Whitepaper

Agile Project Management: What’s the story?

There is no doubt that Agile project management, the adaptive methodology that spread like wildfire around the turn of the millennium, is here to stay.

As organizations make the shift to Agile methods, there remains confusion, skepticism, and rampant misconceptions about Agile. And like any methodology, Agile requires organizations to look at the triple foundation of people, process, and technology. To underplay any of those is to sabotage success.

This white paper will explore Agile from all three perspectives, and will offer guidelines for effective Agile project management. It will expose common myths, offer tips on the cultural shift to Agile, and address concerns about mixed environments where some departments use Agile and others do not. Finally, it will show how Agile is implemented in Planview Enterprise® – with and without adjacent tools such as Atlassian’s CA Agile Central, Version One, and IBM Rational® Team Concert (RTC).

Topics we'll address are:

- What's different about Agile? What are its benefits?
- How does Agile work? What do all those terms mean?
- What roles are required for effective Agile projects? And specifically, how does the role of the project manager change?
- How do we address typical management concerns and skepticism about Agile? Are some of those concerns justified?
- Can Agile work for Virtual Teams?
- How is Agile implemented in Planview Enterprise? What if some departments don't use Agile?
- If we plan to use Agile tools, how do we effectively use them with Planview Enterprise? Which tool is best for what activities?

Whether you're new to Agile or an Agile expert, in addressing these common questions, this white paper will position you for success and show you how to effectively use Planview Enterprise for Agile project management. The Agile movement is undoubtedly expanding, and those who understand its nuances – around people, process, and technology – will be more effective.

Introduction: What's Different about Agile?

Traditional "Waterfall" style project management methods involve detailed up-front planning and design (often known as BDUF or Big Design Up Front). The features and scope are predetermined, and a schedule and associated costs are estimated. This works fine for construction projects, where details must by nature be predefined.

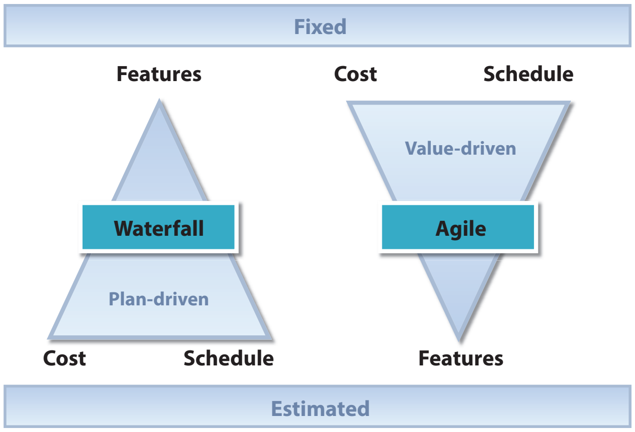
For knowledge work, however, such as software development, research projects, most engineering projects, and other types of work that involve a level of uncertainty and high change rates, traditional methods fall short. In an environment where change is the norm and value can be improved based on information from new or unexpected sources, there is a more efficient and realistic way to manage. This is where Agile comes in.

Although Agile-like methods were employed in Japan as early as 1986, it wasn't until 2001 that a group of thought leaders in software development formalized the approach. This culminated in the Agile Manifesto, which cited 12 principles based on four overarching values that emphasized:

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

Agile focuses on people working closely together in short iterations (generally two to four weeks), delivering some form of measurable value with each iteration and inspecting and adapting as appropriate. And throughout all this, the focus is on the product and the customer.

Another difference about Agile is that instead of having a fixed scope and then estimating the schedule and cost, it involves working in fixed time and cost iterations, and estimating what features can be delivered in those iterations. In other words, it turns traditional project management on its head (see figure below).



Collectively, this collaborative and iterative approach results in:

- Faster value delivered to the user . . . because of the iterative process

- Reduced risk . . . because of iterative product delivery
- Reduced uncertainty . . . because of the product and customer focus
- Better decision making . . . because of the collaborative focus
- Increased trust . . . because of incremental value being delivered
- Fewer errors. . . because of greater communication
- Clearer accountability… because of delineated roles

With such benefits, it is not surprising that Agile has enjoyed such rapid growth in knowledge worker environments, extending beyond just software development where the idea originated.

