International Journal of Evaluation and Research in Education (IJERE)
Vol.1, No.1, July 2012, pp. 25~32
ISSN: 2252-8822

Hybrid Methodology Involving Scrum and Waterfall Model towards the Software Project Development in Academic Knowledge Centers

Sonali Pathak, Pallavi Saxena
Computer Science Department, Lovely Professional University, Phagwara, Punjab, India
Email: pritipathak.pathak@gmail.com, pallavisaxena21@yahoo.co.in

Article Info

Article history:

Received May 19, 2012
Revised Jul 4, 2012
Accepted Jul 15, 2012

Keyword:

Academic knowledge
Effective communication
Presentation skills
Software project development
Waterfall model

ABSTRACT

Software project development in academia has the core objective to provide the student with an effective space to work in a team, interact with users, develop prototype, develop documentation, and improve presentation skills. An academic survey was done and it was found that the major challenge being faced by the institutes is the slack in controlling and tracking the project development activities. This results in poor quality software product which is in no way comparable with the industrial projects. Also, the ineffective use of tools and technologies are of great concern. The lack of proper and effective communication among the team members as well as with their project supervisor is also a major problem. Factors like, lack of enthusiasm, commitment, and financial support invite unwarranted risks. Such unstructured and uncontrolled way of working often leads to unsuccessful projects. All these factors show that academia is lacking more on the managerial issues rather than technological issues.

Corresponding Author:

Sonali Pathak,
Computer Science Department,
Lovely Professional University,
Phagwara, Punjab, India.
Email: pritipathak.pathak@gmail.com

1. INTRODUCTION

As the focus is on the transition from traditional Waterfall model to Scrum, several challenges are highlighted. The differences between both the models were studied by taking notes from several research papers, case studies, articles, and blogs. Also, the support from several industry professionals by social networking sites like LinkedIn, orkut, Yahoo Groups, and Google Groups helped in learning the Agile and Scrum concepts. Many software specialists feel that Traditional model is more comfortable because of its step by step approach toward development life cycle. On the other hand, Agile based developments are also increasing. Adaptation of Agile based methods has a number of reasons including requirements change, tight time deadlines, and fear of over budgeting. An *Agile based methodology* named *Scrum* has recently become the favorite choice of organizations for their software development. People at software organizations and academia find it really hard to switch from conventional Traditional methods to Scrum because of the diverse change approaches of both the methodologies.

Development teams face enormous difficulties in coping with change in customer demands. Besides requirements change, the adoption of new technologies also hurdles when it comes to traditional ways of developments. Higher level of competition in the software market has increased the desire for development models that can adopt and adjust with changes easily. *Waterfall model* lack this feature due to its subtle nature regarding requirements freezing during the whole development process. Keeping the changing nature

of market, organizations have started to adapt to the new dynamic, changing, and evolving nature of development methodologies commonly known as *Agile based development* (Martin, 2001), (Schwaber, 2004).

Agile does not represent any specific technique rather it's the name of a concept of development software systems. As name Agile suggest, moving quick and lightly, is the overall idea. The emphasis in this concept is to cope with changing market demands and requirements and tackle them with efficiency (Haugen, 2006). *Scrum* is an *Agile based development* technique. The term is taken from rugby which describes *'to bring back an out of play into the game'*. This first mention of Scrum can be traced to 1986 by Takeuchi and Nonaka where they presented a fast, adaptive, and self-organizing software development process. Later Schwaber and Beedle worked on it and published their combined work in 2002 over Scrum (Martin, 2001).

The basic theme of this methodology is that system development relies upon many variables. These variables can be environmental or technical such as unstable requirements, time deadlines, resources, and developing technologies. In spite of all precautions, these variables change. Scrum is aimed to provide such flexibility in development process that an efficient system is delivered to the customers on time (Martin, 2001), (Schwaber, 2004).

2. SCRUM AND ITS ROLES

Scrum mainly defines five roles as part of software development team, namely, Scrum Master, Product Owner, Scrum Team, Customer, and Management.

