Fürst, Agile-Stage-Gate expert in Europe and managing partner of consulting firm Five’s Innovation Management.