How Does Agile Work?

Agile is a whole different way of thinking, so, not surprisingly, the terminology is different as well. Below is a summary of how Agile works, along with the relevant terminology:

Features are defined in *stories*, which identify the user(s), action(s), and benefits for that feature. (As an analogy, think of stories in fiction: they involve character, plot, and motive)

User stories are estimated in relative *points* (though some organizations use *ideal* days). Story points are a measure of relative complexity and effort. As work is completed, the team can "earn" points for the completed stories.

User stories are prioritized so that the highest value features or enabling features are delivered as soon as possible.

The team works collaboratively on prioritized stories from a *backlog* in fixed-time iterations, generally 1 to 4 weeks in duration (in the Agile Scrum methodology, popular in software development, these iterations are called *sprints*).

The team meets regularly to share information and progress. In the Agile Scrum methodology, these are daily 15-minute standup meetings. The key is to stay out of the weeds. These are not problem-solving meetings.

After each iteration/sprint, the team and stakeholders meet to assess progress, make adjustments, and plan next steps (i.e. evolutionary planning). This is often referred to as a *retrospective*. It differs from the traditional post-implementation review, which is usually held long after the project and far too late to make a difference on the current project.

Progress is tracked via *burndown charts*, which track work remaining over time (in terms of story points planned vs. earned).

Velocity measures story points completed per iteration/sprint (i.e. the speed at which value is being delivered).

A release of a product is generally made up of multiple iterations/sprints. Often, there is a *planning sprint* at the start of each release to plan out the stories for that release.

Agile Roles

With Agile projects, the role of the project manager must be reconsidered. This is because:

Baselines are irrelevant (A baseline is meaningless if the whole concept of Agile is to plan and adapt the features to fit within fixed time and cost iterations.)

Formal up-front requirements don't apply (Customer needs must be understood, but detailed requirements are not all fixed up front; requirements evolve with learnings from each sprint.)

The devil is in the details (Agile projects succeed or fail based on mutual understanding of customer needs and technical capabilities. Thus leaders of Agile projects need to understand the details.)

It's all about the product (Unlike traditional projects, where the focus is on managing the scope, schedule, and budget, Agile projects are all about the product as opposed to the project. Items like schedule and budget are fixed.)

Agile projects are community-driven (In Agile projects, developers, analysis, product owners, and customers are in constant communication. If they're not, then the Agile methodology is not being followed. Unlike traditional projects, the project manager is not the primary source of communication.)

Agile projects are relatively low risk (With a large, complex project with lots of cross-team involvement and moving parts, Agile may not be the best approach. Agile is often used in software development and other knowledge-work projects that can bear some sort of tolerance for uncertainty and a single team can work at a rapid pace.)

With all this in mind, some may ask the inevitable question: What, then, is there for a project manager to do? It's a valid question, and organizations have taken different approaches, from not having a project manager at all to using the project manager as a facilitator across the product owner, customers, and developers. In some organizations, the Scrum Master serves as the project manager.

Below are typical roles in an Agile project:

- **Product Manager/Owner** – determines the product vision and ensures that features listed in the backlog are prioritized and understood; provides User Stories (users/actions/benefits)

- **Customer** – monitors progress and provides input for valuable deliverables (Note: The Product Manager typically serves as the proxy for the customer when it comes to electronically recording comments and input)

- **Development Manager/SCRUM Master/Project Manager** – populates sprints from the project backlog and updates story points based on planning sprints; facilitates daily meetings and sprint demos

- **Developers** – are assigned to stories and make testing notes against stories; report effort for cost tracking purposes

- **Resource Managers** – optimize resource utilization across multiple projects and ensure resource availability on critical projects

- **QA/QE Manager/Testers** – focuses on quality assurance/engineering, including process improvements, testing, and measurement

Fear of Agile: Addressing the Concerns

Since the beginning of the Agile movement, defenders of traditional methods have been skeptical. Some of their concerns are valid, while others are rooted in ignorance of true Agile methods. Ten common management fears are:

