Customization of Scrum Methodology for Outsourced E-commerce Projects

Nayoung Hong
Dept. of Software Engineering

Korea University
Seoul, Republic of Korea
Email: h4022@nate.com

Junbeom Yoo
Div. of Computer Science
and Engineering
Konkuk University
Seoul, Republic of Korea
Email: jbyoo@konkuk.ac.kr

Sungdeok Cha
Dept. of Computer Science
and Engineering
Korea University
Seoul, Republic of Korea
Email: scha@korea.ac.kr

Abstract — **This paper describes how scrum method was customized for outsourced e-commerce software projects. While the waterfall process was used in the past, outsourced projects experienced more delays and failures than the ones conducted in-house. To overcome such limitations, we decided to tailor the scrum method on three aspects: First, we produced a table that explains roles and responsibilities of project team members for every phase of the Scrum methodology. Second, we divided sprint planning into two phases, a master sprint plan and individual sprint plans. Finally, we monitored project progress based on the number of completed web pages. Application of the modified scrum method on two projects not only improved product quality but also reduced time necessary to complete the project. More than 80% of the software engineers also expressed satisfaction of the proposed approach.**

Keywords-component; scrum, agile, software engineering

1. INTRODUCTION

While many e-commerce companies hire third parties to develop new services on time and reduce costs, these outsourced projects had lower success rates and poor quality. One e-commerce company in Korea currently develops all projects using the waterfall model [1]. Statistical data from this company shows that internally developed projects had a success rate of 89% out of 137 projects completed in 2009. Despite higher success rates for internal projects, outsourced projects had success, delayed, and failure rates of 60%, 20%, and 20% respectively. The quality of the product was also negatively affected. The defect ratio and the fatal defect ratio were much higher than those of internal projects of similar scale, with rates of 17.7% and 6.9% respectively.

Outsourced projects performed in 2009 were reviewed and analyzed for the causes of the failures. We found that the waterfall model was the main cause of the failures. We examined features of completed outsourced projects. Waterfall is a step-down process that was successfully used for many years. It has had incredible success-rates when it comes to completing small projects (<12 man-months) and projects where there are known requirements. We had successfully applied waterfall to internal developments and completed 94% of projects. The remaining 6% of the company's projects were large projects (>12 man-months) that had been outsourced to third party developers. In spite of this high success rate for internally developed projects using waterfall, when projects were outsourced to third parties, the success and defect rates had substantially decreased and increased, respectively, when waterfall was applied. This method therefore was found to be unsuitable for outsourced projects.

Agile methodologies [2] are software development processes that can cope with unfixed and easy-to-change requirements. A survey conducted by Cranky Product Manager in Oct. 2008 [3] reported that more than 60% of the software industry utilized the waterfall methodology until 2006. By 2008 the trend shifted to agile methodologies. Two of the common ones in use, scrum and a hybrid XP/scrum, have become popular among software developers. Its popularity can be attributed to a 22% on-time delivery rate and a 21% acceptance-to change requirements [4].

The Scrum method is popular because it can be easily and quickly applied, but it has the following problems in its implementation as is. First, roles & responsibilities (R&Rs) are not clear among those responsible for the execution of the process in the initial phase. Second, during the planning phase, the original scrum method only focuses on incremental sprints, therefore not allowing a clear picture of the overall project schedule. This limitation makes it difficult to make projections of deadlines that ultimately affect multiple processes or units in the development cycle. Finally, progress reports using burn down charts based on estimation points was very inconvenient during the review phase due to re-calculating estimation-points at the end of each sprint.

A few modifications to the original scrum have been proposed to address these problems. First, roles & responsibilities need to be clearly presented to those who are experiencing scrum for the first time, along with a step-by-step activity list. Second, an additional planning stage was added in order to plan and outline the project as a whole. The difficulty of coordinating sprint plans, due to multiple interdependent operations, could be resolved via sharing of the status bulletin where expected start dates and expected end dates were entered. Finally, for the Sprint review phase, project progress was reported based on the number of completed web pages.

