Meira Strauss

08 Aug 2018

Research paper

CIS 355 – Computer Systems, Networks & Information Technology - Summer 2018

# Abstract

The Agile Methodology used in software development drastically changed the way projects are built. The method was introduced in 2001 by seventeen developers who saw disadvantages in the traditional Waterfall Method. Agile looks for customer satisfaction as well as handles large projects by breaking them down into smaller goals. Some of the reasons for Agile’s popularity in 2018 is because the Agile method accepts feedback from the customer, can handle change midway, delivers quality results.

# The Agile Methodology

# Introduction

“A full 71 percent of organizations report using Agile approaches for their projects sometimes, often, or always” (Langley, 2017). Since the Agile Manifesto in 2001, more and more companies and businesses have been incorporating the Agile Methodology into their project management. In just a few years, Agile has become part of common workplace lingo and in many businesses a requirement for prospective employees.

The reason for Agile’s popularity is understood when looking at statistical success rates. This methodology has proven its success in recent years, as Agile Project Manager Anthony Mersino stated “Agile projects are statistically 2x more likely to succeed, and 1/3 less likely to fail than Waterfall projects” (Mersino, 2018). Because of the popularity of Agile, it is important to know how it was introduced, what techniques are used for its success, and what are the benefits of this methodology over previous ones.

# The History of Agile

In the 1970s through 2000, most businesses were using a methodology called the Waterfall Method. This method was presented in 1970 by Dr. Winston W. Royce and uses a step by step process. First the project is planned and designed in its entirety. Once the preparation is complete, all parts of the blueprint are created then tested and finally deployed.

This method is straightforward and understandable, but there are many disadvantages that were causing developers to search for a new process. Some disadvantages include non-adaptability to the design plan, no mid-process changes and comments, and a delayed testing period causing tasks to not be tested properly (Powell-Morse, 2016).

In 2001, seventeen developers, who later coined themselves “The Agile Alliance”, got together at a ski resort in Utah and created a new methodology called the *Manifesto for Agile Software Development*. The main theme of this Agile Methodology is to create a great project for the customer while working in an environment that “does more than talk about ‘people as our most important asset’ but actually ‘acts’ as if people were the most important” (Highsmith, 2001). The Agile Alliance wrote twelve principles behind the Agile Manifesto which explain the goals and beliefs of this methodology. Some of the principles include “Our highest priority is to satisfy the customer”, “We welcome changing requirements, even late in development”, and “We deliver working software frequently” (The Agile Alliance, 2001).

Since that fateful weekend in 2001, Agile has been implemented in hundreds of corporations across the globe. In addition, there are now different types of methodologies that all work under the Agile Manifesto’s framework, including; Scrum and Kanban. Once a person knows the background behind the birth of Agile and recognizes its goals, now he needs to understand the steps and tools of the Agile process.

# What is it?

The Agile Methodology’s biggest advantage is that it breaks down big projects into its smaller sub units, called user stories, and presents these user stories after short cycles, called iterations. By setting smaller goals, it allows for results after short units of time as opposed to waiting until the end of the project to see the results.

User stories consist of small, attainable tasks written from the view of the user. Instead of a task like “Build an ecommerce website,” the user stories will be “Jane can see a picture of each product for sale” or “Jane can search through the products by name and color” (Menne, n.d.). The way the user stories are formed is through a meeting with the customer where a list of all the necessary parts which the customer would like included in the project is written. These tasks then become the user stories of the developers. This makes the tasks more manageable and specifies exactly what the user wants.

The developer estimates how long each user story will take, the customer prioritizes the list and then the developer will execute the user stories to different Agile teams based on priority and estimated time the task will take to be accomplished (Rasmusson, 2011). Often the user stories are chosen by a group of developers and businessmen in a meeting called a “Planning Game” (Linchpinseo, n.d.). During the planning game before each iteration, each Agile team is given one or more user stories which the developers must complete in the amount of time allotted to the iteration. Most iterations are one to two-week slots of time in which the user stories are taken and built fully into workable, tested software (Rasmusson, 2011).

To keep track of progress and make sure that all developers understand the bigger picture of what is happening, there are stand-ups every day, planning boards for everyone to see, and meetings between iterations. A stand-up is where every member of an Agile team who is participating in an iteration, literally stands up and informs everyone on his or her progress. This allows everyone each day to help others, understand how the different user stories connect, and not have any overlap between user stories. The planning board is a place where all the user stories for an iteration are posted so all the developers can see the projects being done (Linchpinseo, n.d.). Finally, there are meetings between iterations where the past iteration is discussed, and a successful plan to move forward into the next iteration is drawn.

