New-Product Portfolio Management with Agile: Challenges and Solutions for Manufacturers Using Agile Development Methods

Robert G. Cooper and Anita Sommer

When Agile Development and the traditional way of measuring, evaluating, and managing projects clash, companies must reassess how new-product portfolios are managed, how go/kill and prioritization decisions are made, and how performance is measured. This article explores the new solutions and emerging challenges at the boundary between Agile project management and portfolio management, and suggests tools and approaches to deal with these new challenges.


© Cooper & Sommer
New-Product Portfolio Management with Agile
Challenges and Solutions for Manufacturers Using Agile Development Methods

Robert G. Cooper & Anita Friis Sommer

To cite this article: Robert G. Cooper & Anita Friis Sommer (2020) New-Product Portfolio Management with Agile, Research-Technology Management, 63:1, 29-38, DOI: 10.1080/08956308.2020.1686291

To link to this article: https://doi.org/10.1080/08956308.2020.1686291

Published online: 18 Dec 2019.

Submit your article to this journal

View related articles

View Crossmark data
New-Product Portfolio Management with Agile
Challenges and Solutions for Manufacturers Using Agile Development Methods

Agile–Stage-Gate hybrid product development methods demand new tools and processes for portfolio management.

Robert G. Cooper and Anita Friis Sommer

OVERVIEW: While Agile development approaches to physical products have allowed companies to respond more quickly to change and increase R&D productivity, implementing these approaches also poses new challenges for new-product portfolio management. When Agile and the traditional way of measuring, evaluating, and managing projects clash, companies must re-assess how portfolios are managed, how go/no-go and prioritization decisions are made, and how performance is measured. This article explores the new solutions and emerging challenges at the boundary between project and portfolio management and suggests tools and approaches to deal with these new challenges.

KEYWORDS: Agile–Stage-Gate, Portfolio management, Agile portfolio, Scrum, Project management

In recent years, a growing number of leading manufacturers have implemented Agile development methods for new-product development. These early adopters have typically embedded Agile methods within the stages of their traditional Stage-Gate processes. For example, in a German study of 138 manufacturers, 100 percent of the “best performers” implemented Agile sprints between traditional process gates (Invention Center 2018).1 Many have found that for high-uncertainty projects, implementing Agile methods alongside traditional Stage-Gate processes to create an Agile–Stage-Gate hybrid yields faster response to change and higher R&D productivity (Cooper 2016; Cooper and Sommer 2016a, 2018; Schmidt, Weiss, and Paetzold 2018; Sommer et al. 2015).

Besides these benefits, the method also provides solutions to many of the problems of new-product portfolio management; it also creates some new challenges. Portfolio management for new-product development is about making resource investment decisions (go/no-go and project prioritization) to maximize the value of the portfolio, create the right mix of projects, and balance resource needs with available resources. A lack of robust data and the failure to kill projects can frustrate these goals. Agile–Stage-Gate helps address both these problems. The frequent iterations and testing of demos and prototypes with customers and management, which are characteristic of Agile–Stage-Gate processes, provide more and better data on customer experience, product acceptance, expected sales, and even manufacturing costs. And management’s constant engagement with the project team, coupled with better data, heightens the odds that bad projects will be spotted earlier and killed immediately, freeing resources for other, presumably stronger, projects and thus strengthening the portfolio.

The Agile–Stage-Gate method also introduces some new challenges to portfolio management. One is the difficulty of making investment decisions when the product definition and project plan are fluid and constantly evolving. The iterative nature of the decision process coupled with the project’s ambiguity renders traditional financial tools, such as net present value (NPV), less relevant. A second issue is finding the right metrics to gauge the health of the portfolio and the progress of projects in the portfolio. Traditional progress metrics are often too coarse to capture the full range and pace of the work that is now being done in Agile environments.
metrics, such as percent on schedule, are rendered meaningless by the constantly changing project plans and schedules that are part of Agile processes. Standard budgeting methods may also be challenged by Agile approaches, which create dynamic portfolios, as projects are continually being redefined, added, or dropped. Although some of these challenges admittedly require new thinking, others can be addressed with new applications of existing tools.