The customized scrum method was applied to two outsourced projects in Korea, in order to demonstrate its effectiveness. This study shows that the method can enhance the quality of products, reduce idle time in each phase and accomplish timely delivery of outsourced development projects. The Scrum methodology is also proven to be effective in improving customer satisfaction.

The remainder of the paper is organized as follows: Section 2 introduces the existing scrum process and

application of the customized scrum process is demonstrated in Section 3. Section 4 explains project performance when the customized scrum process was applied. We conclude the paper with Section 5.

2.   RELATED WORK

Agile Methodologies, such as scrum, have recently become a hot issue. Agile (swift and efficient) methodologies refer to various applications that enable agile development. Unlike the waterfall methodology where the success of a project lies solely with the project manager, an agile methodology focuses on team work and cooperation to deal with any changes in project demands. This methodology has benefitted such international companies as Hewlett-Packard, IBM, Yahoo, Google, Microsoft, and domestic Korean groups that include Daum Communications, NeoWiz, NHN, M-Game and Open Maru [5, 6, 7, and 8].

According to the survey "Enterprise Agile Adoption" in 2007 by Forrester Research [8], 25% of U.S. and European enterprises have adopted an Agile Methodology for development. By 2007, the adoption rate had rapidly increased by as much as 2-3 times. Also, according to Scott Ambler's "Agile Works in Practice" [9], which targeted 4,232 IT professionals in 2006, improvements in production rates by 60%, quality by 66%, and business satisfaction by 58% were shown.

Some well-known agile methodologies include Extreme Programming, Lean Software Development, Adaptive System Development, Crystal Method, Dynamic System Development, Feature Driven Development, and scrum. The survey "The State of Agile Development" performed in 2008 by Version One utilized only 49% of scrum and the remaining 22% were mixed with XP.

Scrum is the name given to a type of iterative, incremental framework for software development project management. It was first conceptualized in Japan in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka. They envisioned a holistic approach that would increase speed and provide better flexibility in the development process of new commercial products. Each incremental cycle, or sprint, is normally fixed, or "time-boxed", from one to four weeks [2, 10].

Each sprint begins with a sprint planning meeting that prioritizes the demands of the project from the projects backlogs. The scrum team then endeavors to complete each item associated with that periods sprint. Each item listed in a sprints backlog remains fixed for the duration of the sprint and cannot be changed mid-sprint. Daily team meetings are held to report the progress of the sprint on simple charts. At the end of each sprint cycle, a review meeting is held where work-completed and work-planned is reported. At this meeting a demo of the product will also be presented. These meetings are designed to provide feedback to each team member and to give a chance to preview demands for the upcoming sprint. Figure 1(A) below summarizes the scrum process.

3.   CUSTOMIZED SCRUM

When the feasibility of applying the original scrum method to projects at a local e-commerce company was examined, the following problems were found; first, the project team members who were accustomed to the waterfall model were confused about scrum team members and the related R&Rs. Second, the existing scrum process is aimed at producing products that could be released within the shortest period of time. It was thus difficult to maintain the overall view of the project when a project had a long project period. In the original scrum, all scrum team members shared focused only on one goal for each sprint cycle and develop product in the existing scrum. This resulted in idling of team members until the previous unit completed their tasks due to the interdependent nature of product development. Scrum team members make progress reports based on estimation points [11] during the sprint review phase. This process had a lot of room for improvements due to its inherent inefficiencies.

During the organization of the project team developers' R&Rs are clearly defined. Members are assigned similar roles they had from the previous development method. Team members are then made aware of specific tasks they are expected to carry out in each phase. This process will be introduced in detail in 3.1.

In the project planning phase of the customized scrum process, we added the master sprint planning phase, which aims to plan the milestone for the entire project before progressing into individual sprints. During individual sprints, goals are set for that particular sprint in an attempt to minimize stand-by time of each unit in the team. The sprint planning phase will be covered in 3.2. Figure 1(B) below shows the customized scrum process.

