

A Comparison Between How Agile Software Development is Practiced in a Corporate Environment Versus an Educational Setting

by

Edgar Brandon Tucker

An Honors Capstone

submitted in partial fulfillment of the requirements

for the Honors Diploma

to

The Honors College

of

The University of Alabama in Huntsville

December 6, 2018

Honors Capstone Director: Dr. Richard Coleman

Professor of Computer Science

Edgar Tucker 12-4-2018
Student Date

Richard Coleman 12-4-2018
Director Date

H.S. Rangarath 12-4-2018
Department Chair Date

Bill U 12/10/18
Honors College Dean Date



Honors College
Frank Franz Hall
+1 (256) 824-6450 (voice)
+1 (256) 824-7339 (fax)
honors@uah.edu

Honors Thesis Copyright Permission

This form must be signed by the student and submitted as a bound part of the thesis.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Edgar Brandon Tucker

Student Name (printed)

Edgar Tucker

Student Signature

12-4-2018

Date

Table of Contents

Abstract	2
Introduction	3
My Time At ADTRAN	4
My Senior Design Experience	6
Conclusion and Improvements	8

Abstract

For my Honors Project I was tasked by Dr. Coleman to help him improve the senior design course for the Computer Science department by using my work experience as a Co-Op at ADTRAN to point out differences and possible improvements that could be made to the course. In this paper, I look at first what Agile is and how software developers use it, my experience at ADTRAN using Agile daily to improve workflow and development team cohesion, how Agile was used in my Senior Design course and how we practiced it against the ideal of how Dr. Coleman wanted us to use it, and in the Conclusion I give some suggestions as to how I would change the course to more closely emulate my experience in the corporate environment at ADTRAN.

Introduction

Agile software development is a set of guidelines that companies and teams of developers can follow to improve their efficiency, the quality of their product, and to ensure that the end result is what the customer wants. There are many different types of Agile and oftentimes businesses will employ multiple agile methods within their company. The agile methods that I am most familiar with and will be referencing through this paper are Scrum, Kanban, Extreme Programming, Feature-Driven Development, and Lean. Some of these methods overlap in what their main goal is, but they all have something that makes them unique and are easily able to be used in parallel with the others. One of the strengths of Agile is the ability to pick and choose which methods work best in your specific situation. In this paper I will be looking at how I have used these principals in a corporate environment at ADTRAN as a Co-op and my personal experience in Senior Design at UAH.

My Time At ADTRAN

When I began working at ADTRAN I had no experience using Agile and wasn't all that familiar with it either. However, since Agile is such an integral part of how ADTRAN runs, I learned very quickly. Oftentimes co-ops learn Agile simply by being exposed to it so much during their first term, but I was lucky enough to go through a training session during my second term when the company was changing their agile method. This training lasted three days and was led by a representative from Scaled Agile Framework, or SAFe. Throughout this training we went over how the roles of the Product Owner and Scrum Master would be changing, but the largest change with the most push back was changing the composition of the individual teams from horizontal alignment to vertical alignment, meaning, rather than each team having a specialization, each team has one specialist for each particular area. For example, before this rework I was on a team that only worked on front end web development. After changing the teams around, I was one of the only two people on the team that had front end experience. This new formation is called feature teams because each team would be able to create an entire feature from top to bottom rather than having to pass the same feature from team to team in order to complete it. While this may look good on paper, in the short time that I was at ADTRAN after this was instantiated, I also saw a few problems. For some developers, the team that they joined didn't have much work in the area that they specialized in and so they had to work in an area that they were less comfortable. While this is likely only a short term issue, this left several developers unhappy with the work that they were being assigned.