1. It won't work for big, complex projects.

2. It's too open-ended. We can't predict costs. It's "sanctioned scope creep."

3. It sounds like "back of the napkin" design and planning.

4. It's too "techie" focused.

5. Software developers don't talk the same language as customers.

6. Customers don't have time to get involved in planning.

7. We don't want our customers to see our dirty laundry.

8. This "teamwork" approach doesn't sound practical.

9. Daily meetings? Our employees will feel like they're under a microscope.

10. It's too rigid and inhibits individual creativity.

The truth is that Agile involves more planning than they realize – with each iteration in fact. And customer and business involvement is higher and communication is elevated, so the focus isn't strictly on technology. But other concerns are quite legitimate. Overall, the following ten strategies can address typical concerns:

1. Make the strategy fit the situation (use the right methodology for the right job; Agile is best for a cohesive team working on knowledge work with a level of uncertainty.)

2. Reduce risk by focusing on business symptoms over solutions (don't lose sight of the business problem being solved.)

3. Adopt a "see for yourself" policy to assess the user experience (before and after.)

4. Foster a systems-thinking approach (think beyond the software; think also of processes, upstream and downstream systems; stakeholder impact, and so on.)

5. Engage a business analyst to ensure that details aren't overlooked (developers often lack an understanding of business details.)

6. Bridge the culture gap between technicians and customers (this can be done through training for developers on customer interaction or by engaging others to interpret and translate to bridge the gap between developers and customers.)

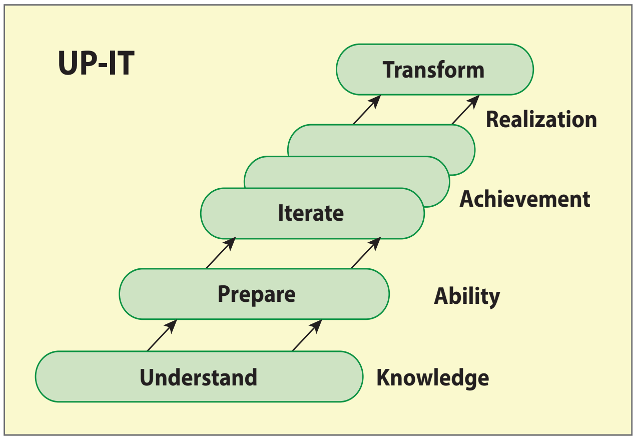
7. Embrace change, but manage it (assume features will change based on learnings, but have discussions about overall scope, and manage and log changes to the scope.)

8. Focus on product evolution, not project evolution; manage by prioritized features and release; have release targets (most sprints do not result in a deliverable to the customer, but provide some level of value; releases do result in customer deliverables; also, be sure to prioritize features and set milestones tied to releases.)

9. Gain management commitment to attend retrospectives and open demos (without management and customer involvement, the Agile methodology loses its power; gaining commitment is key.)

10. Focus on outcomes and value, not activities or hours (don't make people feel like they're under a microscope - focus on what is needed, not how to go about it; allow flexibility and individual creativity, and focus only on agreed-upon outcomes and value for each sprint - freedom with hours and creativity can balance out the heavy focus on achievement, and agreement on what's achievable can reduce pressure.)

One additional way to reduce other shortcomings of Agile is to take an integrated approach to emphasize that which is generally outside of the Agile methodology. To address this, use a four-phased approach that can sit on top of your current methodology. The acronym spells UP-IT:



- Understand (build the knowledge necessary to be successful on the project; do research; examine the current and desired user experience.)

- Prepare (gain the capabilities to execute the project successfully, such as training, the right tools, gaining stakeholder commitment, etc.)

- Iterate (plan and execute iterations/sprints to deliver value through planned releases.)

- Transform (focus on ongoing support and self-sustainability for the customer; be sure to educate the team on lessons learned to be applied to future releases.)