Finally, in the sprint review phase, the method in which progress is reported was changed. The number of fully developed web pages was used as an indicator of the projects progress, rather than the estimated scoring method, given
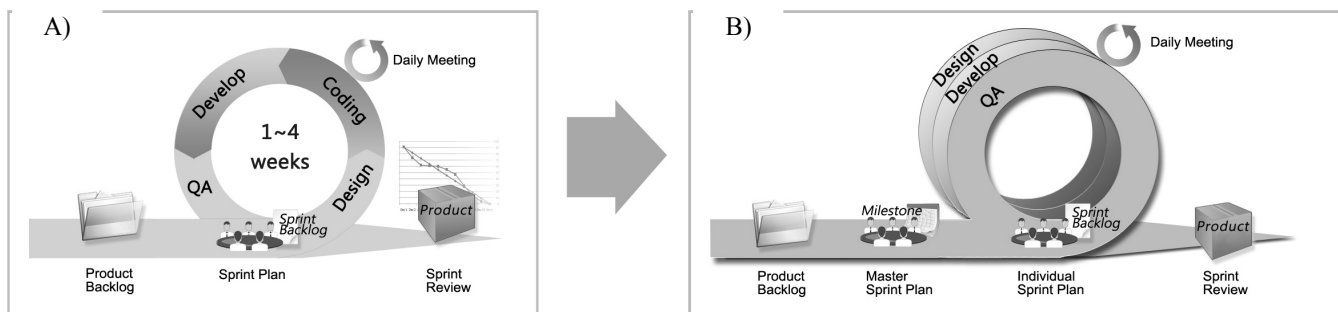

Figure 1. Original Scrum Process vs. Customized Scrum Process

that web pages were similarly complex and yielded similar estimated scores. This method is further discussed in 3.3.

3. 1 Scrum Team Setup & Role Definition by Stage

As with many types of teams and organizations, a scrum team is composed of multiple units with different roles comprising the whole. A typical scrum team is composed of three essential components or roles: a product owner, a scrum master, and other individual team members. If a company is to effectively utilize the scrum method for the first time, they must adequately understand R&Rs of these three roles and ensure that they execute their jobs properly.

The product owner is the person who has the chief responsibility of the products. They represent the client and serves as a liaison to the development team. They are responsible for prioritizing product functions and are able to re-prioritize these functions as the need arises, into the product backlogs to maximize return-on-investment (R.O.I.). The planning department handles this role since it used to take care of planning by collecting feedback from business departments. The scrum master helps the team familiarize itself with and execute the scrum process effectively. They are mainly responsible for resolving and eliminating any complications that arise during each sprint that hinder the development of the project. They ensure that the tasks listed in each sprint are completed on time. The project manager from the previously used method is assigned the role of scrum master. The final roles are just regular team members. They are responsible for the actual development and delivery of the product and may include designers, developers, coders, and quality assurance (QA) people. Refer to Table 1 for the R&Rs of a scrum team by each process.

TABLE 1. SCRUM TEAM R&R BY PROCESS

	Product backlog Review	Master Sprint Planning	Individual Sprint Planning	Daily meeting	Sprint review
Goal	Review and share product backlog	-Measure estimated scoring for product backlog -Determine the number of sprint -Set the milestone	-Set the goal for individual sprint planning -Produce Work progress table	-Share daily progress and updates -Check & resolve issues	-Demonstrate Sprint deliverables -Update the progress
Scrum Master	Review product backlog	-Hold meetings -Review entire product backlog -Determine the total number of Sprint -Review the milestone	-Hold meeting -Review work progress	-Manager progress -Manage issues -Manage schedules	-Update progress -Maintain issues -Maintain schedule
Product Owner	Delivery product backlog	-Measure estimated scoring for entire product backlog -Set goals for the entire Sprint	-Select backlog to execute during individual Sprint -Review work progress chart	Support issue resolution	-Check product -Check requirements that are missing in implementation
Scrum Team (Designer/Coder/ Developer/QA)	Review product backlog	-Measure estimated scoring for entire product backlog -Set the goals for entire Sprint	-Set the goals for individual Sprint by each operation process -Produce work progress chart	-Completed tasks -Planning tasks -Issues	-Demonstrate deliverables

3. 2 Sprint Planning Stage

During the planning phase of the original scrum method, it had focused mainly on individual sprint cycles. It was therefore difficult to establish the overall milestones of the project. We customized the existing scrum process to have the master Sprint planning meeting for the planning of the overall roadmap and milestones and the individual Sprint planning meeting to draw up more specific plans for each Sprint. The master sprint planning stage was added to outline the overall goals of the project whereas individual sprint planning meetings were more focused and detailed.

3. 2.1 Master Sprint Planning

The master sprint planning stage is part of the customized scrum process. It was introduced as a way to plan the goal of the project as a whole. In this meeting, unit leaders, the scrum master, and the product owner estimate the time and number of sprints it will take to complete the entire project. This meeting does not require all scrum team members to attend. Leaders of each area participate in this master Sprint planning meeting to set the goals of each sprint and determine the milestones of the entire project.

The product owner produces the product backlog, as is shown in Table 2, prior to the master sprint planning meeting. The product backlog is a list of requirements that a product owner lists based on priorities. The product backlog can always be modified accordingly due to new priorities and the needs of business environment that may arise. A product backlog is prepared for each web page. All backlogs are maintained by ID, allowing consistent access to all scrum team members. When a web page has many functions to be built, the backlog displays multiple data items that should be presented on that particular page, indicating their depths as well. Notice windows, such as store/delete/modify, etc., that pop up after entries are not specified. Instead, the number of pages, including pop-ups that should be generated in the "Page" column, is indicated on Table 2 below.

Leaders of each unit check product backlogs and the make estimation points for the concerned backlog, based on their experiences. This is called 'Estimated scoring'. The estimated scoring does not exceed. 40 hours. When the estimation exceeds the maximum 40 hours, it is broken down into smaller tasks. When this estimation is complete, unit leaders check if the scope is feasible within the given timeline. Because the product delivery date is fixed, there

may be tasks that cannot be completed within the timeline. Such tasks are put aside and excluded from the backlogs list. When the master sprint planning meeting is completed, leaders share the output to scrum team members such as the frequency and the number of sprints and expected goals of each sprint.

TABLE 2. PRODUCT BACKLOGS EXAMPLE

ID	1 Depth	2 Depth	3 Depth	SBD Page	Page	In-charge	Sprint	Effort	Done
FO1_1	Home	Recommend Items		Shopping main 8~9 p	1	Yu-mi Kim	1	24	o
FO1_2		New Items				Yu-mi Kim	1	8	o
FO1_3		Global Shopping	Best Items			Martin Kim	1	16	o
FO2_1	Category	Main categories		Shopping main 10p	1	Yu-mi Kim	2	16	x
FO2_2		Sub categories	Show Lists	Shopping main 11p	1	Martin Kim	2	8	o
FO2_3			Show thumbnails	Shopping main 12p	1	Yu-mi Kim	3	8	
BO1_1	Back Office	Managing Items		Administrator 3 p	2	Martin Kim	3	24	

3. 2.2 Individual Sprint Planning

Individual sprint planning meetings are held at the beginning of each sprint cycle. In this meeting, the scrum team selects the product backlog to develop for the current sprint and assigns the selected backlog into work units. In the original scrum method, each team member focused only on one goal for that sprint period. This has the potential to cause delays and leave some team members idle when there are differences in completion time between the units, resulting in a kind of "bottleneck effect". The existing scrum process was improved by setting up different sprint goals for each unit. After a scrum team selects the sprint backlog to develop in the current Sprint, they produce the progress dashboard and the task card as shown in Figure 2.

