

Study on Application Case of Scrum Methodology using Visibility

Eun-Sun Chang¹, Neung-Hoe Kim^{2†}

¹*Service PM, Kakaopage, Seongnam, Korea*
Invoke1@naver.com

²*Department of Computer, Korea University, Seoul, Korea*
nunghoi@korea.ac.kr

Abstract

Viewed in the rate of change in the web environment, it is very difficult to remain the initial planning at the time until the time of launch, and there is a need for a method to accommodate changes and satisfy market demands during the development process. Unlike the traditional waterfall approach of maintaining initial planning, scrum is one of the agile methodologies that enables flexibility to respond to changes in the market and customers' needs and drive customer satisfaction and business success. However, to apply the scrum to a project in actual, the practice method itself is relatively simple but not easy to apply. The reason is that the members of the organization need to understand and participate in scrum's philosophy and principles and the continuous observation and change management should be carried out. Therefore, in this paper, we presented the feature dashboard and customized scrum methodology to enable continuous observation and change management using visibility, and we shared the case that periodically reflected inspection and adaptation with the explanation of the main points. Also, based on the experience with participants, the strengths and weakness of the feature dashboard and the customized scrum methodology are summarized.

Keywords: *Agile methodology, Scrum, Software visualization, Software development methodology*

1. Introduction

Viewed in changing speed of the web environment, it is rather strange that the contents of initial plan are fixed to the time of release, the method to accommodate change and meet market demands during development is required. The waterfall approach which must keep the initial plan to the end does not satisfy this mission, and the transition to a repeat and gradual development method is necessary in order to increase the success chance of project. The key of agile methodologies is to driving customer satisfaction and business success by flexibly responding to change in market and customer needs, and quickly applying market response through repeatable, continuous release of product of short cycle through forming of small teams with enough expertise and experience. However, applying scrum, one of the specific methodologies of agile, to actual projects is

relatively simple in itself, but difficult to apply because it is required that the entire members of organization understand the philosophy and principles of the scrum and need to voluntary participation, and always observe that it is being applied well. In addition, changes are generated dynamically, so backlog priorities and grouping operations must be continuously managed from the beginning to the end of the project, but estimates are only estimation that close to the correct answer of that time and are not a promise to develop them within the estimated time frame, so it difficult to apply to projects where compliance with a release date that is important when the development deadline is set [1].

Accordingly, through the feature dashboard and customized scrum methodology, we'd like to share cases that reflected periodic inspections and adaptations based on the accuracy of estimates that were difficult to predict and manage and the high visibility about plan and performance by iteration for additional/change backlogs and delayed task.

This paper is organized as follows. Chapter 2 describes the scrum methodology, chapter 3 presents the iterative & incremental development methodology that has been customized and applied, chapter 4 shares the cases applied to the project, and chapter 5 presents the conclusions and future plan.

2. Background

Scrum is the agile software development methodology for project management, which is close to philosophy. This moves toward the goal, repeating the process over and over to discover and correct the problem.

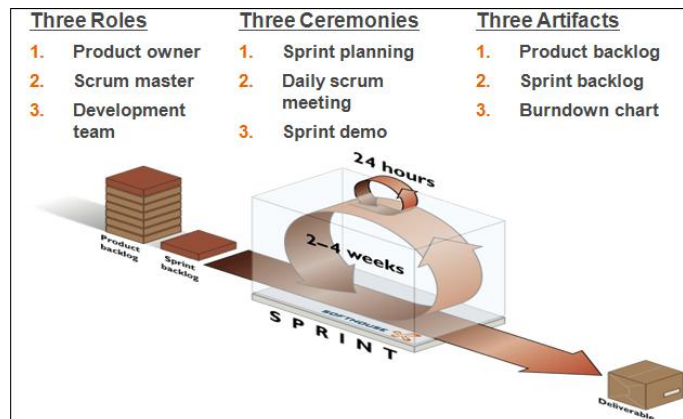


Figure 1. Scrum Methodology

As shown in Figure 1, scrum clearly defines the role of the matters concerned and then defines activities such as sprint planning meetings, daily meetings, demos and products/sprint backlogs, decimation charts.

