Agile: Lessons Learned & Best Practices

Overview

Team Members

1. Afshin Chaharmahalian
2. Lauren Dicristofaro
3. Jeremy Lerch

At the onset of the semester Professor Kokar indicated the class would be using Scrum as the development process for the project. Scrum employs more of an iterative and piecemeal nature to development as opposed to the waterfall development process. The focus is on producing working deliverable code at the end of each iteration in order to ensure valued work is being produced.

The beginning of the semester was rather hectic in being able to establish the process and tools to generate and track work products. Team leadership ultimately settled on JazzHub as the program management tool as it lends itself to tracking scrum work items.

Along the way team leadership has aimed to increase the experience and knowledge as it relates to scrum through several class lessons. The initial Scrum primer served as a session to provide a lexicon for students to communicate within the framework. Students were then shown how to create, update, and track their progress utilizing the program management tool JazzHub. Finally students were able to practice scrum estimation using the planning poker method. This estimation is widely used in industry and serves as a valuable experience in highlighting the importance of the team and its role in the development process. Going forward the leadership team will continue to identify opportunities like this in order to give students the knowledge and experience required for industry success.

Lessons Learned

1. Establish Leadership Team & Process before Development
2. Scrum Training
3. Early Estimation
4. Tool Specification
5. Co-Location

Best Practices

1. Establish Leadership Team & Process before Development: At the beginning of the semester most students are concerned with what work is going to be done to satisfy criteria for a successful project. This is certainly valid, however the process by which that work is going to get done is just as significant. A recommendation for next years project class would be to spend the initial weeks establishing the details of the development process. This includes Scrum, configuration management, team organization and roles, product owner priorities, project management and communication methods, evaluation/grading criteria, meeting management,an integration strategy, and establishing

a basic definition of done for all stories (i.e., design documents, unit tested, integrated into baseline). While team leadership focuses on this, students can focus on becoming familiar with the application and developing a list of improvements that will ultimately serve as the product backlog. This will provide an environment where the development goals and process flow are clear to everyone in the class.

2. Scrum Training: Scrum is still relatively new in industry, however it is quickly becoming the standard by which software projects are developed. It is incredibly important for students to have an understanding and even more importantly have experience in Scrum. There should be a clear focus on the importance of this process and the high level knowledge required to participate in it (Epics, Stories, Tasks, Estimation, etc.).

3. Early Estimation: One of the most valuable exercises in Scrum is Story Point/Ideal Day estimation. What it does is empower the team to identify how much work they will be able to complete in a given sprint. Unfortunately due to the time constraints this wasn't done until after the halfway point. A recommendation would be to do this prior to the development of any features. Not only does it allow for a successful sprint but it serves as a valid team building tool that can be advantageous at the beginning of the semester. It also will present the team with the opportunity to realize the importance of the burndown and velocity charts.

4. Tool Specification: Along with specifying process, the tools that help achieve a team's work are just as important. The class ultimately settled on JazzHub and GitHub and blackboard (as determined by class requirements) as tools to manage progress and deliverables. However SmartGit emerged as a powerful tool to interface with GitHub late in the semester. Although a number of different clients can be successfully used with git, we found that using one shared client was very helpful as a way to share knowledge about git to those who are less familiar with it. It would be wise to determine these tools earlier in the semester. Even a common tool from semester to semester would be valuable for project tracking. JazzHub is developed by IBM and is a subset of features for Rational Team Concert which is being utilized in industry. Specifying these tools as early as possible would provide a much smoother development environment for students to transition to at the beginning of the semester.

5. Co-Location: The increased communication among scrum team members is one of the frameworks given strengths. This is also a byproduct of the every day work environment of the team members. Scrum teams on a project are typically located next to each other on a day to day basis. This makes subject matter experts readily available and provides a more open workspace for a cohesive team. Due to this project being on campus and with an in class meeting of once a week it does not lend itself to being part of one these co-located, cohesive team environments. Many of the day to day interactions are lost in a traditional setting such as this and most of the productivity gains may not be realized. While it is not optimal many companies have tried Scrum on distributed teams with varying success. Colocation may be better addressed by having a more open and easily usable forum for discussion and intra-team communication. Piazza is a "free online gathering place" for open discussion that has been successfully used in other Northeastern classes (CS5700). It has a mobile version, so it is very convenient to use, and might provide a good forum for asking questions and addressing issues online.