

# The Need for Agile Project Management

By Mike Cohn, [mike@mountaingoatsoftware.com](mailto:mike@mountaingoatsoftware.com)

And Ken Schwaber, [ken.schwaber@verizon.net](mailto:ken.schwaber@verizon.net)

One of the common misperceptions about agile processes is that there is no need for project management, and that agile projects run themselves. It is easy to see how an agile process' use of self-organizing teams, its rapid pace, and the decreased emphasis on detailed plans lead to this perception. In a recent egroup, a project manager at a company that was implementing agile had been moved to another area because, "...agile doesn't require management." However, agile processes still require project management. In this article we will look at the reasons why and the types of management.

## ***The Need for Project Management***

Project management is critical to the success of most projects, even projects following agile processes. Without management, project teams may pursue the wrong project, may not include the right mix of personalities or skills, may be impeded by organizational dysfunctionality, or may not deliver as much value as possible. We are beginning to formalize these management responsibilities. In the process, we hope that they become clearer to those beginning to implement agile processes, without those people losing the feeling for people, collaboration, and interactions that underlie all agile processes. The below table summarizes these management responsibilities.

| <b>Activity</b>                  | <b>Owner</b>  | <b>Responsibilities</b>  |
|----------------------------------|---------------|--|
| Manage the vision                | Product Owner | The Product Owner establishes, nurtures and communicates the product vision. The Product Owner achieves initial and on-going funding for the project by creating initial release plans and the initial Product Backlog.  |
| Manage the ROI                   | Product Owner | The Product Owner monitors the project against its ROI goals and an investment vision. The Product Backlog is updated and prioritized to ensure that the most valuable functionality is produced first and built upon. The Product Owner prioritizes and refines the Product Backlog and measures success against expenses.  |
| Manage the development iteration | Team          | During an iteration the team selects and develops the highest-priority features on the Product Backlog. Collectively, the team expands Product Backlog items into more explicit tasks on a Sprint Backlog and then manages its own work and self-organizes around how it desires to complete the iteration. The team manages itself to its commitments.  |
| Manage the process               | Scrum Master  | The Scrum Master is responsible for setting the team up for success by ensuring the project and organizational culture are optimized for meeting the ROI goals of the project. This involves organizing a Sprint Planning Meeting (during which the team expands Product Backlog into Sprint Backlog), a Sprint Review Meeting (during which the newly developed functionality is demonstrated), shielding the team from outside disturbances, holding brief Daily Scrum meetings, and removing obstacles to progress. |
| Manage the release               | Product Owner | The Product Owner makes decisions about when to create an official release. For a variety of reasons it may not be desirable to release at the conclusion of every increment. Similarly, if an official release is planned for after the fifth increment it may be released (with fewer features) after the fourth increment in order to respond to competitive moves or capture   |

|  |  |  |
|--|--|--|
|  |  | early market share. The Product Owner makes these decisions in a manner consistent with the investment vision that has been established for the project. |
|--|--|--|

## Shared Vision

For a team to succeed with agile development it is essential that a shared vision be established. The vision must be shared not just among developers on the development team but also with others within the company. Most plan-driven processes also advocate the need for a shared vision; however, if that vision isn't communicated or is imprecise or changing, the project can always fall back on its detailed (but not necessarily accurate) lists of tasks and procedures. This is not the case on an agile project and agile project participants use the shared vision to guide their day-to-day work much more actively.

The formation of the project vision is not the responsibility of the agile project manager; usually the vision comes directly from a customer or customer proxy, such as a product manager. The project manager, however, is usually involved in distilling the customer's grand vision into a meaningful plan for everyone involved in the project. Rather than a detailed command-and-control plan based on Gantt charts, however, the agile plan's purpose is to lay out an investment vision against which management can assess and frequently adjust its investments, lay out a common set of understandings from which emergence, adaptation and collaboration occur, and establish expectations against which progress will be measured. The project manager works with the customer to layout a common set of understandings from which emergence, adaptation and collaboration can occur. The agile project nurtures project team members to implement the vision

## Obstacles

Imagine yourself on a bicycle pedaling along a road some warm summer day. After the ride is done you may look back on it as having been relatively free of surprises; however, while riding you probably had to avoid a number of obstacles: the glass in the road, the narrow shoulder, the car turning left in front of you, and so on. Most of these obstacles are easily avoided by a little forethought and by looking ahead on the road. You don't need to plan the entire trip in advance ("At 9:43 a.m. remember to avoid the pothole on 95<sup>th</sup> Street") but paying attention and glancing down the road help you avoid obstacles or even head-on collisions. Participating in an agile project is much the same way: a little forethought and looking ahead go a long way toward improving the journey.

Most agile processes prescribe a highly focused effort on creating a small set of features during an "iteration" or "sprint" after which the team quickly regroup and decides on the set of features for the next iteration or sprint. While an iteration is ongoing the team members are expected to focus exclusively on the current iteration.

