

PPM and Agile:

Bimodal Project Management Delivers the Best of Both Worlds

WHITE PAPER: PPM + AGILE

It's no longer a question of 'either/or.' At least it shouldn't be.

Today's PMOs need to be able to combine or easily choose between execution methods to get the right job done right. In this white paper, we'll explore how to successfully employ both agile and traditional methods without sacrificing executive visibility or communicating metrics.

"Intelligence is the ability to adapt to change."

- Stephen Hawking

Table of Contents

- 1. Introduction 3**
 - 1.1 Defining “bimodal” 3
 - 1.2 What is agile? 4
 - 1.3 Agile principles of development: Top four takeaways 5
- 2. The challenge: Integrating PPM and agile processes 6**
 - 2.1 Four common fallacies about agile and PPM 7
- 3. A PPM framework that integrates with agile methods 10**
- 4. Providing executive visibility 12**
 - 4.1 Standardizing cross-project metrics 12
 - 4.2 Calibrating metrics across programs and portfolios 13
 - 4.3 Creating a “single source of truth” for executive reporting 14
- 5. Conclusion 14**

1. Introduction

1.1 Defining “bimodal”

In the “olden days” (namely, the last decade or so), project management offices (PMO) had to choose between execution methods—with more traditional methods, such as waterfall, taking the lead. But today, it’s no longer a debate of “either waterfall or agile.” At least it shouldn’t be.

Instead, PMOs are embracing a bimodal approach to program and project management. “Bimodal IT,” as Gartner defines it, is the practice of implementing more than one execution methodology within a PMO in order to support portfolio, program, and project execution. It’s about being able to choose the right execution model based on the nature of the project at hand. It’s about combining stability and flexibility.

In this white paper, we discuss some of the challenges of integrating agile methods with traditional methodologies, the fallacies that have arisen around doing so, and how to successfully employ both agile and traditional methods without sacrificing executive visibility or communicating metrics, despite varying units of measure.

According to Gartner, Inc., “Bimodal IT is the practice of managing two separate, coherent modes of IT delivery, one focused on stability and the other on agility. Mode 1 is traditional and sequential, emphasizing safety and accuracy. Mode 2 is exploratory and nonlinear, emphasizing agility and speed.”¹

1.2 What is agile?

Unlike more traditional project management methods that focus on a detailed, predetermined plan, agile helps teams adapt quickly to changing realities to deliver a product that is relevant and aligned with the business needs of the stakeholder.

As a methodology, agile has been growing in popularity since the *Manifesto for Agile Software Development*² (aka Agile Manifesto) was published in 2001. The introduction to the manifesto reads as follows:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

The Agile Manifesto continues, outlining twelve principles, including:

- A focus on customer (and/or stakeholder) satisfaction
- Frequent software delivery
- Close cooperation between business people and developers
- Creating an empowered culture with motivated individuals

“Waterfall and iterative approaches are giving ground to much lighter, delivery-focused methods based on the principles of the Agile Manifesto.”

- Forrester Research³

1.3 Agile principles of development: Top four takeaways

The greatest difference between an agile approach and a more traditional project method is the movement away from a highly detailed project plan and timeline. Instead of trying to control change, agile focuses on helping teams react to change by delivering “working software” at iterative stages.

Scrum and Kanban are two of the more popular agile methodologies, however it’s common to find development teams mixing and matching agile practices to find a process that works for them. Whether it’s introducing a daily stand-up meeting, writing story-based requirements, targeting shorter iterations or assigning a product owner, the goal of agile is to create a culture of continuous feedback and a focus on delivering high-quality software that meets stakeholder needs.

1. Self-organized, self-motivated, and empowered teams

Agile encourages a collaborative environment that is transparent and accountable—where individual contributors become true team players. Roles are more fluid and there is a strong emphasis on self-organization as a team. A bottom-up decision-making process is favored—empowering the team to make decisions.

2. User collaboration connects business and IT to guide requirements

Business users are expected to work on a daily basis as part of the agile team. Continuous feedback decreases the chance of a “requirements vacuum” where the development team delivers functionality that does not meet business requirements, and vice versa.

3. A responsive, efficient development process (with less risk)

Agile teams are set up to embrace change and respond to new information, which opposes the traditional project management goals of controlling change and keeping to a plan. The use of a product backlog to prioritize work and deliver software iteratively helps avoid unnecessary work and identifies risks of project failure early in the process.