As far as how Agile was practiced at ADTRAN, immediately after this training we went into planning for the next PI, or Product Increment. This planning lasted two days and consisted of creating stories for each team for the next four iterations, or eight weeks. To figure out how much work each team can accomplish during this time each team calculates their velocity based on previous iterations using story points. Story points are an estimation of how long a particular story will take to complete, with one point generally being one day. It can differ between teams but most story estimations were based on a modified fibonacci sequence, meaning a story could be 0.5, 1, 2, 3, 5, 8, 13, 20, or 40 points with no individual committing to more than 8 points per iteration. After planning was completed, the first iteration would begin. ADTRAN's iterations typically would start on a Wednesday and would run for two weeks when I was there. Everyday, typically in the morning, I would have a daily stand-up with my team. During these meetings each person would answer three questions: what did I do yesterday, what will I do today, and do I have any issues that are blocking me. The other major event that would happen each iteration would be team and system demos at the end. The two teams that I have been on did these slightly differently but the main point is to show other teams in your department or the upper management of your department what was accomplished during the previous iteration. While these were generally low stress, I felt that they were one of the leading motivators to get my work done.

My Senior Design Experience

At UAH my experience with Agile has been quite different. The only time I have gotten to practice this was for CS 499 which is Senior Design. By the time I got to this course I had been working at ADTRAN for a total of about 8 months and had a good grasp on the principles of Agile. Agile was taught slightly differently with it being a more classroom oriented approach but the principles were essentially the same. One difference though is that at ADTRAN we learned specifically what we were going to be using and how we were going to apply it to our company. In Senior Design we learned more generically what different forms of Agile exist and were largely left to ourselves on how we wanted to use them even though some forms were encouraged. Through this class we did several activities that lined up with Agile activities at ADTRAN. In the first couple of weeks we created a Software Development Plan and a rough backlog which is similar to my experience with planning at ADTRAN. There were also presentations at the end of each iteration similar to team demos, but the content of these presentations were a little different.

Speaking from my own experience, I would say that we did not follow Agile as strictly as my experience at ADTRAN and probably not as strictly as the instructor would have hoped, but there were also several inhibitors preventing this from happening. The first of these was the lack of co-located developers. This is something ADTRAN is starting to emphasize even more by moving teams all into the same room rather than offices. While my team and I did maintain contact at virtually all hours of the day, it isn't quite the same as being able to lean over and ask five other developers a

question and get right back to programming. I also realise that this is impossible with students whose schedules are more than likely only going to be compatible for an hour and twenty minutes twice a week. We were able to get together several times over weekends for several hours at a time but I don't believe we were ever able to get the entire team at any one of these meetings. Another unavoidable hindrance was the lack of low level management such as a permanent Product Owner and Scrum Master. At ADTRAN, the Product Owner is a full time position and as such generally does no programming. With larger teams, the Scrum Master is also practically a full time position and frequently doesn't get assigned stories. These two positions are vital to the coherence of the team, ensure that the proper work is being done, and act as decision makers if there is ever any discrepancies with how to progress forward.

Conclusion and Improvements

To conclude I would like to propose a few recommendations for Senior Design to make it mimic more closely my experience at ADTRAN. The first is to put as much planning as early as possible in the semester. My team and I often times felt bogged down by preparing for presentations in the middle of the semester to where it was taking time away from programming. It also felt awkward that we weren't done with our backlog until the end of the second sprint when I am accustomed to it being completed before the start of the first iteration. One potential solution to this would be to move back the start of the first sprint one or two weeks and use this time for planning. Another potential improvement would be to have team demos at the end of each iteration. Instead of having two days worth of presentations for documentation at the end of each sprint, this could potentially be shortened to one day with shorter, five minute presentations per team. The second day that is freed up could be used for each team to give a short demo to the instructor on what they have been able to accomplish during the iteration. This would put a slight amount of pressure on the teams to accomplish something each iteration as well as create more interaction between the instructor and each individual team. My final suggestion would be to encourage the use of story points and a Kanban board. Kanban boards were suggested to be used as a backlog, but I don't believe my group ever got around to actually making one and keeping up with it. While we did fine without one and without story points, I believe these would help keep track of stories and give a better visual representation of the amount of work that remains. While I believe that Senior Design is currently being

taught very well, I believe these small changes would make this course more similar to my time as a Co-op at ADTRAN.