Introduction

The goal of the project was to use Agile software engineering practices to implement a mobile application version of Chinese Checkers that could be deployed to and used on an Android phone. Success can be measured by the artifacts created that support project planning, analysis, design and testing as well as by the final working software product. The main purpose of the COSC 3F00 course was to learn software engineering practices and the software product of this project was actually secondary to the lessons that were learned. This project summary will focus on some key lessons discovered in the areas of communication, planning, analysis, design, development and testing.

Communication

Face to face communication is the most effective method of communication. The team worked hard to meet twice each week for a period of two hours and sometimes more. However, the first problem the team faced was the fact that there was only limited time that everyone could meet together as a group to work on the project. In order to keep information flowing between group meetings the team used several mechanisms. Email alone had drawbacks. Fortunately other social media solutions were available such as Facebook and GitHub. The team took advantage of both of these options. GitHub also had the added benefit of some lightweight project management capabilities such as project task creation, task assignment, bug tracking, discussions and milestones. It also had configuration management capabilities and was used for the team’s software version control. A combination of social media and face to face communication was used with great success and were equally essential in making the progress that was observed.

Planning

An Agile software development approach was chosen because of it’s flexibility and responsiveness to change. During the initial planning phase of the project it was determined that there were many unknowns and risks so the scope needed to be flexible and there needed to be focus on quick, iterative delivery and feedback. Agile was the natural choice.

An estimated work effort capacity was calculated based upon team members availability for the project effort. This was used as a guide when it came to making decisions about how much work was to be accomplished with each iteration. Based on this, decisions were made during each planning meeting about what could be done with the next iteration.

After each iteration a retrospective conversation took place where suggestions were raised and decisions were made about how to modify the process.

The project work was assigned and tracked using project tasks that were loaded to GitHub. Agile “Burndown” charts were created to visualize the progress of the work through the iteration.

The project effort consisted of four iterations of varying duration. This was done intentionally during the initial planning stage. The first iteration was four weeks and was the longest of the iterations. This was done to accommodate for the expected ramp up time needed to get the tools the process foundations in place. The next two iterations were three weeks in length. These iterations were intended to contain the bulk of the development. The final iteration was two weeks in length and was intended to be used to implement any final missing features and eliminate any outstanding defects.

As was previously mentioned during the initial planning stage of the project a work effort capacity was calculated and was to be 528 total hours. The expectation was the each person would commit 6 hours per week to the project. Multiplied by 8 people over the course of 11 weeks. However, the observations of the team over the past 11 weeks show that this was drastically underestimated. A more conservative estimate would be that the effort was at least twice the original prediction. Less time could have been spent but the results would have been a product with less functionality.

Analysis, Design, Development and Testing

The use of conventional artifacts such as UI mockups, data flow diagrams, class diagrams, use cases and test plans all helped to guide the development and testing of the product. Other Agile concepts such as test automation and continuous integration were used to augment the construction process.

The AndroidStudio tool used for development included a framework for test automation. This was leveraged to build out a suite of functional and unit tests that could be run at the click of a button or by a script.

The use of an online continuous integration service called travis (travis-ci.org) was used to automatically checkout the latest changes introduced into the GitHub source code repository, build the software executables and run the suite of automated tests whenever a developer on the team committed a change to the repository. If the travis process failed due to a compilation error or a failed test then an email would be sent to the individual that committed the last change which introduced the problem and it would need to be fixed immediately. This gave confidence to the team that there was always a working build in the repository.

As previously mentioned, AndroidStudio was the development tool of choice. It afforded the team with the ability to seamlessly develop on multiple platforms. Both Windows OS and Mac OS devices were used to develop and integrate this product.

For the testing activities, both manual and automated testing were done successfully on multiple Android devices (phones, tablets, emulators).

The combination of conventional and agile development tools and activities were essential in accomplishing the goals of the project.

Conclusion

In order to create reliable, quality software proven software engineering practices are required. Complexity increases exponentially when working in teams and success with teams requires special commitment to planning and communication in addition to the basics of analysis, design, development and testing. Teams with individuals of varying skillsets are necessary for large software development efforts and there are many lessons to be learned and applied from this experience in COSC 3F00.