Emerald Insight

OCLC Systems & Services: International digital library perspectives
Understanding agile project management methods using Scrum
H. Frank Cervone

Article information:

Users who downloaded this article also downloaded:

Stephen A. Leybourne, (2009),"Improvisation and agile project management: a comparative
consideration", International Journal of Managing Projects in Business, Vol. 2 Iss 4 pp. 519-535 http://
dx.doi.org/10.1108/17538370910991124

Roopesh Kevin Sungkur, Mayvin Ramasawmy, (2014),"Knowledge4Scrum, a novel knowledge
management tool for agile distributed teams", VINE, Vol. 44 Iss 3 pp. 394-419 http://dx.doi.org/10.1108/
VINE-12-2013-0068

Stephen Denning, (2013),"Why Agile can be a game changer for managing continuous
innovation in many industries", Strategy & Leadership, Vol. 41 Iss 2 pp. 5-11 http://
dx.doi.org/10.1108/10878571311318187

For Authors

About Emerald www.emeraldinsight.com

MANAGING DIGITAL LIBRARIES: THE VIEW FROM 30,000 FEET
Understanding agile project management methods using Scrum

H. Frank Cervone
Purdue University Calumet, Hammond, Indiana, USA

Abstract

Purpose – This paper seeks to define and describe agile project management using the Scrum methodology as a method for more effectively managing and completing projects.

Design/methodology/approach – This paper provides a general overview and introduction to the concepts of agile project management and the Scrum methodology in particular.

Findings – Agile project management using the Scrum methodology allows project teams to manage digital library projects more effectively by decreasing the amount of overhead dedicated to managing the project. Using an iterative process of continuous review and short-design time frames, the project team is better able to quickly adapt projects to rapidly evolving environments in which systems will be used.

Originality/value – This paper fills a gap in the digital library project management literature by providing an overview of agile project management methods.

Keywords Project management, Digital libraries, Project planning

Paper type General review

In the past decade, project management has been undergoing a major transformation as it is applied to information system design. When you consider that traditional project management methodology originated from the body of knowledge of an entirely different domain (engineering, mainly of the industrial and civil kind), it is not surprising that project management in the information systems arena has evolved.

Many of the methods for developing systems originated in computer science which itself emerged from constructs used in engineering and mathematics. In the early days of computing this sufficed, however today, developing systems is much more than just engineering the most technically correct or "best" system. In some cases, what we as users may want is not the "best" system. Frequently, what we really want is the most practical system that is focused on and addresses our particular needs.

When we consider traditional project management and software development approaches, several disadvantages are immediately evident. For one, the huge effort required during the planning phase of a traditional project is often so all-encompassing that half (or more) of the resources for the project are expended before any development work even beings. Furthermore, requirements definitions are often so labor intensive and protracted that the requirements for the project have changed before development even begins.

From this context, agile project management developed.

Overview of agile project management

Agile project management is an outgrowth of the agile software development movement. While the origins of agile project management can be traced back to ideas from a paper by Takeuchi and Nonaka in the January 1986 issue of the *Harvard Business Review*, it was not until Jeff Sutherland and Ken Schwaber discussed the first agile method for software development at the 1995 OOPSLA conference that the idea of gained traction. While analyzing common software development processes, they found that traditional development approaches were not suitable for empirical, unpredictable and non-repeatable processes.

Today, there are several different approaches to implementing agile methods but underlying all of the various agile movements are some basic concepts that turn traditional methodologies on their head. The "Manifesto for Agile Software Development" stated four core principles:

(1) Individuals and interactions over processes and tools.

(2) Working software over comprehensive documentation.

(3) Customer collaboration over contract negotiation.

(4) Responding to change over following a plan.

Agile project management is deeply rooted in these principles but slightly modified to make sense in the project management, rather than software development, environment. This can be seen in some of the qualities of the agile project management approach. For example, agile project management emphasizes two important concepts. The first is that risk is minimized by focusing on short iterations of clearly defined deliverables. The second is that direct communication with partners in the development process is emphasized in lieu of creating copious project documentation. The reasons these two concepts are emphasized is simple: both help a project team adapt quickly to the unpredictable and rapidly changing requirements most development projects are carried out in.