Scrum Master

Scrum Master is responsible for ensuring that the project is carried through according to the practices, values, and rules of Scrum and that it progresses as planned. Scrum Master interacts with the project team as well as with the customer and the management during the project. He is also responsible for ensuring that any impediments are removed and changed in the process to keep the team working as productively as possible (Warsta *et al*, 2002).

Scrum Team

Scrum Team is the project team that has the authority to decide on the necessary actions and to organize itself in order to achieve the goals of each Sprint. The scrum team is involved, for example, in effort estimation, creating the Sprint Backlog, reviewing the Product Backlog list and suggesting impediments that need to be removed from the project (Warsta *et al*, 2002).

Product Owner

Product Owner is officially responsible for the project, managing, controlling, and making visible the Product Backlog list. He is selected by the Scrum Master, the customer, and the management. He makes the final decisions of the tasks related to product Backlog, participates in estimating the development effort for Backlog items, and turns the issues in the Backlog into features to be developed (Warsta *et al*, 2002).

Customer

Customer participates in the tasks related to Product Backlog items for the system being developed or enhanced (Warsta *et al*, 2002).

Management

Management is in charge of final decision making, along with the charters, standards, and conventions to be followed in the project. Management also participates in the setting of goals and requirements. For example, the management is involved in selecting the Product Owner, gauging the progress and reducing the Backlog with Scrum Master (Warsta *et al*, 2002).

Communication and Information

Major contribution of Scrum as a management process is towards the effective communication and collaboration. Scrum follows certain practices in each and every phase of development. Some of the practices are discussed below.

Daily Scrum Meeting

A 15-minute Scrum Meeting is held every day. The Scrum Master asks the three questions, and all members of the team and interested parties take part and give feedback. The meeting should be held at the same place every time, so that people know where to go.

In this meeting, each team member was asked to answer three questions (Beedle *et al*, 1998)
What have you done since the last *Scrum Meeting*?
What has impeded your work?
What do you plan on doing between now and the next *Scrum Meeting*?

Sprint Planning Meeting

A meeting at the beginning of every sprint was held. Items from the Product Backlog are selected to be completed in the Sprint, based on the priorities set by the Product Owner.

Review Meeting

A Sprint is closed with a Sprint Review Meeting where the progress made in the last Sprint is demonstrated, the Sprint is reviewed, and adjustments are made to the project as necessary.

Sprint Retrospective Meeting

In the Scrum project management methodology, work is divided into 30 day Sprints. Following a Sprint, a Retrospective Meeting is held. Attendees are the Team and the Scrum Master. Each member of the team should be given an opportunity to answer:
What went well during the Sprint?
What improvements can be made on the next Sprint?

3. RESEARCH AND ELABORATION

Of late, newer technologies are evolving at a faster rate. The industries adapt to these changes swiftly as compared to academic environment. This becomes more evident in software development projects. Academia is facing a major challenge to cope up with such a faster rate of evolution. Students in tier 2 colleges and below (College and university rankings, 2011), follow *heavy weight processes* (Schwaber, 1995) for the software development activities, which are not an industry standard these days. Software project development in academia has the core objective to provide student with an effective space to work in a team, interact with users, develop prototypes, develop documentation, and improve presentation skills. Dapeng *et al* identified four issues in software project developed in academic environment:

- lack of planning
- lack of code management system
- lack of systematic testing, and
- absence of documents

These issues result in poor quality software product which is in no way comparable with the industrial projects. Also, the ineffective use of tools and technologies are of great concern. The lack of proper and effective communication among the team members as well as with their project supervisor is also a major problem. Factors like, lack of enthusiasm, commitment, and financial support invite unwarranted risks. Such unstructured and uncontrolled way of working often leads to unsuccessful projects. All these factors show that academia is lacking more on the managerial issues rather than technological issues.

This research proposes the adaptation of *Agile software development model* called *Scrum* (Schwaber, 2004) (Schwaber, 1995) for developing and managing software projects in academia. In order to learn how the Agile methodology can be useful to academic project is to apply the process model to an academic project. For this a project titled “Project Control Tracker (PCT) is planned to be developed. The hybrid of Scrum and Waterfall methodology is employed for the development of PCT. The case study is divided into three parts, questionnaires, interviews, and a feasibility study. The questionnaires and the interviews are used to get both a quantitative and a qualitative research. A feasibility study is also made to test whether it is possible to use Scrum as a project methodology at academic institutions and if impediments occur how to remove these impediments.