According to the relevant research, there is a case study reports that actual scrum methodology is applied to projects and improve the quality of outputs and reduce the lead time in the development process [2], and it reduce re-work at the sprint stage [3].

3. Iterative & incremental development methodology

The iterative & incremental development methodology, a customized scrum methodology, is summarized as like Figure 2 and the main points are as follows.

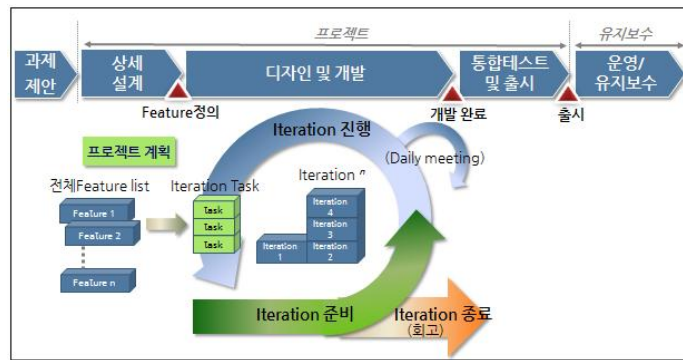


Figure 2. Iterative & Incremental Development Methodology

3.1 Definition of Feature

It supports the communication and decision making of all activities that occur in the process of the project, with the specifications of functional and other non-functional factors that give value to the users who want to provide within the scope of the project.

Based on a higher-level plan, each feature should be described in the form of a use case or a user's view of the form of story in a language that service and developer can understand, and it shall be possible to determine clearly whether the function is complete. The feature list has been completed by agreeing the completion criteria for each feature by combining the service functional elements created by planning and the non-functional elements created by the development team. The template for creating the feature list is shown in Table 1.

Table 1. The Example of Feature List Template

1 Depth	2 Depth	3 Depth	4 Depth	Feature name	Feature description	Detailed design	Priority	Estimated work time

- ① Key feature is derived from planning requirements centering around story. The priority is written on a 5-point scale (1-5).
- ② Features are written by dividing them into appropriate sizes (scale that does not exceed 10MD with 1~2MD level) according to the recommended criteria for the feature division. This is because it is desirable for one feature to be completed within one iteration.
- ③ Feature list should be created with 1 depth ~ 4 depth and maximum 4 depth or less. The object and task should be described in the form of object + predicate (what + to ~) with depth structure for that each depth is too abbreviated or do not write insufficiently enough to do not understand what tasks users can perform.
- ④ Project members estimate and agree to create an estimated time of work, and sum the representative values to calculate the total estimation value [4].

3.2 Project Plan

- ① Iteration period setting: the iteration period should be set according to the project period and environment, but recommend 2-3 weeks, and perform it. Then, discuss the completion criteria of the feature.
- ② Feature list completion criteria: the test methods for the feature/task assigned to each iteration and completion criteria should be defined. All person in charge decide by agreement upon the completion criteria for the features completed by person in charge of QA.
- ③ Definition of member role: the clear role such as scrum master, product manager, scrum team member, and participant assigns to all member of project.
- ④ Communication measures: the organizer of each meeting such as the iteration preparatory meeting, daily meeting, and closing meeting, and attendees and observers shall be determined, and the venue and time for each meeting shall be determined.
- ⑤ OA inspection plan: since the development result is released from initial iteration and demonstration is performed, the test activities can be performed on completed features.

3.3 Iteration Production

- ① Preparation of iteration: it is a stage in performing system configuration and SW structural design and detailed design to implement the development scope.
- ② Progress of iteration: it is a stage to complete implementation of source code by conducting various quality activities for source code preparation and code quality. The obstructive factors should be removed through the daily meeting.
- ③ Completion of iteration: all project team members and the stakeholder of the project participate and take a review time to communicate information about the iteration [5]. All of the features that were to be done to the iteration are produced in a workable form, which are aimed at enhancing visibility into the progress status of the project by demonstrating these operating outputs, and getting immediate feedback from the person in charge of project. In addition, for the work progressed through task re-estimation, the person who conducted the work estimates and records the remaining work time again.

