Scrum Software Development Process

Adrian Kosmaczewski

2006-12-28

In this article I give an overview of the Scrum Software Development Process. This methodology was first described by Takeuchi and Ikujiro in their 1986 book "The New New Product Development Game", and was initially meant to manage any kind of product development project. This methodology, for example, has been used in "real" product development projects by companies such as Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox and Hewlett-Packard (Schwaber). In the nineties it was adapted for software development projects.

Description

The name "Scrum" is borrowed from the game of rugby, and the following paragraph by Gartner describes the purpose and context of a Scrum in a rugby game:

"A SCRUM is also the name of the formal conglomeration of forwards who bind together in specific positions when a scrumdown is called. It is the basic set formation of rugby and occurs after various minor infringements of the law, when the ball becomes tied up, and other times you'll learn about later. It is a face-off of sorts and a favorite among forwards. Form and timing are more important than brute strength (although we'll take some brute strength). A birds-eye diagram might make things more clear:"

"1: Loose Head Prop (sturdy and fearless) 2: Hooker (small, quick, ready to take control) 3: Tight Head Prop (see #1) 4,5: Second Rows (Locks) - (big and strong) 6,7: Wing Forwards (Flankers) - (quick, aggressive) 8: Number Eight (smart, foot and hand skills) 9: Scrumhalf (smart, experienced, quick) – technically not a forward, but the link between forwards and backs - special rules apply to the scrumhalf."

(Source: Gartner)

The official definition of the Scrum Software Development Process is

> Scrum is an agile, lightweight process that can be used to manage and control software and product development using iterative, incremental practices. Wrapping existing engineering practices, including Extreme Programming and RUP, Scrum generates the benefits

> of agile development with the advantages of a simple implementation. Scrum significantly increases productivity and reduces time to benefits while facilitating adaptive, empirical systems development.

(Scrum website, 2006)

The most important difference between Scrum and other methodologies is that Scrum does not defines how systems will be developed, using a defined step-by-step guide, but rather how to coordinate the work among team members:

"In this paper we introduce a development process, SCRUM, that treats major portions of systems development as a controlled black box." (Schwaber)

In this sense, Scrum acknowledges the fact that software development is an inherently chaotic process, complex and often non-deterministic. This lack of determination is given by the following factors (Wikipedia):

- Requirements are never understood at the beginning of the process
- Requirements change
- New technologies bring uncertainty to the processs

As such, Scrum "wraps" existing engineering practices, providing (using Scrum website's own words):

- Complete risk management;
- Ways to manage conflicts;
- Iterative processes in contexts of rapidly evolving requirements;
- Improved communication paths;
- Ways to increase productivity;
- Scalability from small to enterprise-wide projects.

Pragmatics

The Scrum methodology consists of the following phases:

1. **Pregame:** this phase holds the planning, and high level analysis and architecture phases. During this phase, the team defines the scope of the project, the milestones dates, deliverables and delivery dates, performs risk assessment and gets approval and funding for the project, as well as defining and reviewing the architecture of the proposed solution. One team member is named the "scrum master", who "leads the Scrum meetings, identifies the initial backlog to be completed in the sprint, and empirically measures progress toward the goal of delivering this incremental set of product functionality." (Rising and Janoff, 2000)

2. **Game:** the development of the system is done in this phase, during what is called a "Sprint", which is the core and most distinctive element of the Scrum methodology. A sprint is an iterative cycle of one to four weeks, during which developers create the software product. "5 minute daily meetings" are used by the team to quickly coordinate the development

tasks for a particular day. Sprints consist of the following high level tasks (Schwaber):

- Develop
- Wrap
- Review
- Adjust

> Daily meetings allowed everyone on the project team to see the status of all aspects of the project in real time. This allowed the collective neural networks of the team's mind to fine-tune or redirect efforts on a daily basis to maximize throughput. The result was radical alteration of the software development process by allowing sharing of software resources. Development tasks thought to take days could often be accomplished in hours using someone else's code as a starting point.