Interviews & Questionnaire

Data for identifying the problems faced by tier 2 institutes (College and university rankings, 2011) and students while developing software projects has been collected. Data collection was done primarily through open ended interviews and close ended questionnaire that were recorded. From all the institutes, MNNIT & GNIT, we interviewed professors and deans having experience both in software industry and academia.

All the data collected was then reviewed and qualitative data analysis (Haugen, 2006) was done. It was found that, *the major challenge being faced by the institutes is the slack in controlling and tracking the project development activities*. Based on the above finding, a tool was build that could control and track the

progress of software projects in the academia. Figure 1 shows the opening screen shot of *PCT*.

Project Setup

The development work was distributed between two institutes, MNNIT & GNIT located at Allahabad and Greater Naiad, respectively, where teams had their own Work spaces. MNNIT is a deemed university and GNIT is a tier 2 college. The purpose of choosing these colleges was to highlight the divergences in the software development activities of both types of colleges.

The development of *PCT* was started in August 2011 and completed in February 2012. The technical experience of the developers was limited to programming experience and academic knowledge gained from university courses. The project manager and the supervisor devoted substantial amount of time with the business and the project team during the development process.

Figure 1. Opening Screen shot of Project Control Tracker

Team Composition

The project team was composed of one development team at each institute having five developers and a supervisor. The developers were undergraduate (CS & IT) final year students and the supervisors were professors from the respective institutes. In addition, there was a project manager who is a graduate student of MNNIT, and an alumnus of GNIT who is in industry was assigned the role of a real world customer.

As both the teams were geographically separated, it was difficult for *Scrum Master* to keep attending daily meeting on a regular basis. This problem was resolved by maintaining and sharing the *log book* of daily report, using Google document. Backlog of requirements (Liu *et al*, 2008) can be maintained in many ways. *Story writing* is one kind of an activity that facilitates the conversion of raw requirement into a useful business requirement. These writings are a product of conversation involving several people (Seidel, 1998). As new thoughts came, they are added to the backlog.

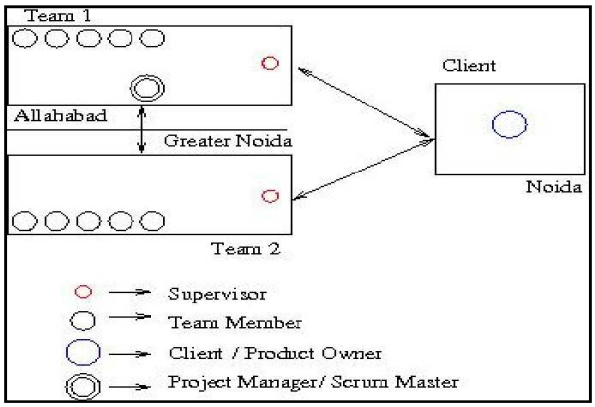

Figure 2. Team Structure

Process

Scrum is an *agile software development* methodology (Liu *et al*, 2008) which is significantly differment from the traditional Waterfall software development methodology. In order to keep pace with the latest industrial trends, practicing Waterfall model for software development is not enough. Transition from Waterfall model to *agile model* (Scrum & Agile, 2012) is Challenging and demanding, as it requires a very comprehensive research methodology and consumes too much of time. One solution for this is to develop a *Hybrid methodology* by combining *Scrum and Waterfall models*. We developed a model for such a *hybrid methodology.*

