

---

# Managing a Project Using an Agile Approach and the PMBOK® Guide

Kathy Schwalbe  
schwalbe@augsborg.edu  
Augsburg College  
Minneapolis, Minnesota

## Abstract

This paper includes excerpts from Kathy Schwalbe's text, *Information Technology Project Management, Seventh Edition*, with permission from Cengage Learning. It emphasizes the unique aspects of using an agile approach in each project management process group versus using a more predictive or waterfall approach. An agile approach is often used for projects in which the team cannot clearly express the scope early in the product life cycle. An agile project team typically uses several iterations or deliveries of software instead of waiting until the end of the project to provide one product. This paper suggests that the project management process groups (initiating, planning, executing, monitoring and controlling, and closing) can be used in an agile approach, but there are several different terms, techniques, and outputs.

**Keywords:** information technology, project management, PMBOK® Guide, agile, Scrum

## 1. INTRODUCTION

Agile project management is not new, but it has increased in popularity in the past few years. "Gartner predicts that by the end of 2012, agile development methods will be used on 80% of all software development projects. PMI's research has shown that the use of agile has tripled from December 2008 to May 2011" (PMI 2012).

Many people believe that using an agile approach to managing projects is totally different from a more traditional approach, especially as described in the Project Management Institute's Project Management Body of Knowledge (PMBOK® Guide). In several editions of Kathy Schwalbe's text, *Information Technology Project Management*, a case study was provided describing key outputs produced in initiating, planning, executing, monitoring and controlling, and closing a project to develop an Intranet site for a fictitious company, JWD Consulting. That case used a prescriptive or waterfall approach where the product was

delivered once at the end of the project. The seventh edition of the text includes a second version of the same project, emphasizing the differences in using an agile approach to manage a project. Instructors teaching a course in project management should be familiar with agile project management and be able to explain when it should be used and how it is different from more traditional or prescriptive methods.

Note that teams do not normally take the same project and just decide if they should manage it using an agile approach or not. It's like deciding if you should take a plane or a car on a long trip. You don't just decide on one or the other without specific logic. If you need to get somewhere quickly, have little concern for sight-seeing along the way, and have no problems with flying, you would probably take a plane. If you want to take your time getting somewhere, see sights along the way, and enjoy driving, you would take a car. Likewise, organizations should use logic to decide when to use a more prescriptive or an agile approach to managing

specific projects. Projects with heavy constraints, inexperienced and dispersed teams, large risks, generally clear up-front requirements, and a fairly rigid completion date are best done using a predictive approach. In contrast, projects with less rigid constraints, experienced and preferably co-located teams, smaller risks, unclear requirements, and more flexible scheduling would be more compatible with an agile approach.

In any case, managing projects still involves actions in each of the five project management processes groups. This paper highlights the unique terms, tools, and outputs involved in managing an agile project while still using these five process groups.

## 2. SCRUM ROLES, ARTIFACTS, AND CEREMONIES

Scrum is the leading agile development methodology for completing projects with a complex, innovative scope of work. The term was coined in 1986 in a Harvard Business Review study that compared high-performing, cross-functional teams to the scrum formation used by rugby teams (Scrum Alliance, 2012). When using agile techniques and its most popular method, Scrum, a team uses specific roles, artifacts, and ceremonies.

Scrum includes three main roles for project participants:

- **Product owner:** The person responsible for the business value of the project and for deciding what work to do and in what order, as documented in the product backlog. In this case, Joe Fleming is the product owner. He is the CEO of JWD Consulting and the person who suggested the project.
- **ScrumMaster:** The person who ensures that the team is productive, facilitates the daily Scrum, enables close cooperation across all roles and functions, and removes barriers that prevent the team from being effective. ScrumMasters have authority over the process but not the people on the team. They must be comfortable surrendering control to the product owner and team. Some experts suggest that traditional project managers do not make great ScrumMasters. In this case, Erica Bell will take on the challenge and serve as the ScrumMaster.