A task card is a list of tasks required to complete the backlog. One task card has a maximum duration of 24 hours. The team member responsible for the backlog produces the task card. When a task card is produced, it is posted next to the expected start date on the dashboard. They are written by team members and contain two important pieces of information, the ID of the product backlog, and the expected date of completion. The product ID allows accurate monitoring of the products progress amongst all the team members. The expected date of completion allows other members to accurately plan their schedule to minimize stand-by time. The expected start date is entered on the dashboard for the comprehensive view of the sequential relationship among units. For example, a coder can start coding when design is complete. We thus cannot estimate the start date of coding without knowing the expected end date of design. After the sprint planning meeting, the leaders of each unit updates the "in-charge" column of the product backlog (Table 2), based on the completed progress dashboard.

In order to reduce the stand-by time between units, persons responsible for each unit carry out each individual Sprint in the following ways. A designer designs the sprint backlog, which is selected for the individual Sprint and hands over the product to a coder as soon as the design is complete during the sprint period. Developers then modify the defects from the product that QA points out in the first one or two days of the individual sprint. Developers then move onto the development of the selected Sprint backlog for the current sprint plan. QA tests the product that developers completed in the previous sprint and reports the results of testing while producing a test case for the product that was developed in the current sprint. The sprint goal of QA becomes in sync with the goals of the past development Sprint. This is how reduction of stand-by time between units can be accomplished.
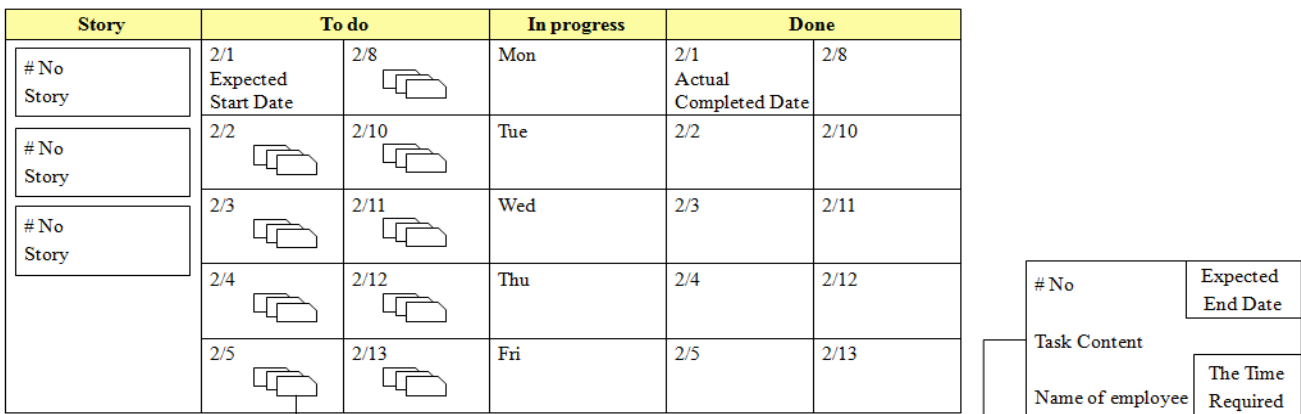
Story	To do		In progress	Done	
# No Story	2/1 Expected Start Date	2/8	Mon	2/1 Actual Completed Date	2/8
# No Story	2/2	2/10	Tue	2/2	2/10
# No Story	2/3	2/11	Wed	2/3	2/11
	2/4	2/12	Thu	2/4	2/12
	2/5	2/13	Fri	2/5	2/13

# No	Expected End Date
Task Content	
Name of employee	The Time Required

Figure 2. Progress Dashboard & Task Card

3. 3 Sprint Review Stage

At the end of each sprint cycle, a sprint review meeting is held. At this meeting, the scrum master reminds team members of the goals for each unit within the current Sprint plan and each member demonstrates their output to the team. They show how much of the product they have completed up to that point. The product owner uses this demonstration to determine whether individual tasks of a sprint have been completed. This is marked in the "Done" column of the product backlog with an "O". If there were any items that were not completed, not including simple bugs, the items are then put into the next sprint cycle.