While this sharp focus leads to greater productivity during the current iteration it can cause a bit of a billiard-ball effect as the conclusion of one iteration can bounce out the start of the next. A project manager who spends a small amount of time looking forward at the next iteration is an excellent buffer against this effect. For example, many organizations have travel restrictions that require plane tickets to be purchased two weeks

in advance. If a team could benefit from having a remotely located employee on site during the coming iteration the time to plan for that is during the current iteration.

Another type of obstacle may be a team member.

While agile processes such as Extreme Programming and Scrum rely on self-organizing teams, an agile project manager cannot simply turn a team loose on a project. The agile manager must still monitor that corporate policies or project rules are followed. Participation on an agile team does not turn all developers into model employees. In most cases the team itself will employ some form of sanctioning on an employee who is not working hard enough or is exhibiting other performance or behavior problems. However, in the most severe cases the collective team usually cannot be the one to terminate or officially reprimand an employee. Performance feedback can always be expressed in terms of team views of the individual's contribution to the team (e.g., "I've discussed it with members on the team and we do not think you are being careful enough in your unit testing"). But if a counseling or coaching session is necessary it is usually best when between just the project manager and the team member.

## **Organizational Dysfunctionality**

Many companies have at least one dysfunctional area. This may be the "furniture police" who won't let programmers rearrange furniture to facilitate pair programming. Or it may be a purchasing group that takes six weeks to process a standard software order. In any event these types of insanity get in the way of successful projects. One way to view the project manager is as the bulldozer responsible for quickly removing these problems.

The Scrum process includes a daily meeting during which all team members are asked three questions. One of these questions is, "What is in the way of you doing your work?" The agile project manager takes it on himself to eliminate these impediments. Ideally, he or she becomes so adept at this that impediments are always removed within 24 hours (that is, before the next daily meeting).

Participate in enough agile projects and you begin to hear the same impediments brought up time after time. For example:

- My \_\_\_\_ broke and I need a new one today.
- The software I ordered still hasn't arrived.
- I can't get the \_\_\_\_ group to give me any time and I need to meet with them.
- The department VP has asked me to work on something else "for a day or two"

The project manager is responsible for optimizing team productivity; this means it's his responsibility to do whatever possible to minimize obstacles. Most organizations will not want every developer calling the ordering department to follow up on delivery dates for software. Similarly, a project manager who can know when to push the IT manager for quick setup of a new PC and when not to push (perhaps saving a favor for later) will be more effective than every programmer calling that same IT manager.

## Politics

Politics are at play in almost every organization. Most organizations have only limited funds that may be applied across a spectrum of competing projects and new project ideas. Projects compete for budget dollars (team size, tools, etc.), personnel (everyone wants the best programmer), resources (time or access to the large database server), and attention from higher level managers. Too many projects fail for political reasons. A project manager uses the various agile mechanisms to minimize politics and keep everything visible and obvious. For in

For instance, the project manager works with the customer to ensure that the product backlog (Scrum) or stories (XP) is visible and everyone understands that it directs the team to the most profitable and valuable work possible. The project manager uses product increments and demonstrations of working functionality to keep everyone aware of real progress against goals, commitments, and visions, thereby minimizing opportunities for rumors, misinformation, and other weapons of political maneuvering. Working with the customer, the project manager helps the customer and organization to value results instead of reports.

## Conclusions

Robert Greenleaf has introduced the concept of the “servant-leader.”<sup>1</sup> Perhaps this is the most appropriate way of thinking of the agile project manager. On an agile project the project manager does not so much manage the project as he both serves *and* leads the team. Perhaps this is one reason why, anecdotally, it seems much more common to see an agile project manager also function as a contributor to the project team (whether writing or running tests, writing code or documentation, etc.).

Plan-driven software methodologies use a command-and-control approach to project management. A project plan is created that lists all known tasks. The project manager’s job then becomes one of enforcing the plan. Changes to the plan are typically handled through “change control boards” that either reject most changes or they institute enough bureaucracy that the rate of change is slowed to the speed that the plan-driven methodology can accommodate. There can be no servant-leadership in this model. Project managers manage: they direct, administer and supervise.

Agile project management, on the other hand, is much more about leadership than about management. Rather than creating a highly detailed plan showing the sequence of all activities the agile project manager works with the customer to layout a common set of understandings from which emergence, adaptation and collaboration can occur. The agile project manager lays out a vision and then nurtures the project team to do the best possible to achieve the plan. Inasmuch as the manager represents the project to those outside the project he or she is the project leader. However, the project manager serves an equally important role within the project while acting as a servant to the team, removing their impediments, reinforcing the project vision through words and actions, battling organizational dysfunctionality, and doing everything possible to ensure the success of the team. The agile project manager is a true coach and friend to the project teams.

---

<sup>1</sup> Servant Leadership, Robert K. Greenleaf.