The Move To Agile

Mitchell Pomery
The University of Western Australia
21130887@student.uwa.edu.au

June 4, 2015

Abstract

The industry is obsessed with agile development practices. Businesses are moving their development teams to agile practices in an attempt to get more value out of the same development time. This paper discusses why teams have been adopting agile practices, and what must be considered when making the switch.

Keywords: agile, waterfall, development, process discussion

1 Introduction

The Agile movement started around the early 2000s and has since taken the industry by storm as companies, both large and small, have been converting from waterfall development patterns to agile teams in an order to deliver better quality software systems to clients faster. In examining the processes companies previously used and the ideas behind the agile development process we can examine this industry wide move. This move is being driven by a need to deliver more value sooner as society becomes geared towards instant solutions to complex problems.

2 Why Move From Waterfall

Waterfall is a very structured development process that aims to deliver a single application at the end of development that meets all of the clients' requirements. The process is carried out in several well defined steps, starting with design documents being created and signed off by the client before development starts. Development then occurs and finishes before testing is done. At any point in time the process can be halted, rewound back one step and then continued.

There are several places where problems in the system can arise while creating a system using the waterfall development process. What the client wants at the beginning of the process may change after requirements are finalized and development starts, meaning that the client is unhappy with the final outcome or that large changes to design need to occur in the middle of development. The design documents might be misinterpreted by the developer or the customer, again leading to the client receiving something that does not meet their expectations. Technologies are always changing and what is made might be completely incompatible or out of date by the time the project is finished. These downfalls all contribute to the industry wide move to agile practices.

3 Why Use Agile Methodologies

Instant gratification for the client has been a main driving point for the industry wide move to agile. As delivery windows are forced to become smaller and more frequent, changes need to occur faster without the side effect of buggier applications. Agile brings in a fast feedback loop where changes by the client side instantly feed back into the development teams and output from the developers can be seen sooner by the client. Agile methodologies bring higher visibility to work done by individuals, fostering the ownership of problems, through the use of smaller, well meshed teams. This leads to better solutions as individual developers deliver answers to their problem, rather than churn out code to complete the task that was assigned to them. These smaller teams work independently of any management, using one of two development processes that best fits their tasks and projects.

Agile has two development processes, Scrum and Kanban, that individual teams can use to ensure rapid delivery. Scrum has preplanned sprints where the work for an iteration is selected beforehand and organised. This work is all completed by a predetermined deadline allowing all the features to be released at once. It is good for delivering a single set of related changes to a system. Kanban deals with a continuous stream of work that can be delivered at any point in time. Work is taken from the top of an ordered backlog and delivered as soon as possible. This is good for tasks that continuously come in, such as bug fixes and maintenance. It is possible for a single project to contain smaller teams using a mix of both of these methodologies working together to create great solutions.

4 Picking Up Agility

The transition to agile is a slow and difficult process. It requires the entire business to be aware of the change to processes, developers and testers to be on board with the drastic changes that will occur. Successful adoption of agile methodologies in a workplace is evident through shorter release cycles and teams that are working well together on well defined tasks.

Agile methodologies are based off of a core set of principles. Customer satisfaction through rapid delivery is the highest priority. Teams should be small, close and self organizing. Individuals take ownership of tasks and see them through to completion. Creating an organizational shift to introduce these core principals is a slow process and both individuals and teams will adjust at different rates.

Common pitfalls when adopting agile are lack of knowledge and the unwillingness to adopt new practices. Forcing the adoption of agile practices straight away will cause harm and make adoption stretch out or become impossible. Changing the fundamental rules of peoples day to day jobs is a difficult process and agile thrives on team members who uphold the agile principles. The best way to start a movement towards agile is through small changes to day to day tasks to enable a seamless integration with agile practices. As teams begin to see the advantages of these small changes they will begin to start making their own changes towards the end goal.

Agile creates small teams of people that work well together without a hierarchy. These teams work closely together on their tasks, employing pair programming, code reviews and continuous integration to improve code quality and system. These teams work with no defined lead and instead make work decisions as a unit. Tasks that require a leader or single delegate are split up through someone volunteering or a team decision as to who is the best fit for the job.

Agile teams do have their own set of problems that need to be considered when creating and changing teams. Work that is considered hard, mundane or worthless can be ignored by everyone in the team in favour of easier more exciting work. This can cause problems where tasks are left undone for large periods of time until they become unmanageable and detrimental to the system. Avoiding this can be achieved by having each team set it's own rules that they self enforce to prevent work being left incomplete. Teams also suffer when working in large enterprises where people are moved between projects frequently. Team sizes can become disproportionate and balloon to eight developers with only one tester, then rapidly change to become four developers and four testers, all while incoming work remains constant. Taking the time to consider changes can help to minimize these issues.

5 The Future Of Development Processes

The agile movement has been great for the overall productivity of the development industry. Better feedback loops involving the client and the ability to easily change scope are great improvements on the waterfall model. What is going to happen next is just as important as what is happening now. The future holds something better than agile, where more value is created from developers and testers time, teams work together seamlessly and clients, developers, testers and managers are happier. Valve Software has taken agile a step further by creating a completely flat structure where projects gain traction and developer time by being more appealing than other projects. Everyone, from developers and testers to artists and writers, freely moves between projects to where they can find tasks to do.

References

[1] Gandomani, T.J & Zulzalil, H & Nafchi, M.Z 2014, 'Agile transformation: What is it about?', Software Engineering Conference (MySEC), 2014 8th Malaysian, pp. 240 - 245.

[2] Moore, E & Spens, J 2008, 'Scaling Agile: Finding your Agile Tribe', Agile, 2008. AGILE '08. Conference, pp. 121 - 124.

[3] Sureshchandra, K & Shrinivasavadhani, J 2008, 'Moving from Waterfall to Agile', Agile, 2008. AGILE '08. Conference, pp. 97 - 101.

[4] Taymor, E; Unknown Publication Date, 'Agile Handbook', available at: http://agilehandbook.com/agile-handbook.pdf

[5] Valve Software; Unknown Publication Date, 'Handbook For New Employees', available at: https://www.valvesoftware.com/company/Valve_Handbook_LowRes.pdf