When the review meeting is completed, the entire project progress is updated. In the original scrum method, project progression was reported on a burn down chart based on estimation points. This was inconvenient because every time an item from a previous sprint cycle is not completed and put into the next cycle, estimation-points change, requiring an update to the burn down chart based on the changed estimation points.

In the customized scrum, progress is monitored based on the number of web pages completed. This is a better approach to monitoring for e-commerce companies because the complexity of most produced web pages are similar. They therefore would have similar estimation point values in the original scrum method. Figure 3 shows the modified way of progress monitoring computed based on web pages completed. This method is much easier as we only have to add the number of fully developed pages out of the total pages to calculate the actual progress.

# of Sprint	Total # of pages to develop	# of Fully Developed Pages	Actual Completion Rate	Expected Completion Rate
1	1	1	20%	20%
2	2	1	40%	60%
3	2			100%

$$\text{Actual progress rate} = \frac{\sum \text{number of completed pages}}{\sum \text{total number of pages to build}} * 100$$

$$\text{Expected progress rate} = \frac{\dfrac{\sum \text{total number of pages to build}}{\text{Sprint count}} * \text{current Sprint}}{\sum \text{total number of pages to build}} * 100$$

Figure 3. Improved Progress Monitoring Method

4. APPLICATION EVALUATION

4.1 Quantitative Evaluation

Table 3 illustrates that outsourced development projects have higher defective ratios of normal test cases, compared with internal development projects and that the defective ratios are significantly improved when we apply the customized scrum process. Notably, the fatal defective ratio was decreased to the level of in-house development projects. Defective ratios could be significantly lowered since the product manager communicated with scrum team members to comprehensively understand the relatively ambiguous requirements via daily scrum meetings during the course of the project. We could reduce the impact of project risk factors by sharing all changes or issues among team members via daily scrum meetings.

According to the statistics of 137 projects a local e-commerce company conducted in 2009, the average stand-by time from planning to design task is 21.8 days and that for development is 12.9 days for large-scale projects. Table 3 also shows that the stand-by time was significantly reduced when the customized scrum process was implemented. Especially, a design task could start 18.5 days earlier than the planned end date and the development task could start 11.5 days earlier than planned. This means that resource utilization could become more efficient utilized at the early phases of the project.

TABLE 3. COMPARISON OF DEFECTIVE RATIO & IDLE TIME BETWEEN EACH UNIT

Project	# of Test Cases	# of Defects	# of Normal Cases	# of Ab. Cases	Defect Rates	Critical	Major	Minor	Design	Coding/ Flash	Develop ment	QA
A (Outsourced)	1951	605	33.4%	3.3%	31.0%	4.6%	31.9%	64.5%	0	-8	-15	-30
B (Outsourced)	764	474	57.2%	94.1%	62.0%	8.4%	46.4%	45.1%	3	-10	7	-30
C (Outsourced)	331	122	26.1%	100%	36.9%	13.1%	58.2%	28.7%	38	-6	38	3
I (Scrum)	278	79	12.4%	70.1%	28.4%	0.0%	68.4%	31.6%	-15	-3	-12	-30
II (Scrum)	1232	259	18.8%	64.4%	22.1%	1.9%	55.2%	42.9%	-22	-7	-11	-56
1 (Internal)	375	100	22.2%	37.6%	27.0%	1.0%	65.0%	34.0%	26	10	21	2
2 (Internal)	122	8	5.3%	25.0%	6.6%	0.0%	75.0%	25.0%	-15	-2	-1	-1