The new challenges that Agile–Stage-Gate brings are considerable and require new solutions. However, Agile–Stage-Gate also offers solutions to traditional portfolio challenges that deliver immediate benefits. Approaching the hybrid system thoughtfully, with an eye to both solutions and challenges, can help smooth the path to adoption.

New-Product Portfolio Management Challenges

Portfolio management is about translating innovation strategy into investment decisions on specific new-product projects. It is a dynamic process, involving continual reconsideration of the business’s active projects and reallocation of resources to fit evolving innovation priorities; new projects must fit into the portfolio’s priorities; and existing projects must be extended, accelerated, or killed (Cooper, Edgett, and Kleinschmidt 1999; Kahn et al. 2013; Meifort 2016).

Effective portfolio management yields a better balance in the portfolio, a higher proportion of high-value projects, and better balance between projects and resources (Cooper, Edgett, and Kleinschmidt 2004; Edgett 2013). The benefits are evident: businesses with systemic portfolio management systems in place tend to perform better at new-product development overall, delivering a higher proportion of product launches that meet sales and profit targets, higher new-product profitability, and open new windows of opportunity. Only a minority of businesses achieve these outcomes, however, as many struggle with the challenges presented by portfolio management. Chief among these challenges are overloaded pipelines and a lack of reliable data on which to base portfolio decisions.

Too Many Projects

The first and perhaps most frequently cited challenge in portfolio management is too many development projects in the pipeline (Dalton 2016; Edgett 2013; Thomke and Reinertsen 2012). As a result, resources become too thinly spread across too many projects, so that every project is underresourced (Elonen and Artto 2003). Lack of needed resources is one of the fundamental reasons projects take so long to get to market, and it is also blamed for shortcuts in the product development process, taken to save time, that come back to haunt the project team (Cooper, Edgett, and Kleinschmidt 2004).

One of the reasons for too many development projects is the failure to kill bad projects in a timely fashion. Many projects are approved initially, since early resource commitments are low and good reasons exist to do each one of them. But projects that look good at their beginning often lose their luster over time as new information becomes available. Testing of underlying assumptions, customer feedback, and the search for technical solutions all yield new insights that are sometimes not as positive as the initial proposal suggested. However, the project has momentum, the result of company politics, sunk-cost reasoning, emotional investment, or the lack of a systematic review process; and momentum can carry the project through the development process regardless of its merit (Aberdeen Group 2006). Stopping projects in mid-development can be like trying to halt an express train at full speed. With too many projects initially approved and the failure to cull out the weaker projects over time, the result is inevitable: an overloaded development pipeline, with too many projects for the resources available.

A Lack of Data Integrity

A second traditional portfolio challenge is a lack of data integrity (Aberdeen Group 2006; Edgett 2013). Not only does unreliable data lead to poor decisions, it also contributes to pipeline overload because it underlies the failure to kill underperforming projects.

Key go/kill decisions must be made as a project moves into and through development. Portfolio managers look to rigorous economic evaluation methods, such as NPV or payback analyses, to sharpen these go/kill decisions. But rigorous analytical methods require valid inputs—estimates of sales, pricing, manufacturing costs, and development costs. When these data are not correct or not verifiable, no financial model will give a reliable result.

Improving data integrity, especially early in the project’s life when key approval decisions must be made, is thus a priority for product developers, not only to support portfolio decisions but also to get product design, pricing, and launch right. The fuzzy front end—that phase of a new-product project between ideation and the beginning of serious investment in development—has long been recognized as a vital part of the innovation process (Koen, Bertels, and Kleinschmidt 2013; Markham 2013); it is here where much of the information needed for decision making should be obtained. Traditional gating processes typically build in an early stage or two, during which the project team is expected to perform the market and technical assessments needed to provide the information required to move forward.