- **Scrum team or development team:** A cross-functional team of five to nine people who organize themselves and the work to produce the desired results for each sprint. A **sprint** normally lasts two to four weeks, during which specific work must be completed and made ready for review. Large projects might require teams of teams. In this case, Michael Chen, Jessie Faue, Kevin Dodge, Cindy Dawson, Kim Phuong, and Page Miller are development team members. Kim and Page are client representatives who do not work for JWD Consulting, but they are key team members, especially for developing the parts of the intranet that external clients would use.

In Scrum, an **artifact** is a useful object created by people. An artifact can be called a deliverable in other project management approaches. The following three artifacts are created with Scrum:

- **Product backlog:** A single list of features prioritized by business value. The highest-priority items should be broken down in enough detail for the team to estimate the effort involved in developing them. Some experts suggest scheduling about ten workdays for each item.
- **Sprint backlog:** The highest-priority items from the product backlog to be completed within a sprint. The Scrum team breaks down the highest-priority items into smaller tasks that take about 16 hours to complete. Examples of a sprint backlog and product backlog are provided later in this section under Planning.
- **Burndown chart:** Shows the cumulative work remaining in a sprint on a day-by-day basis. An example burndown chart is provided later in this section under Monitoring and Controlling.

The ScrumMaster facilitates four ceremonies or meetings when using the Scrum methodology:

- **Sprint planning session:** A meeting with the team to select a set of work from the product backlog to deliver during a sprint. This meeting normally takes four hours to a full day.
- **Daily Scrum:** A short meeting for the development team to share progress and challenges and plan work for the day. Ideally the team members are in the same place, the meeting usually lasts no more than 15

minutes, and it is held at the same time and place each day. If that is not possible, teams can use videoconferencing to have short virtual meetings. The ScrumMaster asks what work has been done since yesterday, what work is planned for today, and what impediments or stumbling blocks might hamper the team's efforts. The ScrumMaster documents these stumbling blocks and works with key stakeholders to resolve them after the daily Scrum. Many teams use the term "issues" for items that do not have to be solved in the next 24 hours and "blockers" for items that need to be immediately addressed. This allows a ScrumMaster to maintain focus on highest priority items (blockers) first and then manage the resolution of other issues over the next day or so.

- Sprint reviews: A meeting in which the team demonstrates to the product owner what it has completed during the sprint.
- Sprint retrospectives: A meeting in which the team looks for ways to improve the product and the process based on a review of the actual performance of the development team.

Figure 1 (see Appendices) displays how you can view the Scrum framework in terms of the project management process groups. Creating the product backlog, developing the sprint backlog, and discussing plans during the daily Scrum would fall under planning. Performing the daily work and sprint, and creating the potentially shippable product increment would fit under executing. Holding the sprint review and discussing challenges as part of the daily Scrum can be viewed as monitoring and controlling. Reflecting during the sprint retrospective would fit under closing. Initiating the entire project is a phase that falls outside the Scrum framework in this example.

Table 1 summarizes some of the unique Scrum activities by each process group. The following sections provide more detail on these activities.

**TABLE 1** Unique Scrum activities by process group

**Initiating:**

- Determine roles
- Decide how many sprints will compose each release and the scope of software to deliver

**Planning:**

- Create product backlog
- Create sprint backlog
- Create release backlog
- Plan work each day in the daily Scrum
- Document stumbling blocks in a list

**Executing:**

- Complete tasks each day during sprints
- Produce a shippable product at the end of each sprint

**Monitoring and Controlling:**

- Resolve issues and blockers
- Create and update burndown chart
- Demonstrate the completed product during the sprint review meeting

**Closing:**

- Reflect on how to improve the product and process during the sprint reflection meeting

©Cengage Learning 2014

### 3. Project Pre-Initiation and Initiation

The main difference between pre-initiation in this case and the first, prescriptive case (not shown in this paper but provided in Schwalbe's text) would be determining roles and deciding what functionality will be delivered as part of each release, how many sprints will be required to complete a release, and how many releases of software to deliver, similar to dividing the project into several smaller projects. A project charter, stakeholder register, stakeholder management strategy, and kick-off meeting would still be created as part of initiation, just as they were in the first version of this case. Using the Scrum roles, however, Joe Fleming would be the product owner, Erica Bell the ScrumMaster, and the other people listed in the project charter would be team members.