# Benefits of Agile over Traditional Waterfall Method

There are many benefits to the Agile Method over the Waterfall Method some of which have been mentioned previously. A few major drawbacks to the traditional Waterfall Method caused those seventeen members of the Agile Alliance to originally meet and think of a solution. Agile, a method that solved all their issues was signed and endorsed by every single one of the members. The main benefits and cause of such widespread popularity of the Agile Methodology are constant results, adaptability to change, and better-quality work.

Constant results mean that the customer does not have to wait months to see progress. Because the user stories are delivered to the customer after every iteration, the growth is witnessed throughout the development. Pieces of the big project are created and easily seen building up toward the end result. Constant results help both the customer and the user feel satisfied and more importantly, allow for feedback to see if the project is working how the customer and programmer imagined. In addition, there are functional and usable parts almost immediately, and because of the constant results, unimportant details are more easily perceivable. The developer does not end up wasting time on unnecessary elements (Menne, n.d.).

This is different from the Waterfall Method where the project in its entirety would be planned and then created. The results would only be given to the client after the completion of the entire project. There is no feedback mid-way and no functional parts until the end.

In addition to constant results, Agile allows for change of plans in the middle of the process. As best-selling author of “The Agile Samurai,” Jonathon Rasmusson said “Agilsts accept and embrace change even late in delivery process.” The reason Agilst can revamp the project midway, is because they are only doing small user stories at one time. After every iteration before moving on, it is easy to change the direction of the project by changing the user stories (Rasmusson, 2011). In addition, the feedback of the client after he or she is given results is used to help finetune and change the user stories to better satisfy the customer.

Conversely, the Waterfall Method sets all its goals in the beginning of the process and it is therefore extremely difficult to adapt them.

The steps of development, analysis, design, code, and test are done in both the Waterfall Method and the Agile Method, but in each respective method they are done differently. For the Agile Method, each user story goes through the steps as part of the user story’s iteration. First comes the analysis, then design, code, and test. Each piece of code can be tested before it is given to the client. In contrast, the Waterfall Method does the analysis of the entire project in the beginning. Then comes the design, then the coding of the whole project, and finally the testing.

Very often there is a crunch time at the end of every project in both methods. For the Agile Method, it means that some unimportant ability or user story might get emitted, but for the Waterfall Method, the end of the line for the steps of development is testing. Therefore, the testing of many projects is emitted. For this reason, the Agile Method returns better quality results (Rasmusson, 2011).

# Conclusion

For many years the Waterfall Method was used in most workplaces. Due to some lacking features, the Agile Method was introduced in 2001. Since then, hundreds of businesses and corporations have incorporated Agile into their software development process.

Agile uses ideas such as user stories, iterations, stand-ups, planning games, and planning boards to accomplish its goals. The main benefits to using the Agile Method are that products are delivered often, change is accepted and appreciated, and the quality of the results is increased. By understanding the benefits and techniques of the Agile Methodology, individuals and companies can optimize their potential and deliver to their clients better, quicker, and more personalized results.

# Bibliography

Denning, S. (n.d.). *Agile: The World's Most Popular Innovation Engine*. Retrieved from Forbes: https://www.forbes.com/sites/stevedenning/2015/07/23/the-worlds-most-popular-innovation-engine/#54deea177c76

Highsmith, J. (2001, February 13). *History: The Agile Manifesto*. Retrieved from Manifesto for Agile Software Development: http://agilemanifesto.org/history.html

Langley, M. A. (2017). Retrieved from Pulse of the Profession 2017: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

Linchpinseo. (n.d.). *A Beginners Guide To Understanding The Agile Method*. Retrieved from Linchpinseo: https://linchpinseo.com/the-agile-method/

Menne, M. (n.d.). *The Agile Process*. Retrieved from Launch Code: https://www.launchcode.org/blog/the-agile-process/?gclid=CjwKCAjwspHaBRBFEiwA0eM3kYPMveYH0G4aDzE1WQ9B4Yu3rafiUfYvSo0FrFWVLuivFuvyv47qrRoCwpQQAvD\_BwE

Mersino, A. (2018, April 1). *Agile Projects are More Successful than Traditional Projects*. Retrieved from Vitality Chicago: http://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/

Powell-Morse, A. (2016, December 8). *Waterfall Model: What Is It and When Would You Use It?* Retrieved from airbrake.io: https://airbrake.io/blog/sdlc/waterfall-model

Rasmusson, J. (2011). Retrieved from Agile in a Nutshell: http://www.agilenutshell.com

Reynosa, J. (n.d.). *Agile Methodology: The Complete Guide to Understanding Agile Testing*. Retrieved from QASymphony: https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/

The Agile Alliance. (2001, February 13). *Principles behind the Agile Manifesto*. Retrieved from Manifesto for Agile Software Development: http://agilemanifesto.org/