4. A focus on high-quality, working software

Software is built and tested continuously, often with automated processes, to catch defects early in the development lifecycle. Agile teams favor delivering software “early and often” to ensure requirements can be validated. Working software is valued over comprehensive documentation, which keeps the team focused on the end deliverable rather than work products that are a “means to an end” in the development process.

2. The challenge: Integrating PPM and agile processes

For the most part, project managers have embraced agile practices—attempting to redefine their roles to focus less on planning and controlling projects, and more on providing an environment that leads to success. Today, many project portfolios include a mix of project types and methodologies (agile, waterfall, six-sigma, and “stage-gate”).

However, even with the willingness to adjust, integrating agile into a PPM framework has become the larger challenge. Project managers are faced with different (and often conflicting) methodologies, metrics, and controls.

Some teams are adopting “hybrid” processes that include elements of waterfall and agile methodologies, and are having to rationalize incompatible methods. To solve these challenges, PMOs need an updated project portfolio management (PPM) framework that:

- Incorporates agile practices
- Provides clarity to project teams on how to communicate project health
- Provides predictability and accountability to executive stakeholders

Instead of trying to control change, agile focuses on helping teams react to change by delivering “working software” at iterative stages. Bimodal teams are using traditional methodologies for stability-focused programs, and agile for speed and innovation.

2.1 Four common fallacies about agile and PPM

Agile methods have had a significant influence on best practices for project management. However, radical differences—embracing change, “just-in-time” planning, and eliminating hierarchal decision making—have led to some misconceptions about the compatibility of agile projects with PPM processes.

Below are four common agile fallacies (and why they’re untrue) that have led to concerns about how to monitor and control an agile project as part of a project portfolio.

Fallacy #1: Multiple methodologies can’t exist in the same portfolio

Of course they can. While this may have been true 20 years ago, incorporating multiple methodologies within the same portfolio is absolutely available today—it’s what makes bimodal possible.

Today’s PPM solutions can integrate multiple methodologies to manage the execution of programs and projects within the same portfolio. That way, data is stored within a single source of truth, and teams are able to access the right system to use for the project on which they’re working.

Fallacy #2: Agile projects don’t provide enough executive visibility

The agile method is all about empowering teams to make decisions relative to change, instead of waiting for executive review at every turn. However, that does not mean executive review isn’t needed. It’s not a “free for all.”

Executive visibility is available; it just looks a little differently than in other methodologies. When it comes to executive reporting, the struggle many teams are facing is the requirement to provide rollups in a format that is consistent with traditional methods, not agile. For example, requiring an agile team to maintain a separate task plan for tracking a “percentage complete” metric can negatively impact the benefits of an agile approach (and waste time). Executives and team members have to find reporting methods that communicate status within the parameters of the methodology being used, instead of asking teams to provide metrics that are irrelevant to the process itself.

Fallacy #3: Agile projects don't have reliable "scheduled finish dates"