Joe met with Erica to discuss what functionality will be delivered in each release, how many sprints will compose each release, how many software releases are required, and the approach required to complete the project. They decided that they still needed to survey potential users to collect requirements for the new software and determine a way to measure the value of the intranet site after it was implemented. They estimated that this work would take about two months. They thought that only a few tasks associated with the survey and benefits measurement could be included in the software being developed. For example, two new software features might collect user enhancement requests and collect project benefits information. Joe and Erica would work together to create a survey that specifically asks potential users of the new intranet site which features would provide the most value. For example, instead of listing general ideas, they would list specific features and have respondents rank them in order of importance. They would also ask survey respondents to submit sample templates, tools, and other useful information to Cindy Dawson, a project team member in the IT department. Collecting this information up front would streamline the software development process. Joe and Erica decided that three software releases would be realistic, given the time and cost constraints. Each sprint would be targeted to last four weeks, and a week would be allotted between each sprint for sprint reviews and creation of the product and sprint backlogs.

Joe and Erica had several meetings with Cindy and other team members. Cindy had the most experience working on agile projects. She discussed the importance of submitting several graphical user interface designs to prospective users to gain their feedback. This approach was extremely useful in the past, based on Cindy's experience, and saves a lot of re-work in development. For example, a team member would create 1-4 designs of a particular interface for prospective users to review in sessions hosted by a marketing team. The feedback obtained results in a better user interface that is more intuitive. Also, asking prospective users to rate the "look and feel" is also very beneficial. Users may be distracted by a selected color scheme. Team members should craft the same interface in various color schemes to gauge which one has the greatest appeal and generates the most positive 'usability' score.

Cindy explained that user interface designers are not normally development team members, but they may be employed by organizations to provide detailed design specifications that will provide a consistent user interface. Erica and Joe learned a lot from Cindy and would incorporate her recommendations into further plans.

#### 4. Planning

Instead of creating a team contract, WBS, Gantt chart, and list of prioritized risks for the whole project, the team would follow the Scrum method. A preliminary scope statement for the entire project can still be a useful tool for planning; the team knows that more details will be added and changes will be made. Because Scrum implies that team members work as a self-directed group, coached by the ScrumMaster, a team contract should not be necessary. High level descriptions of the work to be completed would be identified in the product and sprint backlogs. The team would also develop a more detailed list of technical stories and associated tasks to complete during each sprint. The team must also establish a velocity (or capacity) estimate for each sprint based on team member availability each day of the sprint. Estimates can be provided as hours of work (assuming all developers are contractors who can code 8 hours per day) or provided as points (assuming that developers are employees who can code less than 8 hours per day: i.e. six hours per day would equal a point. So, 36 hours of effort would equal 6 points.)

A Gantt chart for the entire project could still be created, as shown in Figure 2 (see Appendices). Notice that the process groups do not follow a simple linear pattern. Several process groups are repeated for each sprint, resulting in several iterations of a useful software product, as shown by the milestones. Recall that in the first version of this case, there was only one release or delivery of software near the end of the project, on October 17. In this case, two additional deliveries of software are scheduled on August 3 and September 11. Some teams create a release roadmap, as described later in this section, for each software delivery. The entire project is still scheduled for completion on November 1.