What are agile methods?

Just as there are many types of projects, there are several different takes on how to best apply agile methods. Some of the most important include: Scrum, extreme project management, adaptive project management, and dynamic project management method. Of these, the general model of Scrum is most often used.

Those who follow rugby know that a Scrum is a way to restart the game after an interruption. Specifically, the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among the team members. If you cannot envision this, do not worry. Much of the terminology of agile methods is derived from rugby terms; it would seem partly as a way of bringing some levity into the process.

In terms of agile project management, a Scrum is simply an agile, lightweight process for managing and controlling software and product development in rapidly changing environments. Like a Scrum in rugby, it shares many of the same characteristics. For example, agile project management Scrums are intentionally iterative, incremental processes that are predicated on a team-based approach. Given that systems today are usually development in fluid and rapidly changing

environments, one of the major reasons for using an iterative process is to help control the chaos that can result from conflicting interests and needs within the project team. Additionally, iterative processes are used to help enable improvement in communication, maximize cooperation, as well as protect the team from disruptions and impediments. Overall then, the goal is to deliver a more suitable product more quickly than with traditional methods.

Exploring Scrum in depth

The Scrum model is built on three major components: roles, process, and artifacts. The Scrum Master is the role traditionally assumed by a project manager of team leader. This person is responsible for several things, perhaps the most important of which are enacting the Scrum values and practices and removing impediments. The Scrum team typically is a cross-functional team of consists of five to ten people who work on the project full time. The team is self-organizing, which has been interpreted in various ways, but most often means that the leadership role within the team is not fixed and changes depending on the needs of the specific iteration (known as a sprint) in process at the time. It is important to note that membership of the team only changes between sprints. The product owner is typically a functional unit manager who knows what needs to be built to enable the project and how the sequence of builds should progress.

The Scrum process has five major activities: the kickoff, the sprint planning meeting, the sprint, the daily Scrum, and the sprint review meeting. The sprint planning meeting is a meeting of the Scrum team, the Scrum master, and the product owner at the beginning of each sprint (iteration). These meetings, which may take up to a day, consist of two parts. In the first part of the meeting, two major activities occur. First, the group defines the product backlog, which is basically a list of the project requirements. After this, the group determines the sprint goal, which is the formal outcome(s) from this particular sprint. In the second part of the meeting, the focus of work is on creating the sprint backlog. The kickoff meeting is structured similarly to the sprint planning meeting with the major difference being that the group define the high-level backlog for the project and the major project goals.

Once the sprint planning meeting has been held, the sprint can begin. Sprints differ from phases in a traditional project in that sprints are limited to a month-long iteration cycle in which time the functionality of the product is further developed. Another differentiating factor is that during a sprint, no outside influence should be allowed to interfere with the work of the Scrum team. This has several potential implications with the most important being that project requirements cannot be changed during a sprint.

In many projects, but not all, each sprint begins with a daily Scrum meeting. This meeting, typically lasting no more than 15 minutes, is held every day between the Scrum master (who chairs the meeting) and the Scrum team. In this meeting, every team member briefly answers three questions:

(1) What did you do since the last Scrum?

(2) What are you doing until the next Scrum?

(3) What is stopping you getting on with your work?

While it might not be evident, the daily Scrum is not a problem solving session and is not really designed to be a way of collecting information about who (or what) is behind schedule. Instead, the purpose of the daily Scrum is to both track the progress of the

team as well as allow team members to make commitments to each other and the Scrum master so that work can proceed in the most expedient and unimpeded manner.

The sprint review meeting is held at the end of each sprint. During the meeting, the functionality that was created during the sprint is demonstrated to the product owner. Perhaps what most differentiates this meeting from a meeting in traditional project management is that this meeting should be informal and not be a distraction for the team members.

