Akul Goyal
July 11, 2017

RAC 2-SCRUM+Microsoft

Scrum + Engineering Practices: Experiences of Three Microsoft Teams is a paper published in Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement by Laurie Williams, Gabe Brown, Adam Meltzer, and Nachiappan Nagappan. Williams is a lecturer at North Carolina State University while Brown, Meltzer and Nagappan are engineers at Microsoft

.

The paper starts off some commentary on current software practices in the industry and what the paper is trying to study. The paper then goes on with a basic summery of SCRUM and all the different points that make it up.

Different test cases done by companies before Microsoft are summarized in the third part of the paper. While it is shown that there is an overall improvement in an engineering team when one uses SCRUM, the paper stresses that this is just qualitative results vs the quantitative results recorded during its experiments. The authors also point out that the experiments conducted before are case studies while the research they have done is considered a formal lab.

The fourth part of the paper starts to discuss the different teams being used and how the experiment was carried. Of the four authors on the paper, two of them were in charge of interviewing and collecting the data and two of them were in charge of giving off data; i.e. being part of the experiment. There were three teams that were being experimented each with a different kind of setting. Along with using Sprint, the teams also utilized engineering practices created by Microsoft.

During the experiment, among the different teams one took shorten their sprint period and another only met three times a week for their daily scrum meeting. The authors went to then discuss the accuracy of planning poker and the

low estimation error faced by different teams. However, one drawback for planning poker was it required more upfront work for iteration having to plan the planning poker session. The paper then focused on continuous integration and the benefits of creating a different build with each addition of code. Unit test were the fourth topic to be discussed with the paper supporting automated unit testing citing the study in which automated testing lead to less errors. In order to prevent Flaccid Scrum, the teams utilized all or nothing for completion of the project, and this was the fifth topic discussed in the paper. Flaccid Scrum is the process in which teams only utilize the Scrums project management practices.

The team utilized source control, meaning that only certain members could write to the release branch. This made it easier for implementation of the code in to main branch and eventually the product. However, source control was harder with complicated use cases because they contained a lot of files some of which were owned by other engineers. The seventh step involved test converge. Some of the teams in the study had an 80 percent test coverage, any higher would have diminishing returns. However, test coverage told more about the unit test effectiveness rather than high quality code. The teams utilized reviews by senior engineers in the 8^{th} step, making it harder for error to escape into the field. The use of static analysis tools helped clean the code up and prevent small errors in the code. However, using this tool in the beginning was hard and took time. XML lets the code be self-documenting. These ten steps described how the engineering practices used with SCRUM by the Microsoft teams.

In the end, the authors concluded that SCRUM did help the team become a more effective unit based on several indicators: higher productvity and low defect rate. One of the teams had a 250 percent increase in and compared to some other test done, it was clear that there was an improvement using SCRUM.

The most important thing I took away from this article was the effectiveness of SCRUM. Obviously it was known that SCRUM helped teams become better but the results shown here made me realized how effective SCRUM truly is.