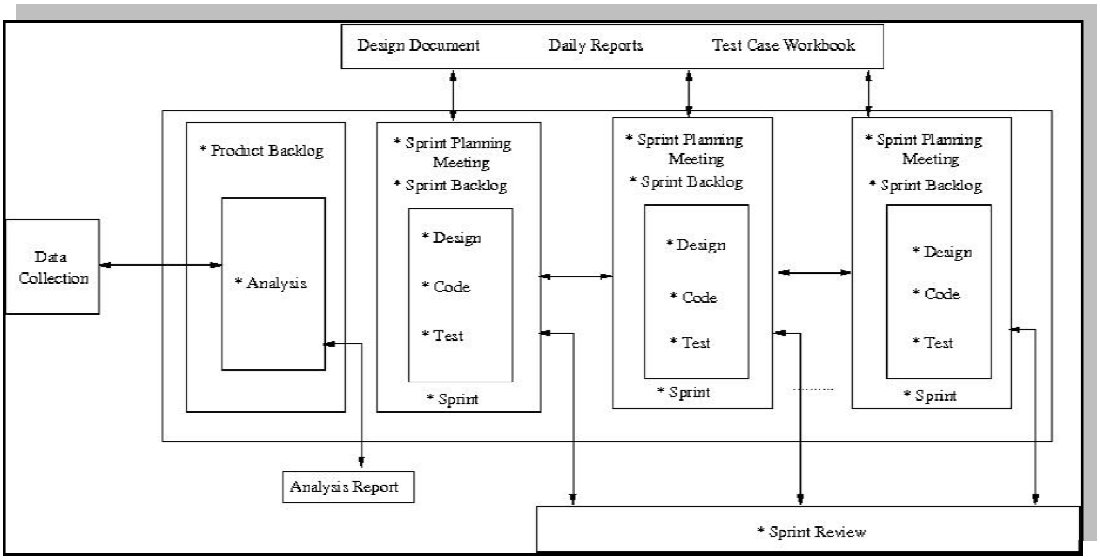

Figure 3. Model of Hybrid Methodology

As both the teams were geographically separated, it was difficult for *Scrum Master* to keep attending daily meeting on a regular basis. This problem was resolved by maintaining and sharing the *log book* of daily report, using Google document. Backlog of requirements (Liu *et al*, 2008) can be maintained in many ways. *Story writing* is one kind of an activity that facilitates the conversion of raw requirement into a useful business requirement. These writings are a product of conversation involving several people (Seidel, 1998). As new thoughts came, they are added to the backlog.Good story writing was the difficult activity, as none of us had prior experience of writing them. Breaking down these stories into simple tasks took lots of team effort. Since both the teams were geographically apart, there was a notable inconsistency that prevailed. For resolving this issue, we used BananaScrum. *BananaScrum* is an internet based powerful tool which includes features like, creation of product.

During *Sprint Planning Meeting* (Liu *et al*, 2008), few stories were picked up from the pool of *Product Backlog* and collected into the basket of *Sprint Backlog*. Every user story was broken down into several tasks and were estimated by team members through the practice of "Planning Poker" (Seidel, 1998), (Scrum & Agile, 2012). Each session lasted for 3 to 4 Hours. *Planning Poker* is a semi-structured estimation method (Scrum & Agile, 2012). In this practice, customer explains the user story and all the members are asked to discuss the story (Seidel, 1998), (Scrum & Agile, 2012).

During this project work, team members were given 5 cards, each having 0, 1, 2, 3 and 5 as point values. Project manager read the story and each one was asked to give points. Members showing maximum and minimum card values were asked to comment on their claims. The story was assigned with that point value, the team agreed to. Otherwise, the team was asked to reevaluate the story. This process goes on till the whole Sprint Backlog was estimated. When team finished with the point estimation, a 20 min. break was given for the refreshments. After the refreshments, prioritization of stories was done. This task was done solely by the project manager. The project manager prioritizes the stories as:

- Low
- Fair
- Medium
- High

After the prioritization of stories, the project manager gives a brief about the meeting results.

4. RESULTS AND FINDINGS

Sprint Backlog was distributed to both the teams by the project manager. The Sprint duration was same for both the teams, usually of 30 days. And it took 4 Sprints to complete the project. Table 1 shows the tabular description of Sprint and its duration. Each team decomposed the user stories into several tasks.Each team was then asked to work on the task list prepared by the other team. This activity was held to enrich the level of learning and for maintaining healthy communication among peers. Although, at industry level the team producing task list is responsible for coding those tasks. But, in this case study we focused on ample learning and not on the development of end product. During the Sprint, all team members used to focus on the allocated tasks only. Weekly *code review* sessions were held so as to make sure that the quality of code was maintained. Once the functional flow (code module) was completed, it was again inter-changed with the other team for *buddy testing and bug tracking*.

