SCRUM for the Small Developer

John Wensink

MIS350–1 Information Systems Analysis and Design

Colorado State University-Global Campus

Dr. William Willette

August 6, 2018

**Introduction**

Hello and thank you for your time. Since I have started working for the company, my team has been asked to develop our projects in the “waterfall” style of product development. In this type of SDLC, progress flows steadily and sequentially through its phases. We are given the scope, budget, and timeframe upfront and hold those initial conditions to be immutable, or unable to be revised. This style of software development has several shortcomings including projects abandoned before completion, successfully completing a project only to find the final product is now irrelevant, employee fatigue, and stakeholder disconnectedness. In some situations, it makes sense to use a waterfall system. An example of this is if we were constructing a building, it is unlikely that our initial project’s parameters would change very much if at all. Software development is nothing like this because the needs of the organization, its members, and employees are constantly changing.

**Summary of the Proposal**

Since I have started working for the company, I have noticed that the company has started to adopt a “spiral” type of SDLC to allow for more flexibility in our development process. In this working paper, I intend to introduce a different style of SDLC known as “agile methodologies” specifically SCRUM with the expectation that we will be able to complete a higher number of projects, more efficiently, at a higher level of relevancy, with a lower cost to our organization. A recent article in the Harvard Business Review (2018) describes how despite eight out of ten companies are committing to adopt an agile systems development life cycle, only about four out of ten companies have adopted it. The companies who have adopted agile methodologies have reported 60% higher revenue and profit growth compared to their non-agile counterparts. (Panditi, 2018)

**Description of SCRUM**

SCRUM is a specific subtype of “Agile Methodologies.” SCRUM seeks to break down large projects into smaller themes and to divide those themes into even more workable features. The focus is on failing, learning, and delivering “fast”. By segmenting a large project into smaller subsections we are able to build sub-modules that can be joined together. The goal is to produce a “minimum viable product” (MVP) so that we can begin gathering feedback from early adopters. Within agile methodologies, more value is placed in teams and individuals rather than traditions and toolkits. Acknowledging that the business landscape is always changing, and the ability to respond to that is valued more than sticking to an immutable plan. Whereas in the waterfall style SDLC scope, budget, and timeframe were all fixed, with agile methodologies we allow just one of these three legs to change, scope. This may seem like a small difference but the implications are immense. When the scope of the project is flexible, it allows our development teams to adapt to changing landscapes and deliver a high-quality product that is both relevant and timely. SCRUM is a term that is derived from the word scrummage. If you have ever watched a rugby match you would know that a scrummage is a type of huddle that the players engage in to get the play started. In a rugby scrum, the players gather together as they attempt to gain the momentum required for the coming play. SCRUM methodologies share many similarities such as the “SCRUM.”

**SCRUM Roles**

In order to talk about why the daily scrum, otherwise known as a “standup” is important, we must first understand the different roles in the SCRUM system. There are three distinct position titles in SCRUM, each with a critical role in the success of the project. First, the “product owner” (PO) is the liaison between the development team and the stakeholders. The PO is a full-time position filled by a dedicated representative of the stakeholders. The PO is primarily responsible for communicating the ever-changing needs of the stakeholders with the development team. Next is the SCRUM Master. The primary role of the SCRUM Master is to remove roadblocks for the development team. Finally, we have a development team. The development team is ideally a seven-person team (+/- 2 people) that between them share most if not all of the competencies required to deliver a successful product. The hiring of outside consultants is acceptible for certain areas where specialized skills are needed.

**The SCRUM Team**

As we have said the SCRUM team is ideally seven members strong plus/minus two members. This may seem arbitrarily specific, but research shows (O’connell, 2017) that adding more team members than this can overload communications channels, decreasing efficiency. Team members should be diverse, qualified, and dedicated to the project. It is essential that each team member is focused solely on the task at hand, meaning that we are not looking for part-time interest in this project. The SCRUM team should, if possible have a work area where all members can work closely together as they are more likely to be able to collaborate with each other easier than if they were located in separate areas. If this is not possible, however, it just means the team members have to work a bit harder and take advantage of tools such as video conferencing, and online collaboration software hosted in the cloud.

