Implementing and Evaluating Scrum in Computer Science Senior Projects

Maral Kargarmoakhar
Computer Science Department
Florida International University
11200 SW 8th St
Miami FL 33199
mkarg002@fiu.edu

Mohsen Taheri
Computer Science Department
Florida International University
11200 SW 8th St
Miami FL 33199
mtahe006@fiu.edu

S. Masoud Sadjadi
Computer Science Department
Florida International University
11200 SW 8th St
Miami FL 33199
sadjadi@cs.fiu.edu

***Abstract* - This empirical study examines the adoption of agile software development, and the role of Scrum in computer science senior projects at Florida International University. This paper describes the senior projects and Scrum implementation. It highlights the advantages of incremental and iterative software development and discusses how Scrum can improve the productivity of software teams. In addition, it illustrates the other benefits such as engagement, transparency, frequent delivery and flexibility to change. Finally, it evaluates the outcomes by tracking the velocity estimations of each Scrum team. It demonstrates how Scrum and our tools can facilitate the software product development and the transition of our projects throughout semesters.**

I. INTRODUCTION

Agile software development (Agile) has been a trending topic in the software engineering during the past decade [1, 2]. Learning and implementing agile software development at school is challenging. Students not only need to work as a team but also must commit to the principles of Agile. The proposed solution in this paper describes the process of selecting projects, assignments, and software development management. Our solution is useful for software instructors at high school and university levels especially for those who assign semester projects to students.

In this paper, we show how we adopted Scrum on a large scale for several teams during two semesters at Florida International University (FIU) that complements the previous research. Our computer science (CS) students at the beginning of the semester join their favorite projects. They learn and practice Scrum in a systematic way, and they work with some best practice Scrum tools that we either adopted or developed ourselves. Benefits of adopting Scrum methodology for bachelor CS senior students are evaluated in this study.

II. SCRUM

Agile software development is an umbrella term for several iterative and incremental software development methods that follow agile manifesto [3]. Agile manifesto values individuals and interactions over processes and tools. Agile concentrates on working software, customer collaboration and rapid response to change [4, 5].

Scrum is a lightweight framework that emphasizes teamwork. It provides an iterative progress toward a well-defined goal [6]. A scrum team includes developers, a scrum master, and a product owner. A product owner provides a prioritized wish list of user stories called a product backlog. The team has a certain amount of time called sprint, which is usually two to four weeks. During a sprint planning meeting, the team selects several user stories from the backlog and then decides how to implement those user stories. Each team has a Scrum master who keeps the team focused on its goal and facilitates the meetings.

III. EXPERIMENT

Our senior projects are mostly focused on the software application development. With our wide advertisement at school and in Miami, FIU faculty members and their industrial partners become aware of this opportunity and submit their projects in our website (vip.fiu.edu). They need to propose their project at least one month before the semester. We evaluate the projects carefully and approve the qualified ones.

At the beginning of the semester, we introduce all projects to students. We provide them with a link to the last semester's deliverables including the main Github repository of the projects (https://github.com/FIU-SCIS-Senior-Projects). This repository consists of introductory videos, PowerPoint presentations, and all the code and documentation. We use one of our own tools (http://spws.cis.fiu.edu) to match students with the projects automatically based on students' skills and the skills required for each project. The course instructor is in touch with students and product owners to make sure students get assigned

DOI reference number: 10.18293/SEKE2017-183

to the right projects. Then we form the teams with three to seven members

To make a bridge between semesters, and a smooth transition, students need to make several videos for the next semester students. These videos not only are the introduction to the project, but also give an overview all of the complete user stories, a wish list of future features and an installation guideline. Students upload their videos to YouTube senior project channel and make a playlist of them in the FIU CIS senior project website (http://seniorproject.cis.fiu.edu).

In addition, we use many other useful tools such as Git and Github for version control, Skype and GoToMeeting for online meetings, and Slack for on-demand team communication. In our study, during Spring and Summer 2016, we tracked students' progress. We taught them how to assign story points to each user story based on working hours and the complexity of each user story. In the next section, we introduce story points, velocity and the evaluation process.