Table 4. Sprint Duration

Sprint	Start Date	End Date
Sprint 1	August 02, 2011	August 31, 2011
Review	September 01, 2011	September 04, 2011
Sprint 2	September 05, 2011	October 05, 2011
Review	October 06, 2011	October 10, 2011
Sprint 3	October 11, 2011	November 10, 2011
Review	November 11, 2011	November 15, 2011
Sprint 4	January 11, 2012	February 10, 2012
Review	February 11, 2012	February 15, 2012

Based on the testing and bug tracking reports, rework was done. Whenever it took longer than the scheduled period for the feature, it was dropped from the current Sprint and added to the next Sprint. Whenever the Sprint get over, the software product developed was shipped to the client for *system testing*.

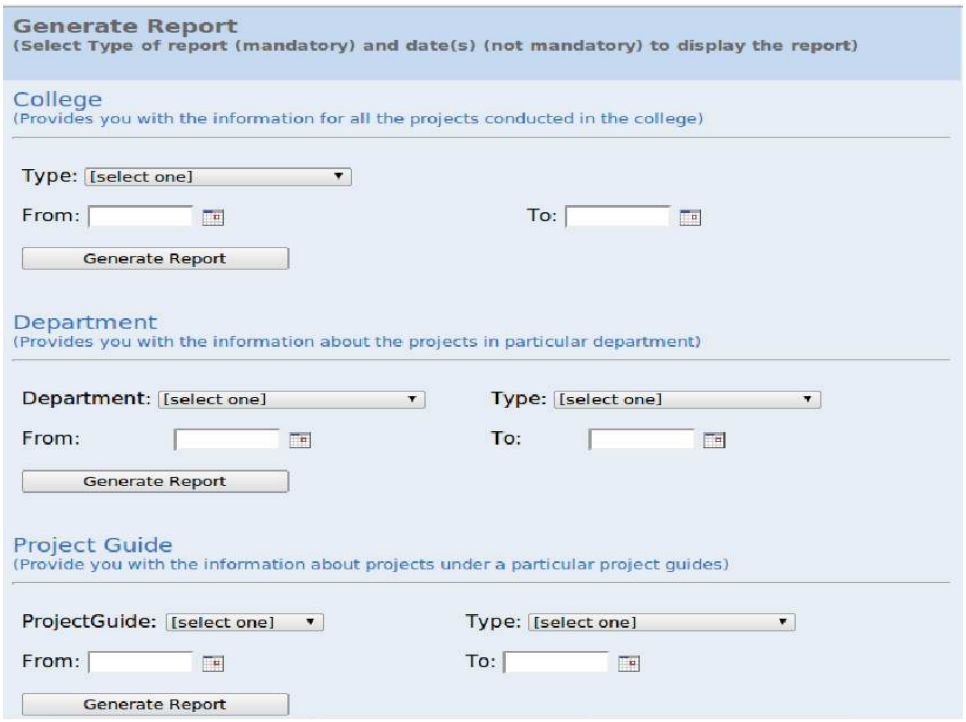

Figure 4. GUI for Report Generation by Dean

Sprint Reviews

Sprint Review was done after every Sprint (Dyb et al, 2008). The associated stakeholders, team members, client, supervisor, and the project manager met on a scheduled date and time. To overcome the geographical limitations we used webcam to conduct the meetings. The Project manager facilitated the walk through to let everyone know about the progress of the project. Any requirement changes and future plans were also discussed. Each Review Meeting lasted for around 2 -3 hrs.

Both the teams completed their work on schedule. The software product developed shows stability, robustness, and efficiency. Four kinds of users were identified:

1. Students
2. Project Guide
3. Dean/Director
4. Administrator

Project guide can register a project group, view the status reports & deliverables, and allocate grades to students. Students can post their daily reports or upload new deliverables.Institutes publish their academic calender for the session at the commencement of the session. In order to keep all the yearly projects up and running towards completion before the session ends, PCT provides a feature of publishing the calender by the dean of the institution. This schedule highlights the activities to be taken up during the project development. by. Dean can also generate reports for managing and track-ing the health of the projects undertaken in the institute by various departments and project guides. These reports were categorized as college reports, department reports, and project guide reports.