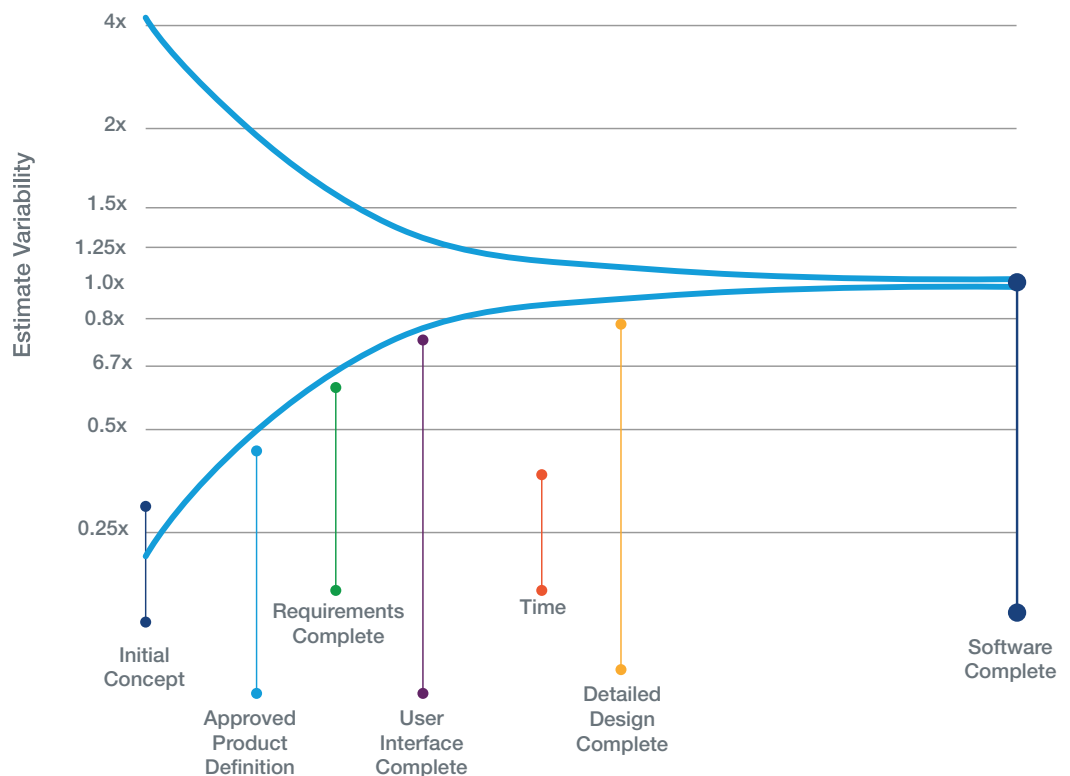
Although agile teams shy away from providing guaranteed delivery dates (given the "cone of uncertainty" (Figure 1) around a project), it's a fallacy that an agile project can't provide a scheduled finish date. If an executive hears that an agile team is only prepared to give estimated delivery dates for the next couple of iterations, it should raise a warning flag about the team's capability to build a credible release plan.

It's true that agile teams use "just-in-time" estimation for iterations and focus on delivering business value incrementally, however release and roadmap planning is used for longer-range estimates. The use of "epics" (large stories) and "themes" (groups of stories) can be used to estimate work that still requires detailed scoping. To be able to estimate with some degree of reliability, an agile team will need to have some degree of normalization around the size of stories and understand their velocity (story points delivered in an iteration).

Figure 1

Definition: Cone of uncertainty

The "cone of uncertainty" refers to when the specific details of the nature of a project (specific requirements, details of the solution, project plan, staffing, and other variables) are unclear. The variability in these factors contributes to project estimates. Over the course of a project, these sources of variability are more easily defined—diminishing the variability itself and transitioning project estimates into more definitive delivery dates.



Fallacy #4: Agile and traditional practices aren't compatible

Some agile practices are fundamentally different than traditional project management practices; for example, the planning and executing process groups in the Project Management Institute's (PMI) Project Management Book of Knowledge (PMBOK). However, the two aren't mutually exclusive.

Multiple practices can be combined to create a powerful project management framework. While agile practices offer a swift, collaborative execution, traditional practices for project intake or initiation processes, as well as for costs, communications, and risk management, are often more mature than in agile. An experienced project manager can add practices from the PMBOK or similar methodologies to support agile projects without disrupting the culture and work practices of an agile team.

