2018 International Conference on Multidisciplinary Research
Track 1: Science, Technology, Engineering & Mathematics
https://doi.org/10.26803/MyRes.2018.14

Five Lessons Learned Using Water-Scrum-Fall in South Africa

Laurie Butgereit
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa

Abstract
The term *Water-Scrum-Fall* was coined by West et al. in 2011 to describe a hybrid methodology which is a by-product of attempting to introduce Scrum into an organisation which usually operates with a Waterfall Methodology. Water-Scrum-Fall typically appears in an organisation which had been using the Waterfall Methodology and in which a development team attempts to migrate to Scrum (or some other Agile Methodology). In such a hybrid, the first few phases (such as feasibility studies, funding exercises, requirements gathering, etc) are executed using a Waterfall Methodology. Then the development and possiby testing phases are executed using Scrum and are broken up into small sprints. The actual deployment, however, is then executed using again the Waterfall Methodology. This paper looks at a specific project in a South African listed company in which the development team implemented Scrum for the development phase and a large portion of the testing phase. This hybrid methodology was not necessary successful. This paper describes five of the lessons learned using Water-Scrum-Fall.

Keywords: waterfall, scrum, water-scrum-fall.

Introduction

In 1970, Dr Winston Royce developed the Waterfall Methodology (or Model) for software development. The Waterfall Methodology is a non-iterative methodology where a software project moved in its entirety from one step or phase to the next step or phase (Charvat, 2003).

In 2001, however, a number of people met in a ski resort in Utah, United States, and clarified certain values and principles which they held with respect to software development. These values and principles became known as the Agile Manifesto (Agile Alliance, 2017; Fowler & Highsmith, 2001). A number of new software development methodologies were created using this Agile Manifesto including XP, Lean Development and Scrum.

Attempting to transition an organisation from a Waterfall Methodology to an Agile Methodology is not without its challenges (Krasteva & Ilieva, 2008; Sumrell, 2007; Sureshchandra & Shrinivasavadhani, 2008). In many large organisations which have been typically doing Waterfall Methodologies, it is the development team who decides to implement an Agile Methodology. As will be more fully described in the Section appropriately entitled *Water-Scrum-Fall*, this hybrid methodology has been named Water-Scrum-Fall by West (West, Gilpin, Grant, & Anderson, 2011).

This paper specifically looks at some of the lessons learned in using Water-Scrum-Fall in a corporate environment in South Africa. The background to Waterfall Methodology, Agile Methodologies, and the hybrid Water-Scrum-Fall is provided in the next four sections. A description of the research environment is in the subsequent section. The specific lessons learned are then itemised followed by concluding remarks.

Overview of Waterfall and Agile

The Waterfall Model is a non-iterative methodology favouring a number of discrete phases or steps such as a requirements gathering phase, followed by a design phase, an implementation phase, a testing phase, a deployment phase, etc. The project moves through these steps in one unit. In other word, one phase must be completely finished before the project can migrate to the next phase. These distinct phases make it easy for project managers because there is well defined documentation which accompanies the project as it moves from phase to phase (Charvat, 2003). In contrast, however, the Waterfall Methodology is often difficult for the end user or customer because the first time that the end user or customer actually views the product is during the last phase.

The Agile Manifesto clarified that the signatories valued (Agile Alliance, 2017; Fowler & Highsmith, 2001): Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan. The Agile Manifesto also describes twelve guiding principles which emphasized frequent face-to-face meetings between business and development groups along with continuous software delivery.

Comparing Waterfall and Agile

The Waterfall Methodology is process oriented. There are well defined processes (or phases) which must be followed. On the other hand, Agile Methodologies are more people oriented.

The Waterfall Methodology is a command and control methodology were project managers assign tasks to various people. In contrast, Agile Methodologies are more collaborative when team members co-operate and collaborate to complete the required tasks. In addition, Agile team members themselves select which tasks out of a backlog of tasks they wish to work on.

The Waterfall Methodology depends on formal sign-off documents as a project moves from phase to phase. Agile Methodologies, however, depend more on informal meetings and face-to-face communication.

And, finally, the Waterfall Methodology is guided by tasks and processes whereas Agile Methodologies are guided by customer requirements (Nerur, Mahapatra, & Mangalaraj, 2005).