Lastly, there are some situations where Agile is probably NOT the best approach. Here are eight examples of where to avoid Agile methods:

1. Large enterprise initiatives that need requirements defined up front, and generally require heavy documentation

2. Where formal specs are needed for auditability, safety, or preciseness

3. Where scope and requirements are known, can be defined, and are unlikely to change much

4. Where lots of approvals are needed by multiple parties

5. Where the organizational or team culture is incompatible with an Agile approach

6. If customers/users are not generally available to participate

7. If the team lacks interpersonal skills or heavy technical knowledge (i.e. they can't be empowered)

8. If the team is too large to be effective at cross-communication (i.e. over 100 people, though this number can vary depending on the culture and technology involved)

Virtual Teams and Agile

According to Gartner, by 2015, three quarters of knowledge-based project work in the Global 2000 will be completed by distributed virtual teams. This presents new challenges in collaboration, which is a central tenet of Agile. With this in mind, an understanding of new media and online collaboration tools will become more and more important.

Agile teams will need to employ methods to stay in regular communication, whether by teleconference, webinars, Twitter-like tools such as Yammer, collaboration websites, or other media. Some organizations have even taken to the use of Wikis to capture, organize, and share knowledge.

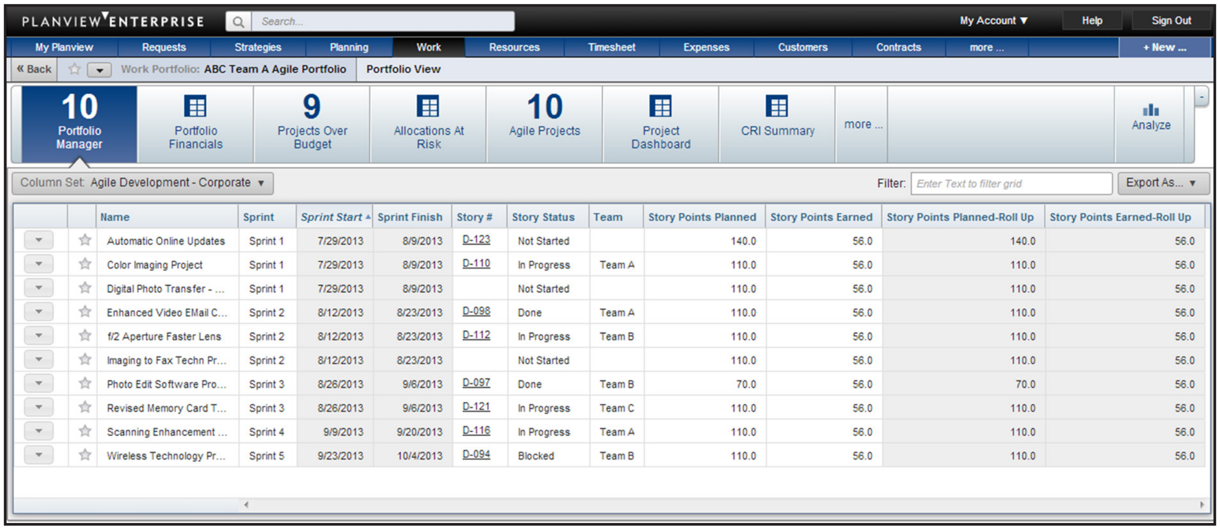
This also applies to organizations that are currently unable or unwilling to co-locate their development teams. Most Agile experts agree that co-location is best, but with the right technology and the right principles, distributed teams can still be effective.

Planview Enterprise and Agile

Planview Enterprise has features to support organizations that use the Agile methodology. These include:

- Agile column sets when examining project schedules in the Work Manager. Built-in fields include:
  - Sprint (the associated Sprint name)
  - Story # - unique identifier code (can optionally link to a URL and/or other software)
  - Story Status (e.g., Done, In Progress, Not Started, etc.)
  - Sprint Start Date / Finish Date
  - Story Points Planned (manually recorded)
  - Story Points Planned Roll-Up (calculated automatically)
  - Story Points Earned (manually recorded)
  - Story Points Earned Roll-Up (calculated automatically)
- To-Do List and Notes functionality for story details
- Content Management for documenting and storing information from sprint retrospectives
- A Community page for discussions and team updates
- Resource authorizations for tracking time and cost
- Resource Reserves for high level resource demand planning
- Defined user roles for the various participants in the Agile methodology
- Custom reporting dashboards and Insight Analytics OLAP reporting tool, offering iteration velocity comparisons, weekly burndown charts, and end-of-iteration reports