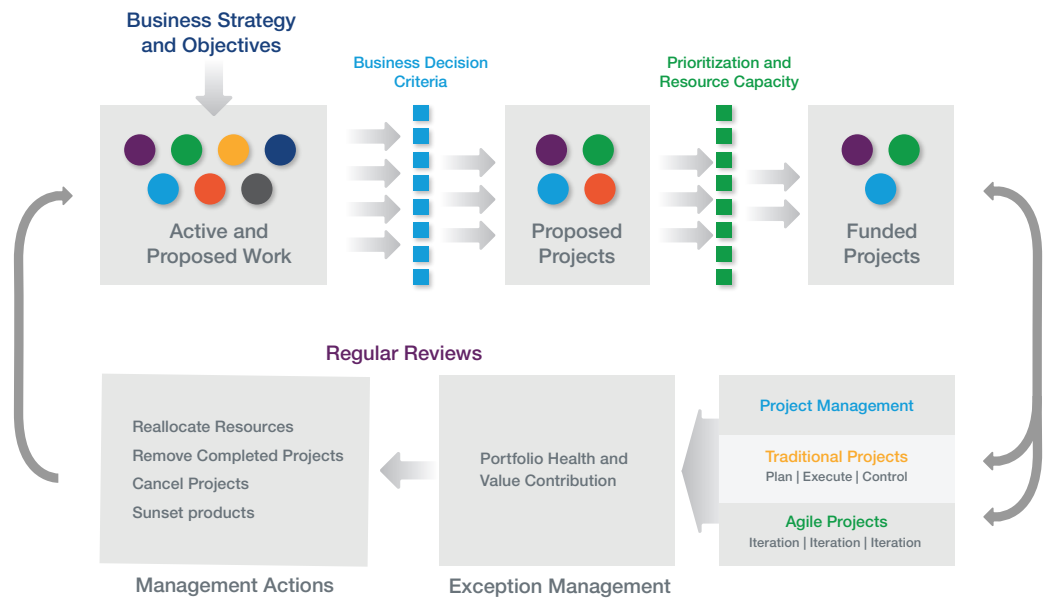
Bimodal mixing and matching

Bimodal extends beyond projects. Portfolios that consist of a mixture of methodologies are also bimodal. For instance, a work package that is to deliver a storage solution will have a hardware component, a firmware component, and a software component. The hardware component lends its development more to a waterfall method—particularly to control the supply chain. The firmware and software could be delivered more efficiently with agile.

3. A PPM framework that integrates with agile methods

For more traditional methods, like waterfall, a PPM framework typically includes processes for project intake and selection, project approval and initiation, and project and portfolio monitoring, along with templates, artifacts, and metrics for the different methodologies used within an organization. An example PPM framework is shown in Figure 2.

Figure 2
A typical PPM framework



Integrating an agile project methodology into a PPM framework is no different than integrating a more traditional project methodology, **with four exceptions:**

1. Don't stifle the process: Agile practices are not for the micromanager. Imposing unnecessary stakeholder reviews, checkpoints, and data capture requirements on an agile project will reduce the effectiveness of the team. Of course, if major decisions need to be made with executive input or the agile project has dependencies on other projects, stakeholder meetings will be necessary but these should be the exception, not the rule.

2. “Story points” track progress: On the surface, this is a fundamentally different way of tracking progress than using a task plan and measuring task-hour completion. Also, in a traditional project, project managers assign tasks to owners, whereas agile teams are self-organizing—making tasks subject to ownership change to ensure delivery is on track. This difference requires that metrics for “project health” and “percentage complete” are tracked slightly differently than projects based on task hours. Instead of gleaning a relative percentage from planned hours vs. tracked hours, agile project managers can take the percentage of planned story points against remaining story points.

3. Project resources are dynamic: Because task assignments are dynamic, it’s better to avoid giving part-time assignments to an agile team. Instead, treat agile teams as a unit and assign resources to them full-time (with the exception of resources such as technical architects and DBAs, which are typically spread over multiple projects).

4. Reviews are based on working software: Regular reviews of agile projects are more focused on “working software” than reaching pre-determined milestones or delivering project documentation. Reviews should be tailored to the type of project to ensure they are valuable to the team.

4. Providing executive visibility

Regardless of the project methodology, project status still has to be communicated up the ladder. Combining multiple methods does affect how metrics are considered. However, there are ways to communicate project completeness without having to fit metrics into a box.

4.1 Standardizing cross-project metrics

Developing a PPM framework with standard metrics that apply to both agile and traditional projects is important. The five common metrics that enable executives to get visibility into project status, regardless of the delivery method, are outlined below:

1. Scheduled finish date: “Planned finish date” is the estimate made at the inception of a project for the planned delivery date, whereas “scheduled finish date” uses current data to estimate the finish date at any given point in time. For a traditional project, this is based on the task plan and critical path for the project. For an agile project, it is based on the release plan. Because a cone of uncertainty (see Figure 1 on page 8) exists regardless of project methodology, the accuracy of scheduled finish dates should be comparable for both traditional and agile projects. Comparing scheduled finish date to planned finish date will give an indication of the team’s ability to estimate delivery dates accurately.

2. Percentage complete: “Percentage complete” provides an indication of project progress. For traditional projects, it is calculated by summing the hours for completed tasks and dividing the total task hours for a project. For an agile project, progress is measured by story points delivered. Percentage complete is calculated by story points accepted divided by total story points for a project.