(Sutherland, 2004)

After each sprint, the team performs a review meeting, where team and management members participate, needed to evaluate the work done in the previous sprint, and to define and prepare the next one. Three questions define all that is needed to communicate during that meeting: (Rising and Janoff, 2000; Sutherland, 2004)

- What have you completed since the last scrum meeting?
- What obstacles got in your way?
- What do you plan to accomplish until the next scrum meeting?

1. **Postgame:** this phase contains the closure of the project, including the preparation of the deliverables for installation, integration testing and maintenance.

Criticism

As any other methodology, Scrum does not apply to all situations. As Brooks said, "there is no silver bullet", nor project methodology that applies to all situations. Scrum applies particularly well to projects with a high level of uncertainty (undefined scope, uncanny technologies, new unexplored business models, etc).

Also, in the positive side of the equation, I think that the tight focus on project risks is one of the most important element of the Scrum methodology. Continuously keeping an eye on obstacles, provides a way to react quickly to possible bottlenecks or problems that might appear during the development process, for which other methodologies do not provide similar quick answers.

On the downside, development teams and project managers should be trained in using the Scrum methodology. I do not think that any project can jump from a formal, "classical" methodology based in defined constraints and feature sets (which usually maps to a more hierarchical team structure), to a model where everyone participates at the same level, providing continuous feedback about possible problems, and communicating in fluid ways. I think that the main problem in adopting Scrum might be a cultural one, rather than a pure methodology one.

As one team leader comments about Scrum, “Give it time to get started before expecting big results. It gets better as the team gains experience” (Rising and Janoff, 2000).

Conclusion

The Scrum Development Process is an interesting example of how important is the acknowledgement of the inherent chaos in the activity of software development, and how important is to communicate clearly and often, particularly when requirements are not defined completely, or when there are other degrees of uncertainty. This recognition leads to a redefinition of the problem, and thus a different viewpoint for managing software projects, in a non-deterministic way. However, as always, a clear definition of the context is needed to ensure the success of the project, choosing the good methodology.

References

Beck, Kent, Thomas, Dave, Fowler, Martin et al., “Agile Manifesto”, 2001, [Internet] http://agilemanifesto.org/ (Accessed June 4th, 2006)

Beck, Kent, “What is Extreme Programming?”, 1999 [Internet], http://www.extremeprogramming.org/ (Accessed June 4th, 2006)

Frederick P. Brooks, Jr., “The Mythical Man-Month - Essays on Software Engineering, Anniversary Edition”, 1995, Addison Wesley, ISBN 0-201-83595-9

Gartner, Lisa, “The Rookie Primer”, Radcliffe Rugby Football Club, [Internet] http://hcs.harvard.edu/~radrugby/rookie_primer.html (Accessed June 4th, 2006)

Rising, Linda and Janoff, Norman S., “The Scrum Software Development Process for Small Teams”, IEEE Software July/August 2000 [Internet] http://members.cox.net/risingl1/articles/IEEEScrum.pdf (Accessed June 4th, 2006)

Schwaber, Ken, “SCRUM Development Process”, [Internet] http://jeffsutherland.com/oopsla/schwapub.pdf (Accessed June 4th, 2006)

Scrum website, [Internet] http://www.controlchaos.com/ (Accessed June 4th, 2006)

Dr. Sutherland, Jeff, “Agile Development: Lessons Learned from the First Scrum”, October 2004, [Internet] http://jeffsutherland.com/Scrum/FirstScrum2004.pdf (Accessed June 4th, 2006)

Takeuchi, Hirotaka, and Ikujiro, Nonaka, “The New New Product Development Game”, Harvard Business Online, [Internet] http://harvardbusinessonline.hbsp.harvard.edu/b01/en/common/i
(Accessed June 4th, 2006)

Wikipedia, “Scrum (development)”, [Internet] http://en.wikipedia.org/wiki/Scrum_%28development%29
(Accessed June 4th, 2006)