That’s the theory; the reality is quite different. The evidence strongly suggests that this front-end homework is
not executed well, if at all, leading to data integrity problems. The majority of projects approved for development—58 percent, according to Barczak, Griffin, and Kahn (2009)—fail, in spite of the homework that was supposedly done to ensure their viability. Many studies show that the reasons for new-product failure are largely the fault of poor front-end work. For example, a recent summary of studies identifies seven common failure reasons; the first four are a lack of understanding of customer needs and wants; a project that addresses a nonexistent customer problem or targets the wrong market; and incorrect pricing (Grabowski 2015; Lane 2017). These failures all have their roots in the front-end homework phase. Even when the homework is done, many studies find that it is not done well; poor quality in front-end activities is common (see, for example, Cooper, Edgett, and Kleinschmidt 2002a; Grabowski 2015; Lane 2017). For many, the front end has become a paper exercise simply to get one’s project approved. The widespread deficient execution of front-end activities helps to explain why a lack of data integrity remains a major challenge to portfolio management and to new-product development in general.

**Agile–Stage-Gate Solutions**

A hybrid Agile–Stage-Gate system that integrates Agile methodology into a Stage-Gate process can help alleviate these challenges, by creating more opportunities to evaluate a project and by generating better data to support assessments of a project’s viability (see “An Overview of Agile–Stage-Gate, right).

The classic Stage-Gate process is typically linear, with the project team moving from task to task and from stage to stage according to a defined timeline and project plan; there are no iterations and little or no experimentation. Testing with end users or customers along the way, before the final “field trial” pre-launch stage, is unusual at best.

In the Agile–Stage-Gate model, the front-end homework tasks—the market, competitive, and customer analyses work and development of the business case and product definition—are much the same as in the classic gating model. But the nature of these tasks is different; Agile iterations and testing with customers provides feedback that is looped into the analysis and definition even as the business case is being developed. Thus, initial assumptions about market acceptance, customer value, and even manufacturing costs are validated often, early, and cheaply. Further, go/kill decisions can be made more frequently than in the classic gating system—potentially at the end of each iteration when results are demonstrated to management.

The iterations continue through the development stage, with each iteration producing a more tangible version of the product; the process moves from rapid prototypes to crude working models and prototypes, to working prototypes, each version closer to the final product, and each version tested with customers or users. This continual customer feedback drives more reliable estimates of market acceptance, expected sales, customer value, and possible pricing. Developing these successive product versions also provides technical proof of concept and builds insights into probable or best technical solutions, feasibility, and manufacturability, again improving estimates of further development costs, manufacturing costs, and the investment required for production. The ultimate pre-launch test is the minimum viable product (MVP), a new product version that has sufficient features to satisfy early adopters; it provides solid information about customer reaction and value, as well as realistic delivery cost information (Reis 2011). These continual validations of the product’s design and economics through iterations and demonstrations not only

---

**An Overview of Agile–Stage-Gate**

Stage-Gate is a popular method for managing new-product development that breaks the innovation process, from idea to launch, into a set of about six defined stages (Cooper 2017). Each stage has a clear purpose and a set of expected activities and resulting deliverables; at the conclusion of each stage, the project must pass through a gate—a decision point where a go/kill decision is made and, if the decision is go, resources are committed for the next stage. This structure is intended to manage risk by breaking the product development process into a series of incremental bets based on the progress made in each stage. The gates provide defined points at which go/kill decisions are made. Stage-Gate is not a project management method per se, but rather a high-level model to guide the product development process.

Agile is a set of project management methods and a mindset that developed in the software industry to promote rapid response to change and constant customer collaboration and feedback. In an Agile–Stage-Gate hybrid model, Agile methods are employed within each stage of a Stage-Gate model, replacing traditional planning methods such as Gantt charts, timelines, and critical path plans (Cooper 2016; Cooper and Sommer 2016a, b, 2018; Sommer et al. 2015).

In such a hybrid process, each stage in the Stage-Gate process comprises a series of time-boxed iterations, called sprints, each of which lasts two to four weeks. Each sprint is planned in real time, on the fly, in response to the outcomes of the previous sprint, yielding a process that is highly responsive and adaptive. At the end of each sprint, the project team produces a tangible outcome—results of a study, a concept or prototype, design drawings, a prototype or model—something that can be demonstrated to stakeholders, including customers and management, for validation and feedback. The feedback from the outcome feeds into the next sprint. The end of a sprint may also present an interim opportunity for a go/kill evaluation, particularly if the outcomes do not meet expectations.