4.2 Qualitative Evaluation

Results from a survey taken by 18 members after using the customized scrum showed that 17% of respondents were very satisfied, 66% satisfied, and 17% nothing special. There were no unsatisfied respondents in this survey. Reasons for satisfaction with scrum included easy understanding of progress reports and a clear process. Any problems or issues that arose during development were also dealt with and resolved quickly at daily meetings, thus efficiently reducing the risk management. 83% of those surveyed found the daily scrum meeting highly efficient or efficient and only 17% found the daily meeting redundant. When asked if they

would apply this method to a future project, 78% agreed. The biggest reason why those surveyed were in favor of the scrum process was mainly successful on-time delivery.
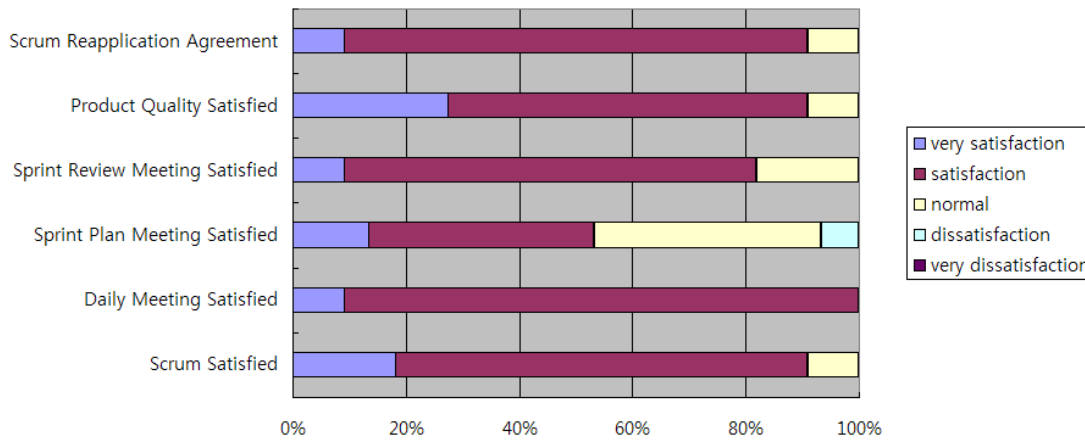

Figure 4. Scrum Surveys

5. CONCLUSION

This study proposes a customized scrum methodology for e-commerce companies. Scrum roles are assigned based on their similarity to the waterfall method. By upgrading to the sprint dashboards, we can efficiently plan the organic process for each task. The sprint review phase presented a method to maintain the progress rate based on the output of the development, reducing the complexities in the burn down chart of the estimated values of all pages.

As a result, we produced two outsourced projects with the customized scrum methodology, successfully decreasing the defect rate to rates near that of internal projects' defect rate. This method also uses resources more efficiently, reduces waiting time and enabled on-time delivery of the product. Surveys conducted also suggest more than an 80% satisfaction rate and respondents thought this methodology was helpful for scheduling and quality. The suitability of the customized scrum method was therefore verified for outsourced projects in the e-commerce industry.

Due to unforeseen circumstances, this study was not able to examine the impact the scrum methodology has on productivity. Further study is needed to verify how productivity is affected by the scrum methodology and how it can be efficiently applied when the work space of the outsourced development company is in a different location.

REFERENCES

[1] I. Sommerville, "Software Engineering(8/E)", Addison-Wesley, 2007.

[2] K. Schwaber, "Agile Software Development with Scrum", 2002

[3] The Cranky Product Manager Polls & Surveys, "Software Development Methodologies (Agile vs Waterfall)", Oct. 2008.

[4] Version One, 3rd Annual Survey: 2008 "The State of Agile Development", 2008.

[5] xpert.org, http://xper.org/wiki/xp, Website

[6] C. Schwaber and R. Fichera, Forrester Research, "Corporate IT Leads The Second Wave of Agile Adoption", Nov. 2005.

[7] G. Benefield, "Rolling out Agile at a large Enterprise", The Scrum Paper:Nuts,Bolts, and Origins of an Agile Process, 2007

[8] Forrester Research, "Enterprise Agile Adoption in 2007", Feb. 2008.

[9] S. Ambler, "Survey Says: Agile Works in Practice Agile Software development methods and techniques are gaining traction", 2006, www.ddj.com

[10] H. Kinberg, "Scrum and XP from the Trenches How we do Scrum", 2007

[11] M. Cohn, "Agile Estimating and Planning", 2006