I. EVALUATION

For our experiment, we asked students to reflect their sprint planning in Google drive and Mingle [7] (https://fiu-scis-seniorproject.mingle.thoughtworks.com). Thus, we had access to all the information about implemented user stories. In the Spring semester, we had 15 teams and 45 students. The number of students in each team varied from two to six depending on the complexity of the project. During the first sprint we worked with each team one by one and made sure that they used Scrum correctly. We provided the students with several guidelines, and tutorials which allowed them to practice the story points and velocity estimation.

In each semester, students work five to seven sprints (Summer semesters are shorter). During the first week, they try to bring their projects up to speed. The last week of the semester is reserved for preparation of the showcase. All students need to work at least 20 to 25 hours a week depending on their semester. Students use a Google sheet called timecard to reflect their working hours every day. Scheduling the best time for meetings is not always easy. We provide students with a weekly schedule template, a guideline and a Google calendar link to sync their availability with the team.

In our experiment, story points and velocity are two of the main metrics for evaluation. A story point is used to measure the effort required to implement a user story. In other words, it is a number that identifies how difficult the user story is. The raw numbers we estimate are not important, rather the relative values matter. For instance, if we assign four points to user story A, and we assign eight points to user story B, this means that user story B is two times more difficult than user story A.

There are two ways to compare the user stories with the baseline user story. The first way is using the total required hours for completing a user story. The second one is using the level of complexity. In our experiment, we used the second one that means that story points have no relevance to actual hours. It makes it easy for our teams to estimate the points. However, teams identify their hours for each task of a user story in the Mingle as a reference of their actual daily work on each user story. The most common series of points are the Fibonacci series (1, 2, 3, 5, 8, 13, 21, etc.).

We use Scrum planning poker for story point estimation. We believe this game improves the visibility of each user story for all students. To start estimating the story points for each user story, each team picks a well-known user story as a baseline. It is important that the product owner defines all the tasks and acceptance criteria carefully. The team does not necessarily need previous experience on implementing that user story. However, it is essential that team members understand it. The team assigns a random point to this story. Then other user stories have to be sized based on it. A story point estimation must include all tasks involved in getting a user story completed.

Velocity is the total number of story points for all fully "Done" user stories during a single sprint. Velocity is calculated at the end of each sprint. It is important not to compare the velocity between teams. As we described, story point estimation depends on the baseline user story. Teams are likely to have different baseline user stories. Even if they have the same baseline user story, they may estimate different story points for that user story. For example, a team with a velocity of 100 is not better than a team with a velocity of 70.

We tracked all of the velocities for 15 senior project teams in the Spring semester, and 6 teams in the Summer semester. Figure 1 shows the user story points and the growth for the projects. The x-axis shows the number of sprints, and the y-axis shows the number of the total user story points. As some of the students were not familiar with Scrum at the beginning of the semester, we did not consider the first and second sprints in our charts. In addition, the final sprint of the semester was not considered in our charts since students wrapped up their projects and worked on the final deliverables. In Figure 1, the chart start from

number 1 instead of 3 just for better visualization of the process. In this chart, the growth of some of the projects is highlighted with a percentage representation. As the results show, all of the teams made a remarkable progress in implementing the user stories. For example, for the VIP website project, the total story points at the beginning of semester is 25. After four sprints, the total story points are 50 which means they has increased their productivity 100%. Figure 2 shows the total velocity of the teams. Teams have different velocities; but as we described before it does not mean that a team with the highest velocity is better than the other teams.