The last major component of the Scrum model is the Scrum artifacts that include the product backlog, the sprint backlog, and burn down charts. The product backlog is the requirements for the project expressed as a prioritized list of backlog items. Unlike a traditional project, this list is managed and owned by the product owner. This list can be created using project management software (like MS-Project) but can just as effectively be created as a spreadsheet. In most projects, the product backlog is a major deliverable of the kickoff or spring planning meetings. As is the case with sprints, the product backlog cannot be changed until the next sprint planning meeting.

During the sprint planning meeting, the team performs an estimation of each product backlog item. Two methods of review are typically used, expert review or creating a work breakdown structure. These estimates are specifically intended to be forecasts and not exact measurements. Regardless of method of estimation chosen, the estimation includes placing the backlog item into a size category, discussing the story points (a relative measure of the complexity of a particular feature within the project) and using that to estimate the amount of hours or days of work that will be involved to complete the item. Based on this estimation, a collective decision can be made that establishes the team's velocity or amount of effort that can be reasonably handled during one sprint.

Similarly, the sprint backlog is the subset of product backlog items that are defined as part of the work for a particular sprint. However, unlike the project backlog, the sprint backlog is created only by the Scrum team members. Ideally the sprint backlog is updated every day and contains no more than 300 tasks. The team may need to break down a task if it is determined that it will take more than 16 hours. Furthermore, the team may determine that items may need to be added or subtracted from the sprint but this is the team's decision, it is not something that is directed by the product owner.

Unlike traditional project management, Scrum intentionally focuses on work done through the use of burn down charts. Three types of burn down charts are commonly used: the sprint burn down chart documenting the progress of the sprint, the release burn down chart documenting the progress of the release, and the product burn down chart documenting the overall project progress. A goal of a burn down chart is to provide information in an easy to comprehend manner. As such, each task is typically represented in terms of time (the x-axis of the display grid) and duration (the y-axis).

For example, a typical sprint burn down chart would depict the total backlog hours remaining in the sprint per day as an estimated amount of time left in the sprint. Ideally, the sprint burn down chart would "burn down" to no time remaining by the end of the sprint; however as during the sprint, setbacks could result in an increase in estimated time, not all burn down charts do burn down to zero. The release burn down chart functions in a similar way but represents the remaining time until the release will be done. Not surprisingly then, the product burn down chart is used to indicate the overall project progress.

22

Conclusions

The advantages of agile project management and particularly the Scrum-based approach is its simplicity. Within an agile project, roles are clearly defined and do not cross boundaries. Features can be completely developed and tested in short iteration cycles. Because each team members bears major responsibility for their part of the project, ownership of the project is more broadly based. The methods of agile project management enforce extensive communication, which helps teams organize more effectively. This, ultimately, can lead to greater productivity for everyone involved.

However, some cautions are in order before embarking on a project managed with agile methods. Frequent communication does not mean that no documentation is required and without adequate oversight, a project could degenerate into "undisciplined hacking" because documentation is not maintained. It is also difficult for novice agile project managers to enforce some of the most important facets of agile methods, such as no changes during a sprint. In order to effectively implement an agile approach, all parties involved including upper management must be committed to the process.

Nonetheless, agile project management is one of the most effective means of enhancing the development cycle within projects. By focusing on eliminating unnecessary bureaucracy, process, and practice in managing the project, the methodology makes it possible for all parties to eliminate administrivia and actually do productive work.

Further reading

Deemer, P., Benefield, G., Larman, C. and Vodde, B. (2010), "The Scrum primer". Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf?1285931497

James, M. (2010), "Scrum reference cards", available at: http://scrumreferencecard.com/

Schwaber, K. (2004), *Agile Project Management with Scrum*, Microsoft Press, Redmond, WA.

Schwaber, K. (2009), detailed information on Scrum methods available at: www.Scrum.org

This article has been cited by:

1. Michael Jahr. 2014. A Hybrid Approach to Quantitative Software Project Scheduling Within Agile Frameworks. *Project Management Journal* **45**:10.1002/pmj.2014.45.issue-3, 35-45. [CrossRef]
2. H. Frank Cervone. 2014. Improving Strategic Planning by Adapting Agile Methods to the Planning Process. *Journal of Library Administration* **54**, 155-168. [CrossRef]