Below is a typical Agile column set for an Agile project:



Column Set: Agile Development - Corporate ▾

Filter: Enter Text to filter grid

Name	Sprint	Sprint Start	Sprint Finish	Story #	Story Status	Team	Story Points Planned	Story Points Earned	Story Points Planned-Roll Up	Story Points Earned-Roll Up
Automatic Online Updates	Sprint 1	7/29/2013	8/9/2013	D-123	Not Started		140.0	56.0	140.0	56.0
Color Imaging Project	Sprint 1	7/29/2013	8/9/2013	D-110	In Progress	Team A	110.0	56.0	110.0	56.0
Digital Photo Transfer - ...	Sprint 1	7/29/2013	8/9/2013		Not Started		110.0	56.0	110.0	56.0
Enhanced Video EMail C...	Sprint 2	8/12/2013	8/23/2013	D-098	Done	Team A	110.0	56.0	110.0	56.0
f/2 Aperture Faster Lens	Sprint 2	8/12/2013	8/23/2013	D-112	In Progress	Team B	110.0	56.0	110.0	56.0
Imaging to Fax Techn Pr...	Sprint 2	8/12/2013	8/23/2013		Not Started		110.0	56.0	110.0	56.0
Photo Edit Software Pro...	Sprint 3	8/26/2013	9/6/2013	D-097	Done	Team B	70.0	56.0	70.0	56.0
Revised Memory Card T...	Sprint 3	8/26/2013	9/6/2013	D-121	In Progress	Team C	110.0	56.0	110.0	56.0
Scanning Enhancement ...	Sprint 4	9/9/2013	9/20/2013	D-116	In Progress	Team A	110.0	56.0	110.0	56.0
Wireless Technology Pr...	Sprint 5	9/23/2013	10/4/2013	D-094	Blocked	Team B	110.0	56.0	110.0	56.0

The typical Agile process using Planview Enterprise is as follows:

- Managers and developers populate sprints with user stories from the project backlog with a simple cut/paste function.
    - Note: If stories are finished early, the remaining time in the sprint can be spent on other stories from the backlog.
- Developers may assign themselves as resources to stories, make testing notes directly against the story for viewing and editing by all team members; they can also track actual effort expended against a story via Planview's timesheet for project costing purposes.
    - Note: Assignments can be done in a number of different ways. Allocations are NOT the best approach because effort is not planned and schedules should not "slip." The reserve/authorization combo is best. Reserves (soft booking named resources) and Authorizations (allowing named resources to enter time) can be at a project or phase level.
- Iterations or sprints may be planned for whatever duration is determined by each project team. And stories can include those from developers, testers and other team members to ensure that working software or tangible value can be delivered on a frequent basis.
    - Note: Some companies, such as Planview, have all projects on the same sprint schedule. Others have a separate schedule per project.
- There is no need for scheduled start, scheduled finish, duration, or planning of effort. Sprint dates are fixed, and progress is tracked via planned and earned story points.
    - Note: This frequently requires a cultural adjustment, so plan accordingly.

Using Planview Enterprise with Other Agile Tools

Many Agile developers use products such as Atlassian's JIRA, CA Agile Central, Version One, and IBM Rational Team Concert. These products and others, have full-featured Agile capabilities, each with their own emphasis and strengths.

Planview Enterprise and Agile tools all offer story point entry and tracking, velocity and burndown charts, and collaboration. This may lead to the question: What's the benefit of using Planview Enterprise with these other Agile products?