Adaption of Agile Methodologies in South Africa

The adoption of various Agile Methodologies has grown world-wide. In his masters' dissertation, South African Master's student Vanker has researched the adoption of Agile methodologies inside South Africa in his Master's dissertation *The Adoption of Agile Software Development Methodologies by Organisations in South Africa.* Vanker collected 85 responses from 25 software development companies in South Africa. Of those 85 responses, 71 (or 83.5%) used Scrum as their Agile methodology (Vanker, 2015).

Scrum

Scrum is one of the many Agile Methodologies. Jeff Sutherland was one of the original signatories to the Agile Manifesto and is often attributed to be the founder or author of Scrum. Sutherland himself, however, attributes many of the ideas basic to Scrum as coming from a 1986 article by Takeuchi and Nonaka in the Harvard Business Review entitled *The New New Product Development Game* (Sutherland, 2012; Takeuchi & Nonaka, 1998).

Regardless whether Sutherland or Takeuchi and Nonaka were the author/s of the Scrum Methodology, Scrum teams are self-organising teams which collaborate to do the planning, design, development, and testing of a software project in an iterative manner (Sutherland, 2012; Sutherland & Schwaber, 2013). Scrum defines a number of artifacts, a number of roles and a number of meetings or events.

The artifacts include the Product Backlog, the Sprint Backlog, and the Product Increment. The roles include the Product Owner, the Scrum Master, and the Team Members. The meetings or events include Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective and on-going Backlog Grooming.

Water-Scrum-Fall

In 2011, Dave West coined the term Water-Scrum-Fall. Water-Scrum-Fall describes a situation where a programming team (and possibly the testing team) are "doing Scrum" while the rest of the organisation is still doing Waterfall Methodology as can be seen in Figure 1 (West et al., 2011).

West uses the term *water* to describe the upfront work done by the organisation in doing feasibility studies, funding exercises, and requirements gathering. West uses the term *scrum* to describe the middle process where developers use the Scrum methodology to do actual development (and possible testing). And finally, West

uses the term *fall* to describe the final deployment step where the organisation's existing release policy is still implemented.

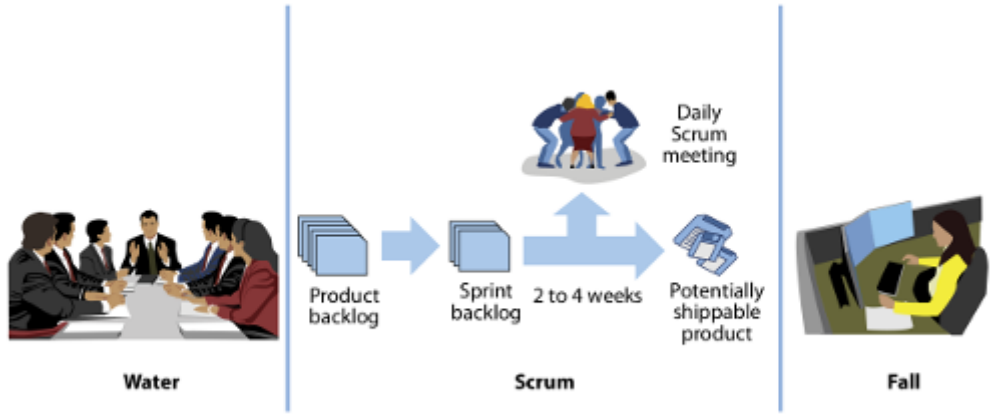

Figure 1: Water-Scrum-Fall as described by West (2011)

Research Environment

The company under research is a corporate listed on the Johannesburg Stock Exchange. The company is one of the largest vendors of secure tokens which can be redeemed for electronic goods such as airtime, data, wifi connectivity, electricity, and gambling facilities. In addition the company supplies other electronic facilities such as bill payments, traffic fine payments, event ticketing, bus ticketing, and lotto.

The company offers a switch which connects to the various cell phone providers, bill payment aggregators, ticketing agents, etc, available in South Africa. Prior to this research, the company offered a Windows Operating System (OS) based device which provides a user friend graphical user interface (GUI) interface to the switch. The Windows based device was placed at merchant stores and shops and would enable the merchant to on-sell the various products and facilities offered by the company to the end customer. The company did not deal with the end consumer of the product directly. All transactions when through a middle merchant.