Each sprint ends with a retrospective meeting, during which the team determines whether the planned tasks for the sprint have been completed and considers how the team can work together better in sprints to follow. As in Agile for software projects, the project team is ideally dedicated 100 percent to the one project and collocated in one team room. Daily stand-up meetings (also called scrum meetings) keep the team on track and facilitate communication.
Another key challenge is in the area of portfolio metrics. In the traditional Stage-Gate model, senior leaders viewed the portfolio from a plan-based perspective. Many companies gauge the progress and continued value of a project by whether it meets milestones and stays on budget, and they use tracking metrics such as traffic light indicators to indicate whether a project is hitting milestones and staying on budget. If the project is on schedule and on budget, then the default decision is usually to continue. But merely staying on schedule is a poor reason to continue spending on a project; indeed, a project can go off-schedule for reasons that actually contribute to its value.

Adopting Agile–Stage-Gate requires senior management to adopt a new mindset and use different metrics to gauge the health and progress of the portfolio. Useful metrics escape the plan-based perspective and emphasize value creation as a function of time, rather than on progress against a timeline. Admittedly, these kinds of metrics can, and perhaps should, be applied to traditional projects as well. Some traditional projects have rather fluid timelines, presenting situations where more dynamic metrics focused on value creation might also be appropriate.

In Agile–Stage-Gate, the project plan evolves over time, along with the product definition. Traditional Stage-Gate envisions product development as a smooth trajectory with a fixed target and clear progress metrics; but in an Agile project, the trajectory is constantly changing—the project plan is fluid and the target is evolving (Figure 1). Measuring progress under these dynamic circumstances is not straightforward. “On schedule,” “on target,” or even “on budget” are meaningless concepts, because there is no firm schedule, the product definition (target) is evolving, and the budget is fluid. To some extent, senior management must learn to live with this new reality: Agile projects simply cannot offer the clarity of traditional progress metrics. Different tools and
metrics are needed to support decision making with uncertainty and fluidity. Although some new metrics are needed, the burndown chart, the Productivity Index (PI), and the expected commercial value (ECV), but significantly modified, provide some needed measures.

**The Burndown Chart: Tracking Progress Against a Fluid Timeline**

The burndown chart is a standard Agile tool; it captures the project team’s speed and progress by showing how fast the team is “burning through” the tasks agreed to for a given sprint (Cooper and Sommer 2016b). Usually, a burndown chart is developed for a single sprint, for which the plan is fixed. A few firms, however, are experimenting with a burn-down chart for the entire stage, across many sprints (Figure 2). In this case, the team develops its project backlog (a visualization of all the tasks to be completed in the stage), a very high-level and tentative plan for the stage, and an estimate of the number of sprints required to clear the backlog.

But things change as the stage progresses: tasks get added to the backlog, other tasks are subtracted, and some tasks take longer than expected. To accommodate these changes, the number of sprints required changes and thus the plan for the stage evolves. As a result, the burndown chart for an entire stage is dynamic. However, the number of sprints remaining to be done can be tracked at any point during the stage, and thus provide a gauge of progress toward the target that accommodates the dynamic nature of the process. A similar metric is the proximity metric—the percent of the stage that is completed, defined as

![Proximity Metric](image)

$$\text{Proximity Metric} = \frac{\text{Sprints Done}}{\text{Sprints Done + Estimated Sprints Remaining}}$$

Sprints remaining and the proximity metric, both derived from the burndown chart, are both useful gauges of progress when the overall timeline keeps changing.