In addition to listing all of the preliminary project requirements in a scope statement, Joe, the product owner, created a product backlog to prioritize the most important functions of the

new intranet software in terms of adding value to the business. Joe created the product backlog after analyzing the survey results and discussing options with several people. He included separate items for the most important templates and tools, instead of grouping them in one item, and requested a "point person" to contact for questions about each template or tool as well. Notice that this approach combined a few items in the scope statement shown in the first case to focus on what would add the most value. Having a point person would be useful before the team developed the more complex "Ask the Expert" feature. After reviewing the product backlog, the Scrum team would do its planning to create the sprint backlog based on the items the team could complete in the first sprint. Table 2 provides an example of the product backlog and a sprint backlog for the first sprint. The team would then have its first daily Scrum meeting to plan the day.

The Scrum team would break down the items in the sprint backlog into more specific work items, often in the form of user stories, technical stories, and related tasks. **User stories** are short descriptions written by customers of what they need a system to do for them. These descriptions should be about three sentences long and provide the basis for time estimates for the sprint planning meeting. User stories should also be testable and small enough so that the programmers can complete and unit test their code in a timely manner. User stories are composed of technical stories. Technical stories are used by the developers to translate user requirements into technical specifications necessary to create the defined user functionality. The technical stories can contain one or multiple technical tasks that developers can use to chart progress on a sprint board as work is conducted throughout a sprint. One item in the sprint backlog may or may not be documented in one user story, one technical story and one task, based on its complexity. There could be a one-to-one relationship or a many to one relationship if it is highly complex.

**TABLE 2** Product and sprint backlogs

<b>Product Backlog</b>	<b>Sprint Backlog</b>
1. User story templates, samples, and point person	1. User story templates, samples, and point person
2. WBS templates, samples, and point person	2. WBS templates, samples, and point person
3. Project schedule templates, samples, and point person	3. Project schedule templates, samples, and point person
4. Ability to charge customers for some intranet products and services	4. Ability to charge customers for some intranet products and services
5. Ability to collect user suggestions	5. Ability to collect user suggestions
6. Business case templates, samples, and point person	
7. Ask the Expert feature	
8. Stakeholder management strategy templates, samples, and point person	
9. Risk register templates, samples, and point person	
10. Etc.	

For example, someone in the finance area of the company should write a user story called "Ability to charge customers for some intranet products and services." It might read as follows: "As a financial manager, I want our site to use Company B to process payments so that we save money on transaction and customer service costs." User stories are then broken down into a detailed technical story and then into tasks, similar to tasks in a work breakdown structure. See the companion Web site and other resources for more information on creating user stories and other planning documents when using a Scrum framework.

Some organizations use a release roadmap to lay out all the work for an entire release, which can include more than one sprint. It is usually represented as a chart consisting of several columns. The release roadmap gives a clear picture of what stories (scope) will be contained in each sprint, and velocity estimates versus actuals can be easily reviewed and updated. A release roadmap should be derived from data that is contained in the Project Management Information System (PMIS) or related agile management system such as Microsoft's Team Foundation Server, Rally, VersionOne, or JIRA, to name a few.

## 5. Executing

As discussed earlier in this chapter, the most time and money should be spent on executing, where plans are implemented to create the desired product. The team would complete tasks each day, as it would in the first version of this case, but by using an agile approach, the team would produce several iterations of a useful product. For example, at the end of the first sprint, JWD Consulting would have some functionality available on its new project management intranet site. Users would be able to access templates, samples, and point persons for user stories, WBSs, and project schedules, as listed in the first sprint backlog. Users could also make suggestions for functionality they would like to see in the intranet site as it was being developed. The first iteration of the software would also provide the ability to charge customers for some products and services. By using the Scrum methodology, the business could benefit from these new software features a few months earlier than they could using the approach described in the first case.

