THE AGILE SOFTWARE DEVELOPMENT PROCESS: A Study and Practical Application of Local and International Best Practices

Brian Bird

Computer Information Technology Department
Lane Community College

September 17, 2017

1. INTRODUCTION

The purpose of this sabbatical was to gain knowledge and experience in the latest "best practices" in the Agile software development process.

A. About Agile

"Agile" is the name given to a relatively new approach to software development. The Agile movement began to attract national attention in 2001 when a group of software developers published the *Manifesto for Agile Software Development*. This document, and the movement, initially, emphasized certain values and priorities more than particular practices or techniques. The values listed in the Agile Manifesto are:

- Individuals and interactions over Processes and tools
- Working software over Comprehensive documentation
- Customer collaboration over Contract negotiation
- Responding to change over Following a plan

While the Agile movement originally emphasized a philosophy more than a set of practices or techniques, Scrum, one particular approach to software project management, has become closely associated with Agile over the past ten years. Scrum is a team software development methodology that employs lightweight project management techniques, which allow the development team to continuously adjust to changes in requirements based on ongoing feedback from the customer.

B. About Myself

I worked as a software developer for over fifteen years before becoming an instructor at Lane Community College. My career started in the early-nineties and continued until 2009 when I started teaching at LCC. Although I had heard of Agile, the company where I worked, Axian Inc., had not at that time formally adopted an Agile approach to software development. Since then, Agile has gained even wider acceptance and most companies consider it to be the only way to do software development. As an instructor at Lane, I have spent considerable time reading literature on Agile development and incorporating what I have learned into the classes I teach, but until this sabbatical, hadn't had the opportunity to see how real companies are using Agile development methodologies.

2. ACTIVITIES AND LEARNINGS

- Studying current literature on Agile methodologies
- Visiting software development groups at local companies to do observation and interviews
- Attending a conference on software development methodologies
- Applying what I learn to contributing to an Open Source software project that uses an Agile approach

A. Survey of current literature

The following list consists of some of the most highly regarded books on Agile and Scrum.

i. Cohn, Mike. *Succeeding with Agile.* Addison-Wesley Professional, 2009

ii. Lacey, Mitch. *The Scrum Field Guide: Practical Advice for Your First Year*. Addison-Wesley, 2012

iii. Martin, Robert. *Agile Software Development, Principles, Patterns, and Practices*. Prentice Hall, 2002

iv. Rubin, Kenneth. *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Addison-Wesley, 2012

v. Sims, Chris and Johnson, Hillary. *Scrum: a Breathtakingly Brief and Agile Introduction*. Dymaxicon, 2014

vi. Shore, James. *The Art of Agile Development.* O'Reilly Media, 2007

vii. Schwaber, Ken. *Agile Project Management with Scrum*. Microsoft Press, 2004

B. Interviews and observation of local software development teams

This was the core of my sabbatical project. I visited seven local companies with the purpose of understanding how they are applying Agile software development principles

i. Below is a list of local companies where I have interviewed and/or observed software development teams. I have included notes from my interviews and observations

   1. ***IDX***
      100 East Broadway
      Eugene, OR 97401
      http://www.idxbroker.com

      a. *High-level plan with Gant chart*

      b. *Using Agile with Scrum*

         i. Start with a one or two week analysis and design phase

         ii. A backlog of user stories is built

            1. Features and issues mainly come from the customer support team

            2. Stories include acceptance criteria

        3. Estimated in hours, not story points

    iii. Sprints are one week long

        1. Using a Git workflow with developers working on branches and issuing pull requests—code reviews are done by at least two developers
        2. Developers write unit tests
        3. Using Jira to manage sprints and backlog and generate burn-down reports

    iv. Each developer has a Kanban board to track progress

c. *Doing CICD (Continuous Integration, Continuous Delivery*

d. *QA department does testing: regression, integration, acceptance and manual UI testing.*