The calculations are the same, the units of measure are not

For example, let’s look at “percentage complete”

Traditional method

x = Tracked hours

y = Planned hours

(x / y) = % complete

Agile method

a = Completed story points

b = Total story points

(a / b) = % complete

3. Scope changes: Percentage complete gives an indication of progress, but does not show if the scope is changing on a project. For traditional projects, this is typically represented by the number of change requests. For an agile project, it is typically measured by the change in total story points over time.

4. Actual cost vs. budget: Both traditional and agile projects have capital and operational expenses that are tracked. Actual cost vs. budgeted cost should be reported, regardless of the project methodology.

5. Project health: Project health is a summary metric that indicates if a project is on track, needs attention, or is in trouble. This metric varies by organization and is calculated using conditional logic based on the four metrics above, as well as other data, such as outstanding issues. One of the simplest ways to calculate project health is to compare actual percentage complete with the expected percentage complete based on the start date and scheduled finish date (assuming the velocity of work delivery across the project timeframe is uniform). A project health status of “needs attention” or “in trouble” is often triggered when projects are not delivering business value, there are significant scope changes, costs exceed budget, or there are major unresolved issues on the project.

Providing these metrics in a dashboard format using a PPM system enables executives to quickly identify projects that require focus or intervention, regardless of the project management methodology employed for its execution.

4.2 Calibrating metrics across programs and portfolios

The above cross-project metrics are a good way to provide a common scorecard for project execution. One challenge to consider, however, is calibration across projects, programs, and portfolios. For projects based on task hours, calibration is not an issue because the unit of progress is the same (hours). However, for agile projects, the unit of progress is typically story points, and the definition of a story point may vary from project to project based on how an agile team estimates work.

A common calibration technique is to bring agile teams together on a quarterly basis and normalize point size by picking sample stories of different sizes and comparing effort. This can be done by comparing “ideal days” for a given story and calibrating across different teams to provide guidance on story-point size. If a “scrum of scrums” exists, this technique will mostly resolve calibration issues at a program level.

When it comes to programs that combine multiple methodologies, it may look like comparing apples to oranges. However, as previously shown, forecasting to 100% is available regardless of the execution method. Instead of looking for ways to uniformly combine methodologies across a program, uniquely tracking methodologies within the same work package provides a more accurate view of how each methodology performed.

4.3 Creating a “single source of truth” for executive reporting

A PPM system is the best way to provide a “single source of truth” for executive reporting and decision making on a project portfolio, providing **three major benefits:**

- 1. Complete transparency:** A PPM system provides a clear view over agile and traditional project execution, and helps eliminate the issue of a team “sugar coating” project status if things aren’t going well
- 2. Consistent project reporting:** A common reporting framework within the tool standardizes how reports are generated and communicated
- 3. Integration improves efficiency:** Integration with different methodology tools eliminates multiple (and manual) data entry into the PPM system

5. Conclusion

For mature bimodal teams, integrating multiple methodology tools provides a way to use specialized agile project and traditional project management tools, while still enabling a single source of truth for project portfolio reporting.

In terms of bimodal principles, integrating agile and traditional practices provides your teams with varying methods that can be employed on a project-by-project basis. Bimodal advocates for business-sustaining projects to be executed using a traditional method, and innovation-oriented projects using agile.

Although some fallacies still linger regarding agile methods, they should not deter PMOs and executives from adopting agile methodologies where appropriate.

Carefully consider which projects are suitable for agile, and develop a PPM framework that integrates agile and traditional project methodologies. By finding what works for your organization, you’ll achieve better executive visibility and execution processes.

About Changepoint

Changepoint delivers market-leading solutions in Business Execution Management™ (BEM) to companies around the world. Our solution suite is comprised of Project Portfolio Management (PPM), Enterprise Portfolio Management (EPM), Professional Services Automation (PSA), and more. Today, thousands of organizations—large and small—rely on Changepoint to get the job done. With Changepoint, smarter decisions are easier to make and flexibly adapting to changes is easier to do. The result? A shorter, clearer road to innovation and customer satisfaction. Visit: www.changepoint.com.



© 2016 Changepoint

02.03.16

¹ The Gartner IT Glossary: <http://www.gartner.com/it-glossary/bimodal> ² Visit agilemanifesto.org to read the original Agile Manifesto. ³ Agile Development: Mainstream Adoption Has Changed Agility, Forrester Research, 2010