**The Productivity Index: Monitoring Economic Health in a Dynamic Environment**

Another useful metric for Agile projects is the PI, which gauges the economic well-being of a project in real time (Matheson, Matheson, and Menke 1994). The PI was first proposed many years ago, and some firms have adopted it for traditional projects, but because it is a dynamic, real-time metric that emphasizes value creation over time, it is particularly applicable to Agile projects. The PI measures the value gained from additional work spent on the project or, alternately, for every additional dollar spent on the project. For projects with relatively certain financial estimates, the PI is defined as:

$$\text{PI} = \frac{\text{NPV}}{\text{Money remaining to be spent}}$$

or

$$\text{PI} = \frac{\text{NPV}}{\text{Number of work days left to be done}}$$

NPV is the traditional measurement of the economic value of the project to the business. Here, the NPV is divided by the work remaining to be done—in funds or workdays. Sunk costs (resources already invested) are not relevant to the decision to move forward. The PI is plotted over time, over the life of the project as development proceeds; if the project is proceeding well, the PI should increase, and its curve should asymptotically approach infinity just before launch. Any project whose PI starts to turn downward is a candidate for killing or reprioritization. The PI thus provides a dynamic indication of a project’s merit.

Because it measures the value accrued for every additional resource input to the project, the PI can also show which projects are the most productive, and thus highlight which projects should be higher priority. Since development cost and workdays are usually the constraining resources in any new-product portfolio, the PI can be very useful for ranking and prioritizing projects to maximize the value of the portfolio with resource constraints. In a PI-based prioritization exercise, projects are ranked by their PI values, with high PI projects going to the top of the priority list—these projects will yield the most bang for buck. Note that the PI will tend to favor high-value projects that are closer to
completion (and thus have less remaining to be spent). Caution should be taken, however, to ensure a PI prioritization does not shut out high-potential early-stage projects.

**Expected Commercial Value: Estimating Economic Value for Ambiguous Projects**

While metrics that focus on value creation over time are desirable, accurately estimating the economic value of an ambiguous Agile–Stage-Gate project with uncertain estimates is difficult. Thus, making effective numbers-based go/kill and prioritization decisions is also challenging. A number of project selection tools have been developed for application when estimates of sales and costs are uncertain or fluid (Cooper 2013). These include qualitative scoring models, options pricing, probability-adjusted NPV, Monte Carlo simulation, and Expected Commercial Value (ECV).

Any of these may be appropriate for early-stage Agile–Stage-Gate projects; however, ECV is particularly appropriate for highly uncertain, iterative Agile projects, especially in conjunction with the PI. ECV provides a realistic financial model for handling incremental or stepwise investment in a new-product project, in the form of a decision tree approach that evaluates the investment at a step at a time. The ECV method was originally developed for traditional new-product project evaluation in the specific case of the four-stage investment model typical of classic Stage-Gate (Cooper 2013). However, ECV is applicable when the incremental investment decisions are even more frequent and involve smaller investments. That’s exactly the situation in an Agile–Stage-Gate sprint—the project could be stopped after any of the sprints, and the decision to move forward after a sprint usually only extends to the next sprint or two. Thus, there may be many investment steps and the investment amount at each step is quite small. This is the ideal context for ECV.

In ECV analysis, the simplest case is a one-stage or single-iteration model with two possible outcomes, “good” and “bad” or “success” and “failure” (Figure 3). Here the ECV is simply the expected value of the outcome, ECV = (R x P) − C, where R is the reward or payoff from the project (usually the present value of future earnings), C is the cost of undertaking the project (remaining development and commercialization costs), and P is the probability of success. The traditional NPV would be calculated as NPV = R − C; by including the probability of success, the ECV thus gives a more realistic result.

For the situation with N iterations (Figure 4), if the investment is approximately equal across all iterations (that is, for each sprint), then the investment for any one iteration is C/N. The ECV is:

\[
ECV = (R \times P) - (C \times RMFactor)
\]

Here, as in the single-iteration calculation, R is the reward, P is the probability of success, and C is the cost of the project. The RMFactor is a risk-mitigating factor, calculated as:

\[
RMFactor = 1 / N \times (1 - P) / (1 - P(1/N))
\]