2. ***Symantec Corp.***
Norton Engineering Group
555 International Way,
Springfield, OR 97477
http://www.symantec.com

    a. *Using Agile with Scrum*

        i. A backlog of user stories is built

        ii. Sprints are one week long

            1. Using a Git workflow with feature branches, pull requests and code review by at least one developer
            2. Developers write unit tests
            3. Moving toward using Kubernetes to test code after each pull request

    b. *Doing Continuous Integration with Jenkins, moving toward using Continuous Delivery*

    c. *Recently eliminated the QA department-- developers do their own testing*

        i. UI testing done with Selenium

        ii. Moving toward just doing unit tests in React

3. ***PacificSource***
   110 International Way
   Springfield, OR 97477
   http://www.pacificsource.com
   a. *This is a larger company. There are 4 teams of developers in the enterprise division, a few in another division for a total of about 45 developers.*
   b. *Using Agile with Scrum*
      i. Sprint planning and retrospective are done at the beginning of each sprint
         1. Requests for features and bug fixes come from the help desk
         2. Estimates are done in hours, not story points
      ii. Team members:
         1. Scrum master—manages backlog, leads sprint planning, leads daily scrum meetings
            a. *Using Urban Turtle to plan and manage sprints*
            b. *Generates burn-down charts*
         2. Product owner (usually the manager)
         3. Business Analyst
         4. Testers
   c. *Using VSTS for version control and workflow management*
      i. Using Kanban boards to track progress
      ii. Using Urban Turtle to manage sprints

4. ***Palo Alto Software, Inc.***
   44 West Broadway
   Suite 500
   Eugene, OR 97401
   http://www.paloalto.com
   a. *Using Agile with a "pretty standard Scrum work-flow"*
      i. A backlog of user stories is built
         1. Estimated in story points
      ii. Sprints
         1. Using a Git workflow with developers working on branches and issuing pull requests and getting code reviews
         2. Developers write unit tests
      iii. Feedback at the end of each sprint
         1. For new product development, comes from the VP, who is the product owner

2. For existing products, comes from customer service

b. *Integration and Testing*
   i. Doing Continuous Integration with Jenkins
   ii. Used to use Selenium for UI testing, now it's manual

5. ***IEQ Technology***
   220 5th Street, Suite 100
   Springfield, OR 97477
   https://ieqtechnology.com
   a. *Using Agile loosely*
      i. Project management
         1. Legacy projects use Jira for issue tracking
         2. New projects use a Kanban board for project tracking
         3. Issues and feature requests are entered by the manager
   b. *Using Git for version control, but no pull requests or code reviews*
   c. *Testing: developers enter and close their own bugs. Don't usually write unit tests.*

6. ***Mindbox***
   44 W 7th Ave.
   Eugene, OR 97401
   https://mindboxstudios.com
   a. *Using Agile with remote developers*
      i. Start by creating a roadmap and gathering requirements
         1. Use AhHa project planning tool
         2. Use Pivotal Tracker for Agile project management
      ii. Sprints are two to four weeks
         1. Daily code reviews for each merge or patch. Using Gerrit for code reviews
         2. Developers write unit tests
      iii. Test
         1. Deploy from Git to the test environment
         2. Do some behavioral testing

7. ***Lunar Logic***
   115 W. 8th #300
   Eugene, OR 97401
   https://lunarlogic.com
   a. *Using "Mountain Goat" Agile (Cohn, 2009)*
   b. *Roles*
      i. One scrum master for all teams
      ii. Product Owner for each team
         1. Advocate for the client
         2. One person may be product owner for ten to forty products
         3. Does a lot of communicating with the client and development team
   c. *Sprints are two weeks*
      i. Backlog of user stories
         1. Estimates are in story points
         2. Used to include tests with user stories, but now tests are written by QA based on the requirements so that two people have interpreted the same requirements
      ii. Planning meetings are about one and a half hours
      iii. Retrospect meetings are half an hour
      iv. Code reviews just for big projects
   d. *Practices*
      i. Version control is half SVN and half Git
      ii. Unit testing
      iii. Pair programming
   e. *Testing*
      i. Done by QA
      ii. No formalized acceptance testing
      iii. Using Post It to move code to the staging server

ii. Summary of findings

The way different companies implement Agile varies depending on the type of business they do and the personal preferences of management. This is appropriate since Agile is not a methodology, nor is it a set of practices, it is a philosophy and a mind-set. Scrum, on the other hand is a methodology, just one of many Agile methodologies.