The lessons learned itemised in this paper are specifically from a project to rewrite this Windows OS based GUI into an Android based app which could be deployed on much less expensive devices with embedded thermal printers.

Over the period of time which these lessons learned are obtained, the team varied from three (a product owner, a project manager/scrum master, one development team member) to eight (a product owner, a business analyst, a GUI specialist, a tester, a project manager/scrum master, and three development team members).

Lessons Learned

The project described in this paper entailed rewriting a Windows OS based application into an Android app. During this rewrite exercise, a number of improvements were made to the original flow of information and screens. In otherwords, it was not intended that the Android app would be an exact copy of the

Windows OS application. Certain operations would be streamlined, some new functionality would be added, and some obsolete functionality would not be incorporated in the Android app.

This section itemises five of lessons learned while using Water-Scrum-Fall in the project.

Lesson #1 – A Waterfall Project Manager and an Agile Scrum Master are two completely different types of roles

A project manager who works in a Waterfall methodology is accustomed to a command-and-control structure where the project manager assigns work to team members and the team members follow instructions. In contrast, however, in a Scrum environment, the team members' daily stand-up generates a list of impediments which are hindering their progress and the team members expect the Scrum Master to clear the impediments. This is, in effect, the team members giving instructions to the Scrum Master. In the case where a previous project manager is now filling the role of a Scrum Master, reversing the flow of instructions from project manager to team member and then from team member to Scrum Master can be difficult for a person to get used to.

During the course of the project under discussion in this paper, there were times when the team members looked to the Scrum Master and said things like "The lack of good test data is impeding my progress" or "The corporate firewall is blocking the Android devices from connecting to the server", the new Scrum Master did not understand those sentences to be instructions to him to solve the impediments. He understood the sentences to merely status updates and he expected the team members to solve their own problems. This change in attitude took a while to achieve.

This is a common problem as organisations migrate to scrum. For example, in some of their online training materials, Berteig Consulting argues that the number one common myth about Scrum is that a Project Manager is the same as a Scrum Master (Berteig, 2015).

Lesson #2 – A Business Analyst is not a proxy for a Product Owner

Over the course of this project, there were periods of time when the Product Owner attended the various Scrum meetings (especially the daily stand-ups) and there were times when the Product Owner did not attend those meetings. When the Product Owner did not attend the meetings, the Business Analyst attended as a proxy for the Product Owner. When this happened, the Business Analyst did not feel she could make decisions on behalf of the Product Owner. This slowed down development. Questions which the development team had about things such as screen flows or colour schemes could not be resolved quickly.

In researching this phenomenon after it happened, it was found that this appears to be a common issue. Various Scrum training organisations such as Manifesto (Bates,

2018) and Roman Pichler (Sumrell, 2007) (both based in the UK) maintain that a Business Analyst must not attend as a proxy for the Product Owner.

Lesson #3 – Formal Specifications are not Tasks or User Stories
Formal specifications as written up by a Business analyst often consist of ten to twenty tasks or user stories. In the beginning of the project under research, it was assumed that one specification correlated to one task. The specification (masquerading as a task) was placed in the backlog as written. It was quickly learned that the specification was, in fact, multiple user tasks or stories. As the project advanced, the development team learned to break up any formal specification presented by the Business Analyst into smaller tasks or user stories.

Lesson #4 – Continuous Delivery can not happen using Water-Scrum-Fall
The first principle of the Agile Manifesto is "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software". In a Waterfall environment or a Water-Scrum-Fall hybrid environment, the last step of software delivery or software deployment is controlled by a gate keeper who collects sign-off documents from testers and product owners before deploying software.

This not only extends the period of time between deliveries, it is often complicated in a Water-Scrum-Fall environment because one specification may be broken up into multiple tasks or user stories.

Lesson #5 – MVPs can not be Updated and Deployed Quickly using Water-Scrum-Fall
A number of Agile Methodologies including Scrum and Lean Development support the concept of a Minimum Viable Product (MVP). An MVP is the smallest version of the product which can provide some value for the end user or customer. Through the employment of continuous integration and continuous delivery techniques, an Agile team improves on the MVP and supplies new versions of the MVP to the end user or customer weekly or bi-weekly. In a Water-Scrum-Fall environment, however, the actual deployment of the software is typically still under control of a Waterfall Methodology and it is not possible for the Agile team to timeously deploy new software to the end user.