Thus, having multiple iterations in effect reduces the size of the investment required and thereby mitigates risk. (For details on the derivation of the ECV for N iterations, see Cooper 2019; Sommer 2018). The project team can use this equation to generate ongoing estimates of the ECV as the project progresses. Alternately, the team can create its own model or decision-tree diagram, much like Figure 4, with iterations, probabilities, and outcomes that reflect its specific project. The calculations will be similar.²

When there are many iterations, as in an Agile project, the RMFactor ranges from 0.66 to 0.84 when the probability of success, P, is between 40 percent and 70 percent (a typical range for riskier projects). Thus, using an Agile approach with many iterations increases the project’s value by 16–34 percent of the investment, C. Further, C (the remaining cost) steadily decreases with each iteration—in other words, the downside risk is significantly reduced over time. The investment at any decision point before a sprint begins is C/N, so the risk/return ratio is also small and declining for an iterative Agile project.

Calculating the ECV in this way offers a number of advantages for Agile projects:

- **It accommodates project ambiguity.** The ECV method builds in probability estimates in an appropriate way. Any evaluation method that ignores these probabilities, as, for instance, NPV does, in effect assumes a 100 percent chance of success and is inherently invalid for Agile projects.
- **It accounts for the incremental nature of investment decisions.** In an Agile project, investments are made a step at a time; it is an incremental commitment process, much like buying

²One added complexity in using ECV is the need to estimate probabilities of success, both technical and commercial. Project teams can usually arrive at reasonable estimates using a modified Delphi method, in which the team engages in a facilitated group discussion with a few subject matter experts (Cooper, Edgett, and Kleinschmidt 2002b).
a series of options on a property. Thus, one possible outcome at the end of every Agile iteration is a decision to terminate the project. ECV has this incremental commitment and kill feature built in: investment is cut and thus saved in the case of a “bad” outcome. By contrast, NPV is a capital budgeting technique; it assumes the entire investment is made. NPV will thus tend to understate the financial attractiveness of Agile projects.

- **It allows probabilities to be adjusted in real time.** As successive iterations of the project generate more information, the probabilities of success change, up or down. The ECV can be recalculated as probabilities shift to better reflect the real value of the project as it develops.
- **It can be used to prioritize projects.** Traditionally, NPV has been used to calculate the PI for project prioritization. For ambiguous projects, however, when the PI is used to prioritize projects, it should be calculated using ECV rather than NPV.

\[
PI = \frac{ECV}{\text{Money remaining to be spent}}
\]

or

\[
PI = \frac{ECV}{\text{Number of work days left to be done}}
\]

**Strategic Buckets: Dynamic Budgeting for Dynamic Portfolios**

Introducing Agile into the Stage-Gate process also creates a portfolio-level challenge, namely, the increased portfolio churn and the resulting dynamic portfolio. Because Agile sprints make new information available more frequently, projects whose value is declining can be spotted and killed more quickly. Killing bad projects releases resources to more worthwhile projects, which can be added to the portfolio.

As a result, traditional calendar-based budgeting becomes difficult—and perhaps even meaningless. Fixed annual budgets that allocate resources to a particular project conflict with the dynamic portfolio of projects that results from Agile-Stage-Gate method. The manager of the Agile-Stage-Gate portfolio must have the flexibility to start or stop projects and reallocate resources on the fly, as the portfolio evolves. Traditional and annual project budgets do not allow for this kind of change.

The Strategic Buckets method provides the needed flexibility. In a Strategic Buckets system, the senior management team establishes buckets of resources for different types of projects each year; the buckets reflect the firm’s strategic goals (Figure 5). Buckets may be defined by project type, technology type, product line, geography or market sector, or any meaningful division. Development projects are categorized by bucket, and then projects within each bucket are ranked, from top priority to low. For example, one bucket would include Agile projects, which may be ranked by their...
Productivity Index (using ECV as the project’s value). Smaller, lower-risk projects in another bucket are usually ranked by some simpler metric, such as benefit/cost. Once ranked, projects are then funded from the top of the list within a bucket until the resources in that bucket are exhausted. Projects lower in the list may be killed or put on hold until resources become available. Once a resource bucket is defined, portfolio managers are free to move resources around within the bucket based on go/kill decisions and prioritizations, as long as the total bucket allocation is not exceeded. This approach facilitates the ongoing go/kill decision making inherent in a dynamic Agile portfolio.