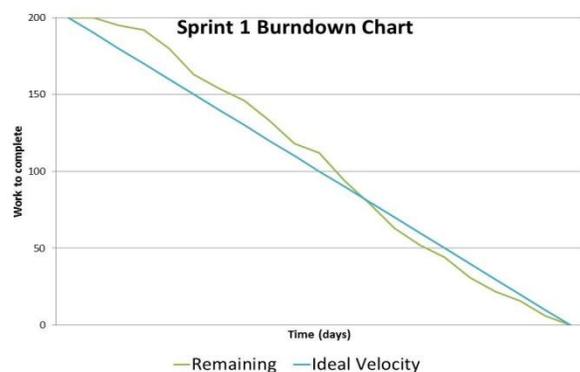
Because Erica and some of the project team would be inexperienced in using Scrum, they would have to work out several challenges. For example, communications are very different because the project team meets every morning, physically or virtually. There would also be communications issues with users of the new intranet site, who might be confused by getting three iterations of the product instead of just one. See Chapters 10 and 13, Project Communications Management and Project Stakeholder Management, for additional information.

## 6. Monitoring and Controlling

The two main items for monitoring and controlling in the Scrum framework are the daily Scrum and the sprint review. The daily Scrum includes a brief discussion of any issues and blockers the team is facing. The ScrumMaster would work with the appropriate stakeholders to address these problems as part of monitoring and controlling. Removing obstacles so the team can perform well is one of the main job duties of the ScrumMaster. Daily Scrums are held each morning to plan and communicate work for the day and bring up any risks, issues or blockers. The ScrumMaster would document these issues and blockers, similar to a list of prioritized risks.

The work progress within a sprint can be represented on a sprint board which is maintained by the ScrumMaster. The sprint board contains cards where each card represents a task to be worked on during the sprint. Each task card contains the control number, task name, estimate to complete, rank or priority number, and team member assigned. As tasks are opened, worked, and closed, they are moved physically around the board to their appropriate section including: Not Started, In Progress, Ready to Test, Tested and Closed. The developers will update status for Not Started, In Progress and Ready to Test. Testers will update status for Tested. The business owner should be responsible to review that the functionality is working as expected and to change status to Closed.

A burndown chart is an important artifact used to graphically display progress on each sprint (see Figure 3). If the first sprint for the JWD Consulting intranet project is scheduled to last four weeks and produce five items or user stories listed in the sprint backlog, the burndown chart illustrates progress for that time period. Each user story should be broken down into specific tasks, and the team should estimate how long it will take to complete each task for each user story. A task should fit within a day, and the work a team member claims in the daily Scrum or stand-up meeting should fit within a day. Each day, the team should again estimate the number of hours remaining in each task. Some tasks might be added and some might be dropped as the scope becomes clearer. It does not matter how many hours are spent; what matters is how many hours of work remain to complete the user stories for that sprint.



©Cengage Learning 2014

### FIGURE 3 Burndown chart

The burndown chart plots each day's sum of estimated hours or points remaining. It also shows an idealized burndown line, as if the team were completing an equal amount of work or 10 tasks each day in this example. The chart then clearly shows whether the team is doing well in that sprint or if there is potential for trouble. Burndown charts help the team understand whether user stories might not be completed within that sprint. If stories are at risk, the chart is an indicator that they should be removed. If progress is better than anticipated, the burndown chart can indicate that stories should be added to the sprint.

At the end of each sprint, the ScrumMaster, or Erica in this example, leads the sprint demonstration review meeting. The team demonstrates to the product owner what it has completed during the sprint. In this case, Joe would review the five features created during the first sprint. After the sprint demonstration review, he would update the product backlog based on the latest information and business needs, and the next sprint cycle would begin.

### 7. Closing

After the sprint review, the ScrumMaster leads a sprint retrospective. During this short meeting (about half an hour), the team reflects on what happened during the sprint. The ScrumMaster normally solicits team member feedback via email first and collates these results before the meeting. This saves time and reduces the discussion to the items that are most important. The retrospective is similar to a lessons-learned report, but it focuses on a shorter period of time. The sprint retrospective is intended to answer two fundamental questions:

- What went well during the last sprint that we should continue doing?
- What could we do differently to improve the product or process?

The sprint retrospective meeting is usually led by the ScrumMaster, who documents a list of action items for implementation. Those items may be added to the product backlog if the product owner agrees with them. For example, after the first sprint for the JWD project

management intranet site project, the team might suggest that the site be mobile enabled. This new requirement could be added to the product backlog and selected as an item for the next sprint. Recall that an agile approach welcomes new requirements as long as they focus on business value.