Companies that do software development for other companies (as opposed to developing their own products) tend to have a less structured approach to Agile development—especially if they do a lot of small projects. Examples of this would be IEQ and Lunar Logic. On the other hand, companies that develop and sell their own products, like PacificSource and IDX have much more well developed Agil e development methodologies

Scrum is by far and away the most popular methodology and is often combined with practices from eXtreme programming like *pair programming* and *test driven development.* All the companies I visited claim to be using Agile Scrum. In the companies that apply Agile more loosely, Scrum means little more than having a daily Scrum-style stand-up meeting, while other companies apply it much more completely. While none of the companies that I visited were using Kanban as a methodology, many of them use Kanban boards as a way to track either project or developer progress.

The most widely used work-flow is the Git feature-branch workflow with developers making pull requests and getting code reviews. The most popular tool for Agile project management is Atlassian Jira as well as other Atlassian software tools.

C. Attend an Agile conference

I wasn't able to attend a conference in person, but I was able to watch videos of relevant sessions from Agile 2016, the premiere Agile conference of the year. One of the sessions that particularly caught my attention was Women in Agile, the Changing Face of Agile. In this session, two women who are in high-tech management positions explained how Agile was their entre to a high-tech career. They found that the concepts and practices of Agile resonated with them and they were able to develop knowledge and passion around this approach to project management. This gave them expertise and credibility that had otherwise proved to be elusive in a male dominated profession. This is relevant to use at LCC since we see very few women (around 10%) entering our programs in Computer Information Technology.

D. Participate in an open source project that follows Agile principles

Many open source project teams employ a working philosophy that is very similar to Agile—if not explicitly Agile. One of the main things that interested me in working on an open-source project was that most contributors to open source projects are working remotely. I wanted to see how Agile principles and practices could be applied to a team where none of the contributors are ever in the same room together.

Applying Agile to remote teams is of particular interest to me since we teach courses at LCC where students work together on class projects in Agile teams. Now that we are moving more and more of these classes online, we need to facilitate Agile teams in which the team members might never meet face-to-face.

The project I chose to join is the Visual Studio Code project. I chose this project because Visual Studio Code is a software development tool that I already use and like. It is also a very active project. The project currently has 378 active contributors and in a typical month, around 100 code contributions will be accepted by the team. Here's how the process works:

- Requests for bug fixes and new features come from the user community. Users enter requests in the issue tracking system, https://github.com/Microsoft/vscode/issues.
- Project managers plan iterations (sprints). These typically last about four weeks. In each iteration particular sets of features and bug fixes are prioritized.
- Team members (contributors) will have already cloned the source code repository. Any team member can select an issue to work on. It should be an issue that has been prioritized for the current iteration. They will mark the issue in the tracking system as an issue they are working on. Once they have completed the features or bug fix in their own copy of the source code, they issue a pull request.
- Members who have been approved by the project managers can "pull" the new code that is described in the pull request and test and review it. If the code quality is good, it will be added to the official source code repository. If the code has problems, comments will be sent back to the contributor who will then work on the code some more and issue another pull request.
- At the end of the iteration, more testing will be done and after fixing any problems, a new version of Visual Studio Code will be released.

3. CONCLUSION

Here is a summary of my learnings from this sabbatical

A. Agile is now the way to do software development

Every company I visited, and every company I read about, describes their software development process as Agile. However, there is quite a bit of variability in the way Agile principles are interpreted and applied.

B. Scrum is the Agile methodology used by most companies.

Every company I visited and most that I read about are using Scrum as their Agile methodology. In fact, many developers use the terms Agile and Scrum as if they were synonymous.

C. Women may find Agile project management as a new entry point to the high-tech industry

Not only was this the main topic of one of the conference sessions that I watched, but I also found that at two out of the seven local companies I visited, women who had did not have computer related degrees and who had very little software development experience were the Scrum masters.

D. Agile works well for both collocated teams and remote teams

Open source projects have a long history of developing software in a context where none of the team members are together in the same room (often not even in the same state or country). Many of the software development principles that have arisen in the open source community are parallel to those in the Agile movement. Agile principles and practices (Scrum in particular) can be easily adapted to work with remote teams.

E. Teaching a modified version of Agile Scrum is appropriate

Given the wholesale adoption of Agile by the local and national software industry, and the predominance of Scrum as the main methodology, we need to be teaching Agile Scrum to our students. But, there is minimal consistency as to what aspects of Scrum are applied or how they are applied by different companies. We should teach students the "official" version of Scrum, but have them participate in project teams that use a more stripped down version that is closer to what they will encounter in the local industry. We can also adapt Scrum to work with teams of students who are participating in an online class.