**SCRUM Procedures**

The vision of a project tells everyone involved where this project is headed, and what kind of value we aim to bring to market. From this vision, we can start to deconstruct a large idea down into smaller parts. We call these parts themes and they are a taxonomy of similar ideas about what needs to be developed. Themes are then broken down further to features. For example, if we are making a ride-hailing app some themes might be “registration, rides, and payments” and from these themes, we might breakdown registration into the features of “data security, database additions, and ID verification.” Smaller pieces yet are needed to bring the features down to a functional work level. We call this level “user stories” they share a similar format. (USER) wants to accomplish (TASK) so that there is a (BENEFIT). An example being:

“As a developer, I want to add a user to the database, so that they can register for the service.”

As we generate more user stories, we must also have some sort of “acceptance criteria” where the SCRUM team, the PO, and the Scrum Master agree on what criteria are required to label something as complete.

One of the cornerstones of SCRM is the daily stand-up meeting. Every day for no more than 15 minutes, the SCRUM team gathers together at the taskboard for a stand-up huddle. In the stand-up, three items are discussed:

1. What did you do yesterday?
2. What did you do today?
3. Is anything Blocking your progress?

(O’Connell, 2017)

The daily stand-up is an area where we can expect to benefit from immediate feedback and problems can be addressed. Issues that cause work blockages are brought up and the expectation is that SCRUM team will assist the struggling teammate. If the team can not fix the problem within one day, the next day the SCRUM Master steps in to remove the roadblock. If he can not, the next day the PO steps in to assist. This allows for rapid problem resolution and allows the SCRUM team to continue working towards the end deliverable.

**Why SCRUM Now?**

Now that we have talked about the basics of how SCRUM works, now let's explore the benefits of employing an agile methodology like SCRUM. For the past two decades, agile methodologies have been trending towards the SDLC of choice for software engineering. This is because of the constantly changing needs of all stakeholders involved in a software development project. Loosening the restrictions on the scope of the project, agile methodologies allow the team to be flexible enough to adapt to changing needs.

The goal is to get the project done, and it is impossible to know from the start exactly how that project is going to unfold. In rugby, the goal is to win the game one or two yards at a time and SCRUM is very similar in how it splits up deliverables into more manageable pieces. If we are able to win each and every small battle, with proper planning the final deliverable will fall into place. Another highlight of SCRUM is the balance between the parties involved such as the SCRUM team, the PO and the Scrum Master. Let’s say for example that the PO was stressing the team out with more and more demands in a faster and faster time frame. This is balanced by the SCRUM Master stepping in to relieve some of the pressure and set realistic expectations for the PO. Finally, SCRUM has the intangible benefit for the SCRUM team that is a sense of accomplishment that they are able to feel pride in their work frequently as the measurements for success are constant. Team members can develop a sense of trust for their teammates and learn to rely on their expertise fomenting team cohesion.

**Conclusion**

It is my sincere belief that agile methodologies will take Moonlighting Software LLC into the next decade and beyond. I have attempted to convince you of why SCRUM is a superior SDLC when compared to the waterfall method, and how relaxing restrictions on project scope allow for greater flexibility to keep up with changing business needs. I have laid out the basics of SCRUM and agile methodologies so that their merits can be discussed. SCRUM is a sufficiently deep topic that many professionals have dedicated their careers to its implementation. I ask that you think about what I am proposing and continue to learn about the benefits of SCRUM and agile methodologies in general. Thank you for your time.

References

Balaji, S., & Sundararajan Murugaiyan, M. (2012, June). Waterfall Model Vs. Agile: A Comparative Study on SDLC. Retrieved from <http://jitbm.com/Volume2No1/waterfall.pdf>

O’Connell, K. (2017, March 23). SCRUM: The Basics. Retrieved April 7, 2019, from <https://www.lynda.com/Business-Skills-tutorials/Scrum-Basics/550619-2.html?org=csuglobal.edu>

Panditi, S. (2018, March 29). Survey Data Shows That Many Companies Are Still Not Truly Agile - Retrieved from <https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile>