

# **What are the effects of the scrum philosophy on software development in relating to software developers productivity and effectiveness?**

**COMP150 - Agile Development Practice**

1605109

March 20, 2017

This paper will look at the effect the scrum philosophy has on developers productivity and effectiveness software development. Further to this, the paper will look at other types of agile development and compare these with non-agile philosophies.

## **1 Introduction**

This paper will explore in depth the methodology of the scrum philosophy and its impact on the overall effectiveness of software development. Through comparing the requirements for effective software development against the results scrum philosophy in development, this paper aims to understand the effects of scrum on developers productivity. This paper will highlight a few

different philosophies and look at their effectiveness in different scenarios for a wider view point. Understanding the effectiveness of developmental philosophies, such as scrum, impacts the game industries as it will inform them on how best to produce games and make use of their developers during the process of coding.

## **2 Advantages and Disadvantages of Scrum**

When producing a piece of software within a team you need to use a method and one of the most popular if not the most popular is agile, This paper looks specifically at scrum. Some of the main benefits of scrum is the use of a scrum master. The scrum master will support the team meaning it will take away any strains from them so they can concentrate on the job they have been giving, this has a massive increase on the effectiveness on the use of the developers time. Next there is the product owner who as the software will organise and prioritise the backlog [1]. This means they can properly look at how the software is being developed then analyse and see what is next most important in the backlog. This is very important in the game development industry as when producing games, it is very common for new problems to appear. Communication is very important with scrum philosophy this means that if the team has poor communication it will jeopardise the whole piece of software.

## **3 Scrum vs a non-agile philosophy**

Waterfall development is another very commonly used methodology of software development so we will use this to compare against the scrum methodology. As you can see in the diagram below the agile methodology is far more

efficient when it comes to game development, instead of doing a single long sprint they constantly go over and make sure what they have produced is working and not bringing up new problems.

All of this shows that the agile methodology more specifically the scrum philosophy has a massive improvement. when doing the waterfall, you risk spending a lot of time on a piece of code that could bring more problems. Therefore, the agile methodology is better as it constantly gives you a regular check-up on how your piece of a code is coming along, this does have an added time in the whole processes of making a game but saves the risk of coming across a massive problem and then not having time to fix.

## **4 Agile vs Component Based**

These two types of development are far from similar [2] these two approaches are quite different in terms of technological and organizational. Agile gets the software made quickly and efficiently this contrasts the component based where people get given a section or component to produce. Component is all about making a high quality rather than fast production time meaning it is supposedly not to have any problems. This further supports my question due to agile being the faster more effective solution if you are thinking about time management. This relates to game production as the investors will have to make up their mind depending on what they want, if they want a really high quality piece of code they go for the component based as opposed to the high speed and cheaper agile methodology.

## **5 Agile Outside of Software Development**

The philosophy of using the agile in development can be further used outside of just software, it has been used and proven to be very effective outside of the software development. This proves how well the method works. The following list of agile characteristics was abstracted from the agile values and principles: Interpersonal interaction, Working product or service, Customer/user collaboration, Responsiveness to change, Continual delivery of customer value, Self-organizing, multifunctional collaboration, Leadership by the motivated, Technical excellence and simplicity [3].

All these items listed are all transferable skills given to you by the agile philosophy showing how useful and effective it can be used for. Each one of these could be used outside of agile, for example a sport team they will need to be responsive for change and need all user interaction and so on, proving this is a well diverse philosophy. It relates to game production due to the sheer efficiency usable in everything.

## **6 Conclusion**

To conclude everything, I have read and looked at shows that the use of scrum in general programming practice and especially games development when team work is key it is very important to get the correct methodology to optimise everyones timing. Methods such as waterfall and many other are outdated and are now being replaced by the agile philosophy as it is far more efficient. It makes the most effective time use of the developers without wasting much time but potentially saving many hours.

## References

- [1] D. X. Houston, “Agility beyond software development,” in *Proceedings of the 2014 International Conference on Software and System Process*, ser. ICSSP 2014. New York, NY, USA: ACM, 2014, pp. 65–69. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/2600821.2600837>
- [2] W. Radinger and K. M. Goeschka, “Agile software development for component based software engineering,” in *Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, ser. OOPSLA '03. New York, NY, USA: ACM, 2003, pp. 300–301. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/949344.949422>
- [3] R. P. Maranzato, M. Neubert, and P. Herculano, “Moving back to scrum and scaling to scrum of scrums in less than one year,” in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, ser. OOPSLA '11. New York, NY, USA: ACM, 2011, pp. 125–130. [Online]. Available: <http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/2048147.2048186>