It was managed using Jira as other relevant tools. At this time, it is possible to estimate the estimated time and completion time of the feature through regular work log time tracking work by using the burn down using the GreenHopper as a plug-in, and it is also possible to see the change in the overall estimation. At this time, the important thing is version control, and two versions are controlled for feature dashboard management. The generated version records the history of all iteration for which put the worker's effort and time for the task, and if one work of the planned tasks in the iteration-1 is not "resolved" in the iteration-1 and proceed in the iteration-2 and it is "resolved," the both versions of which put worker's effort and time are recorded. The revision is a version that records the iteration in which the task is "resolved". If the work is progressed in iteration-1, iteration-2, and iteration-3 as above and the work is resolved in iteration-3, only one version of iteration-3 that such work has been processed finally was recorded.

4. Analysis of application result

4.1 Strengths and Weaknesses of Feature Dashboard

- Strength
 - Regularly identify performance compared to plan based on frequent sharing and fast feedback
 - Facilitate visibility of features and production capacity for add/change/delay (KLOC)
 - Facilitate manage the expected resource and input resource data by each task
 - Available to check the distribution of division size for feature

- Weakness
 - Seeking of the countermeasure assume that securing available resources or adjustment of specification priorities are available
 - Necessary of visibility due to the missing dashboard of sub-task tasks for non-functional elements

4.2 Strengths and Weaknesses of Iterative & Incremental Development

- Strength
 - Efficient as project progress is visible and risks can be detected early
 - Satisfaction of customer needs through fail fast and fast feedback through demonstration in every return
 - Quality improvement of development output
 - Faster time to delivery

- Weakness
 - Obscuring the role and the responsibilities of the members at each stage if failure to clarify this at the starting stage
 - Occurrence of risks that work is pushed to the time of deadline or the schedule is delayed if the work is not completed within iteration due to the internal/external factors
 - Difficulty of calculation and estimation for initial schedule (it required a lot of time and the accuracy is not high)

5. Conclusion and future plan

This paper suggests the feature dashboard and customized scrum methodology to enable continuous observation and change management using visibility, and we shared the application case with the explanation of the main points. In addition, the strengths and weakness of the feature dashboard and the customized scrum methodology are summarized by discussing with the participants of the application cases. Not only were all production indicators visible by applying feature dashboard and customized scrum methodology to the project, but also improved the customer satisfaction from the positive responses from project members and created the new development culture from demonstration of development outputs by applying fast response to progress and quality through periodic monitoring such as number of iterations in performance compared to plan, number of features, and input M/D. In addition, the analysis of project size became possible by identifying the feature distribution, the utilization of available resources for important projects of the company is increased, and it has effect to reduce the overall schedule and cost of the project. Also, the measurements by project and organization can be possible by collecting output and velocity.

In the future, based on the collected data, we will specifically classify whether the project scope & quality & cost will be improved or not, trend grasp of estimation accuracy, impact on QA and maintenance of operation. And the feature dashboard and customized scrum methodology will be improve and supplement so that they can be safely established and executed. Also, as it is important for participants to understand the philosophy and principles of the scrum, the experiences through continuous education and workshops will be carried out together.

References

- [1] K. Schwaber, M. Beedle, *Agile Software Development with Scrum*, Pearson, 2001.
- [2] N.Y. Hong, J.B. Yoo, and S.D. Cha, "A Practical Study on the Application of Scrum to Outsourcing E-commerce Projects in Korea," *Korea Computer Congress 2010*, pp. 64-65, Jun. 30-Jul. 2, 2010.
- [3] J.K. An, H. In, D.H. Lee, and N.H. Kim, "An Agile Scrum Process based on Requirement Simulation Tool for Offshore Outsourcing Project," *Korea Software Congress 2009*, pp. 31-35, Nov. 27-28, 2009.
- [4] M. Cohn, *Agile Estimating and Planning*, Prentice Hall, 2005.
- [5] K. Schwaber, E. Derby, and D. Larsen, *Agile Retrospectives: Making Good Teams Great*, Pragmatic Bookshelf, 2006.