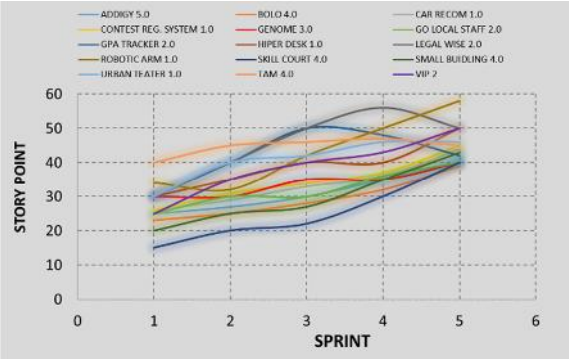

Figure 1. Teams' story points in Spring 2016

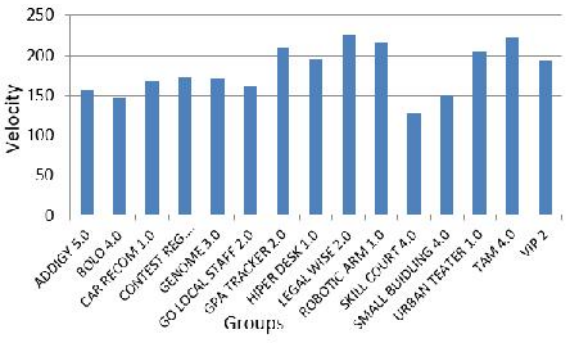

Figure 2. Teams' velocity in Spring 2016

II. DISCUSSION

Transition of the projects between semesters was a challenging task for us. Most of our projects are long-term and have become completed incrementally throughout years. After students submit their final deliverables, we have to transfer the materials to new students. Projects should have a clear and understandable programming structure. This allows the future groups to move forward and develop a new version of the project. If new students spend a lot of time to understand the code, their progress would not be as we initially expected. We use four main tools including Github, Mingle, Google Drive and Youtube to transfer the ownership of a project completely.

For our version of scrum adoption at school, we added another bi-weekly meeting to the semester schedule which is called "Feedback meeting". In this meeting, the instructor plays another role as a mentor. After each sprint, in an in person or remote meeting, the instructor and the team review the user stories together. The feedback meeting helps to keep students focused during the project and resolve some of the inconsistencies. We are still doing Scrum for our senior projects and we will report our new findings soon.

III. CONCLUSION

This study presents the Scrum adoption and evaluation for CS senior projects at FIU. We described the Agile, and Scrum in senior projects. We showed that Scrum helps students significantly in the progress of their project. The short iterations allow teams to deal with uncertainties. We evaluated the teams and showed the improvements. Finally, we discussed the transition of the projects between semesters.

ACKNOWLEGMENT

This research was partly supported by the Leona M. and Harry B. Helmsley Charitable Trust via the Georgia Tech's Vertically Integrated Projects (VIP) Program. The authors also thank Dr. Francisco Ortega, Ravi Kiran Agarthi and Shefali Nakum and the FIU graduate students attended Dr. Sadjadi's 2014- 2016 senior project, VIP, Advanced Software Engineering class for their technical assistance and insightful discussions during the class sessions. This material is based in part upon work supported by the National Science Foundation under Grant Numbers of I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, and MRI CNS-0959985.

REFRENCES

[1] Dingsøyr, Torgeir, et al. "A decade of agile methodologies: Towards explaining agile software development." (2012): 1213-1221.

[2] Serrador, Pedro, and Jeffrey K. Pinto. "Does Agile work?—A quantitative analysis of agile project success." International Journal of Project Management 33.5 (2015): 1040-1051.

[3] Manifesto, Agile. "Manifesto for Agile Software Development, 2001." URL: http://agilemanifesto.org (2010).

[4] Fowler, Martin, and Jim Highsmith. "The agile manifesto." Software Development 9.8 (2001): 28-35.

[5] Ashmore, Sondra, and Kristin Runyan. Introduction to agile methods. Addison-Wesley Professional, 2014.

[6] Rubin, Kenneth S. Essential Scrum: A practical guide to the most popular Agile process. Addison-Wesley, 2012.

[7] Taheri, Mohsen, and Seyed Masoud Sadjadi. "A Feature-Based Tool-Selection Classification for Agile Software Development." SEKE. 2015.