The benefits of using Planview Enterprise in the mix with other Agile solutions are as follows:

- To align program and portfolio management with development work
- To better integrate operational plans to Agile execution
- To enable resource managers to balance demand with capacity
- To share project status seamlessly across management and development teams
- To enable project managers and developers to communicate in their own languages – everyone gets to work within their own preferred tool
- To make high-level approval, timing, and resource allocation decisions based on how you plan capacity and demand, be it FTEs, money, user story points, ideal developer days, and/or other measures
- To improve efficiency, productivity, and collaboration

The integration with an Agile tool provides a different workflow:

- In Planview Enterprise, a product or project manager inputs project phases, iterations, and high level tasks
- The data is sent to the Agile tool
- In the Agile tool, developers assign and implement the work to deliver features
- Story Point and Status updates go to Planview Enterprise for the project manager
- A story identifier in Planview Enterprise links to the story in the Agile tool
- Resource Authorizations and Reserves are entered in Planview Enterprise at the project or phase level, and time tracking (if desired) is done in Planview Enterprise

Some organizations only use Planview Enterprise for resource authorizations and reserves, doing all their tracking and reporting in their Agile tool. Others allow stories to be entered in the Agile tool or Planview Enterprise, depending on the development team's preference, with everything being electronically replicated in Planview Enterprise and the other software via a batch process.

To summarize:

Planview Enterprise has Agile support, including Agile column sets that support story creation and story point tracking, as well as team-based capacity planning, content management, collaboration, and burndown and velocity charts.

Planview Enterprise software can also be used in conjunction with leading Agile software, such as JIRA, CA Agile Central, Version One, and IBM Rational Team Concert.

When using Planview solutions with an Agile tool, multiple options exist, and the approach depends on organizational preference and overall use of Planview Enterprise.

About Planview Enterprise

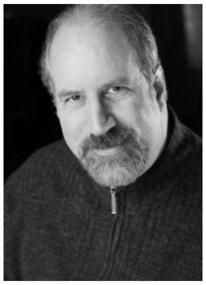
Today's business leaders are under pressure to execute on strategy, given the reality of constrained people and financial resources, in an increasingly competitive marketplace.

Planview Enterprise is a portfolio and resource management solution that connects strategy to execution by improving decision making across the enterprise, including product development, IT, and services. By integrating planning and execution, Planview Enterprise enables organizations to prioritize their portfolios, balance organizational capacity against demand, link plans and resources to project execution, and manage the underlying financials of the entire process. Planview's focus on portfolio and resource management drives a unique level of innovation and commitment to customer success.

Planview Enterprise, a market leading solution for end-to-end portfolio management, from ideation through operational planning, execution, and value delivery.

Learn more about Planview Enterprise for Agile in the Planview Enterprise demo.

About the Author



Jerry Manas is an internationally best-selling author, speaker, and consultant. He is frequently cited by leading voices in the world of business, including legendary management guru Tom Peters ("In Search of Excellence"), who often references Manas's bestselling book *Napoleon on Project Management* for its insights on simplicity and character, and Pat Williams, Senior VP of the Orlando Magic, who called Manas's book *Managing the Gray Areas* "a new path for leaders."

Jerry's latest book is *The Resource Management and Capacity Planning Handbook* (McGraw-Hill), which Judith E. Glaser, noted author of *Conversational Intelligence*, touted as "the first book dedicated to what is essentially the drivetrain of organizations—the effective use of its people toward its most important activities." He also co-authored the bestseller, *42 Rules for Creating WE* with Glaser, which Angela Ahrendts, CEO of Burberry, called "today's greatest guide for team success."

A sought-after consultant, Jerry helps clients maximize their organizational people resources, leading to a greater capacity to innovate, a more value-focused and productive workforce, and an increased ability to adapt to change. He is a popular speaker at events around the world, speaking on resource planning, organizational change, lessons from history, and other topics.

In addition to his writing, consulting, and keynotes, Jerry is a founding member of The Creating We Institute, an international thought leadership community dedicated to fostering WE-centric thinking in organizations through research in neuroscience and positive psychology. He. was also part of the leadership team for several Project Management Institute standards.

Jerry's work has been highlighted in a variety of publications, including the Houston Chronicle, Chicago Sun Times, National Post, Globe and Mail, Huffington Post, and others. He has appeared on radio and TV internationally, including Iceland, where he appeared on a National TV program talking about their economic recovery.

Follow him on Twitter **@jerrymanas**

PLANVIEW®