In the case of this project, often three months passed between new versions of the MVP.

Conclusion
As organisations try to move from a Waterfall Methodology to an Agile Methodology, there is often a hybrid state where parts of the project is still managed using a Waterfall methodology and parts of the project is executed using an Agile Methodology. West named this hybrid methodology Water-Scrum-Fall.

This paper looked at a specific project to rewrite a Windows OS based application as an Android app. The development team adopted a Scrum Methodology as defined

by Sutherland. However, the other steps of the project such as requirements gathering, testing, and implementation/deployment remained under a Waterfall methodology.

Five important lessons were learned during this project. Three of the lessons dealt with the issue of there not being a one-to-one mapping between the roles in the two different methodologies and the documents in the two different methodologies: 1) project leaders are not the equivalent of scrum masters 2) business analysts are not proxies for product owners and 3) specifications are not the equivalent of user stores or tasks. Two of the lessons dealt specifically with the difficulty (or even impossibility) of trying to do continuous delivery in a Waterfall environment.

Large organisations often find smaller organisations strongly competing with them in the marketplace. The new smaller organisations were often created Agile at their inceptions and the larger organisations struggle to compete. A new small Agile organiation can get an MVP of a new product to market very quickly while the larger organisaation is still struggling with writing user requirements documents.

If such large organisations wish to attempt to embrace Agility successfully, these five lessons learned and itemised in this paper need to be avoided.

References

Agile Alliance. (2017). *Agile alliance website.* Retrieved March 17, 2018, 2018, from https://www.agilealliance.org/

Bates, S. (2018). *Scrum in practice: the role of the business analyst.* Retrieved March 17, 2018, March 13, 2018, from https://manifesto.co.uk/scrum-in-practice-the-role-of-the-business-analyst/

Berteig, M. (2015). *Scrum myth 01 - the ScrumMaster is a project manager.* Retrieved March 17, 2018, 2018, from https://www.youtube.com/watch?v=a7mTMgUEjUQ

Charvat, J. (2003). *Project management methodologies: selecting, implementing, and supporting methodologies and processes for projects*. New Jersey: John Wiley.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28-35. Retrieved from http://andrey.hristov.com/fht-stuttgart/The_Agile_Manifesto_SDMagazine.pdf

Krasteva, I., & Ilieva, S. (2008). Adopting an agile methodology: Why it did not work. Paper presented at the *Proceedings of the 2008 International Workshop on Scrutinizing Agile Practices Or Shoot-Out at the Agile Corral,* pp. 33-36.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM, 4*8(5), 72-78.

Sumrell, M. (2007). From waterfall to agile-how does a QA team transition. Paper presented at the *Proceedings of AGILE,* pp. 291-295.

Sureshchandra, K., & Shrinivasavadhani, J. (2008). Moving from waterfall to agile. Paper presented at the *Agile, 2008. AGILE'08. Conference,* pp. 97-101. Retrieved from http://ieeexplore.ieee.org/abstract/document/4599456/

Sutherland, J. (2012). *The scrum papers: nuts, bolts, and origins of an agile framewor*k. Cambridge, Massachusetts: Scrum, Inc.

Sutherland, J., & Schwaber, K. (2013). *The scrum guide: the definitive guide to scrum: the rules of the game*

Takeuchi, H., & Nonaka, I. (1998). The new new product development game. *Japanese Business: Part 1, Classics Part 2, Japanese Management Vol.2: Part 1, Manufacturing and Production Part 2, Automotive Industry Vol.3: Part 1, Banking and Finance Part 2, Corporate Strategy and Inter-Organizational Relationships Vol.4: P(TRUNCATED),* *6*4(1), 321. Retrieved from https://hbr.org/1986/01/the-new-new-product-development-game

Vanker, C. (2015). *The adoption of agile software development methodologies by organisations in south africa.* Unpublished University of KwaZulu-Natal,

West, D., Gilpin, M., Grant, T., & Anderson, A. (2011). Water-scrum-fall is the reality of agile for most organizations today. *Forrester Research, 26*