5. CONCLUSION

In this research, it has been shown that how *hybrid methodology* can do wonders. On one hand, *Scrum* suppresses the weaknesses of Waterfall model while on the other, elements of Waterfall model dissolves the resistance from the young developers who primarily study Waterfall model in their academics. For achieving the best results, it is the responsibility of the team to adhere to the *Scrum management policies*. The overall experience of using *Scrum Practices* has been very positive. Challenges faced and lessons learned paved a new way for students to experiment with the software methodologies.

The study concluded that organization should not adopt Scrum or any other Agile methodology in rush. Gradually implementation of Scrum Principles along with proper training and Scrum friendly corporate culture will increase success rate. After successful implementation of this study, it has been found that the academia is lacking on many grounds and needs major improvements. Following are the few suggestions:

1. Generally, the less powerful members accept and expect that power is distributed unequally. However, in Agile development, power is distributed equally among the team members.
2. People at academia believe that gender roles are clearly distinct; men are supposed to be assertive, tough, and focused on material success, whereas women are supposed to be more modest, tender, and concerned with the quality of life. However, in Agile development, gender roles overlap. Both men and women are supposed to be modest, tender, and concerned with the quality of life.
3. Project team members are threatened by the changing requirements. However, practicing Agile development lowers the possibility of being threatened by ambiguous or unknown situations.
4. Project Planning was found to be upfront. However, in Agile development, the project planning is continuous.
5. It is believed that the team needs to be told what to use. However, in Agile development, team has the last word.

This work is not such research which only becomes part of a bookshelf; it could serve as white paper in academia as it directly deals with an issue faced in many academic organizations concerning the adaptation of Scrum model. Future work may involve the application of hybrid methodology to some large and diverse project teams.

ACKNOWLEDGMENT

We thank the IJERE for giving us opportunity to get our research completed. We thank the Lovely professional University for its support throughout the research work. This project was supported by the University efficiently, so we thank everyone involved through out the work.

REFERENCES

Agile Model, *http://www.agilemodeling.com/*, Last accessed: 2011.08.12

Beedle, M., Devos, M., Sharon, Y., Schwaber, K., and Sutherland, J., *SCRUM: An extension pattern language for hyperproductive software devel-opment*, Washington University Technical Report TR WUCS-98-25, 1998.

College and university rankings, *http://en.wikipedia.org/wiki/College and university rankings*, Last accessed: 2011.08.12

Dyb T., and Dingsyr T., *Empirical Studies of Agile Software Development:A Systematic Review*, In Information and Software Technology, January, 2008.

Haugen N. C., *An Empirical Study of Using Planning Poker for User Story Estimation*, Proceedings of AGILE 2006 Conference (AGILE06), 2006 IEEE

Haugen N. C., *An Empirical Study of Using Planning Poker for User Story Estimation*, Proceedings of AGILE 2006 Conference (AGILE06), 2006 IEEE

Liu, D., Xu, S., Brockmeyer, M., *Investigation on Academic Research Soft-ware Development*, 2008 International Conference on Computer Science and Software Engineering, IEEE

Martin, R.C., *Agile Processes*, Prentice Hall, 2001

Mountain Goat Software, *http://www.planningpoker.com/*, Last accessed: 2009.08.10

North D., *Whats in a story?*, http://dannorth.net/whats-in-a-story, February 11, 2007, Last accessed: 2011.10.12,

Schwaber, K., *Agile Project Management with Scrum, Microsoft Press 2004*

Schwaber, K., *SCRUM Development Process*, OOPSLA'95

Scrum & Agile, *http://www.cprime.com/about/scrum faq.html*, Last accessed: 2012.01.14

Seidel John V., *Qualitative Data Analysis*, Qualis Research, 1998

Warsta, J., Abrahamsson, P., Salo, O., & Ronkainen, J., *Agile Software De-velopment Methods*, VIT Publications 478, 2002.