Over time, the Strategic Bucket method ensures that projects and spending breakdowns truly mirror the strategic priorities of the business. Further, resources can be reserved for bolder projects, by creating a separate Agile bucket so that riskier and more ambiguous projects do not compete for resources with smaller, more predictable ones. And the method allows flexibility in making the real-time go/kill decisions needed to take full advantage of the Agile–Stage-Gate method.

Management Implications

Agile–Stage-Gate creates considerable benefits for new-product portfolio management, as well as new challenges. The hybrid process generates and validates project data in real time, producing more, and more reliable data to drive project selection and prioritization decisions. And constant management engagement with the project ensures more timely kill decisions, thus alleviating pipeline gridlock.

New challenges are significant, however; many arise from the conflict between the dynamic portfolio an Agile approach creates and the predictability on which traditional metrics and budgeting approaches rely. Companies implementing an Agile–Stage-Gate approach can implement a number of solutions to address these challenges.

Project progress, which cannot reliably be tracked by Gantt charts, timelines, or other linear tools, can be monitored using the Agile burndown chart. For this purpose, the chart should cover the entire stage, rather than a single sprint. A multi-sprint chart can provide a good sense for how close the project is to its target for the stage at any point in time—the proximity metric—even as the total number of sprints evolves.

The PI and ECV can help measure the value of a project as it proceeds. The PI tracks the value added to a project for each additional unit of spending or work effort expended on it. Plotting the PI as the project moves forward can reveal when projects are heading into trouble; it can also serve as a good index for prioritizing projects—developing a rank-ordered list of projects—for resource allocation. The ECV is a powerful tool for estimating the value of an ill-defined, high-uncertainty project that has multiple iterations. The ECV provides a much more valid estimate of the economic value of such projects than NPV and thus can support both go/kill decisions and prioritization processes when used in conjunction with the Productivity Index.

Finally, Strategic Buckets can be used to ensure that resources are allocated across projects in a way that is consistent with the business’s strategy, even as projects enter and exit the portfolio. Strategic Buckets should be used for Agile projects instead of the more traditional approach of annually budgeting against specific projects, as annual budgets cannot accommodate the dynamic portfolio created by Agile’s many go/kill decision points.

Conclusion

New methods often create new challenges. Agile–Stage-Gate projects are ambiguous, constantly evolving, and iterative, which makes it more difficult to assess the economic value of a project and gauge its progress over time; and dynamic portfolios created by the approach conflict with traditional budgeting methods. Some tools, such as the Productivity Index, Expected Commercial Value, burndown charts, and Strategic Buckets, are available to address these challenges, but more work is needed to test the validity of these methods in practice, and to develop new methods and metrics.
On the positive side, Agile–Stage-Gate has already yielded dramatic increases in speed and R&D productivity for early adopters in the manufacturing sector. Only now, however, is its potential to improve R&D investment decisions and portfolio management becoming clear. By making better information available to decision makers earlier in the innovation process through repeated iterations and validation demos, and by engaging management throughout the process, Agile–Stage-Gate brings us closer to the goal of not only doing projects right, but most important, doing the right projects—getting portfolio management right.

References
IRI’s Networks bring practitioners together to explore member-chosen topics through a combination of peer presentations, interactive round-tables, guest speakers, and workshops. Unlike the more traditional conference format, IRI Networks Conferences are designed to be highly interactive. Instead of going from session to session listening to lecturers, attendees are divided by innovation team function and are encouraged to share pain points, troubleshoot with peers in similar job roles, and gain valuable insights from the IRI community.

ATTEND OUR 2020 CONFERENCES

MARCH 14-16
NEW ORLEANS, LA

SEPTEMBER 15-17
MILWAUKEE, WI

PARTICIPATING NETWORKS
External Technology
Information Services/Technology
Innovation Leaders
Intellectual Asset Management
New Business Development

WWW.IRIWEB.ORG/NETWORKS