As demonstrated in reviewing the two approaches to the same project, the agile approach has similarities and differences compared with the traditional approach. If done well, an agile method can produce several iterations of useful software. This approach allows the organization to work together quickly to address changing business needs.

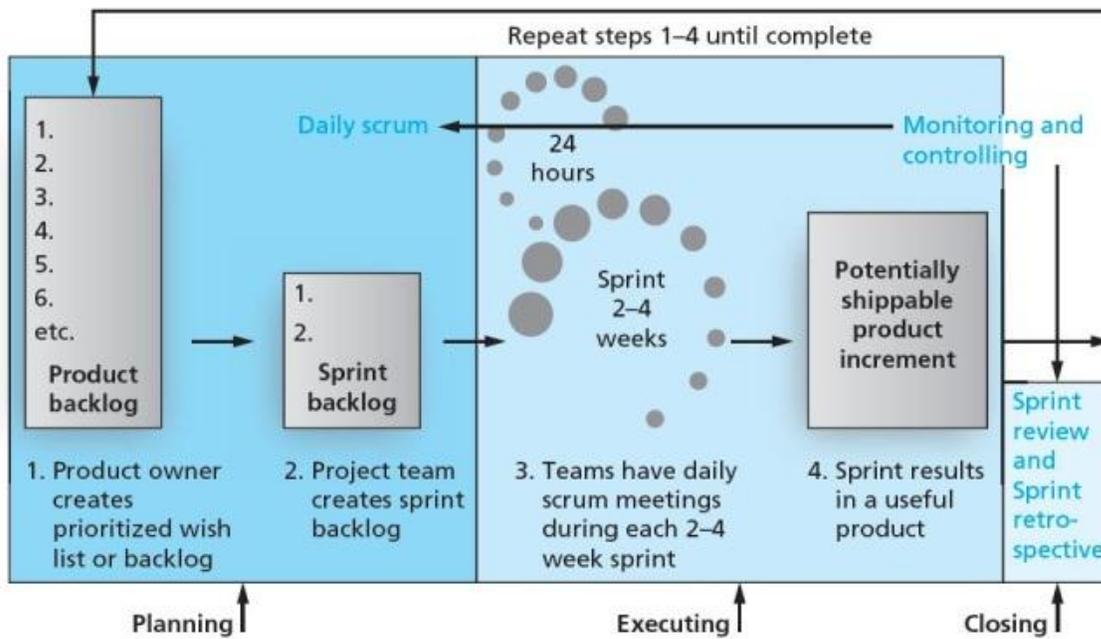
### 8. CONCLUSIONS

This paper demonstrates that you can still use the five process groups described in the PMBOK® Guide to manage an agile project while also using unique aspects of a Scrum approach. The increased interest in agile is based partly on the hope that it will somehow make project management easier. Many books, courses, and consultants are capitalizing on this "new" approach. Seasoned project managers understand that they have always had the option of customizing how they run projects, but that project management is not easy, even when using agile.

### 9. REFERENCES

- Project Management Institute (2012), PMI Agile Certified Practitioner, Retrieved August 13, 2012 from [www.pmi.org/Certification/New-PMI-Agile-Certification.aspx](http://www.pmi.org/Certification/New-PMI-Agile-Certification.aspx)
- ScrumAlliance (2012), Why Is It Called Scrum?, Retrieved August 13, 2012 from [http://scrumalliance.org/pages/what\\_is\\_scrum](http://scrumalliance.org/pages/what_is_scrum)
- Schwalbe, Kathy (2014), Information Technology Project Management, Seventh Edition, Cengage Learning, Boston. (Note: The text will be available in November, 2012.)

10. Appendices



©Cengage Learning 2014

FIGURE 1 Scrum framework and the process groups



©Cengage Learning 2014

FIGURE 2 Intranet site project baseline Gantt chart using Scrum approach