

What are the key challenges game developers encounter when applying agile in a remote context whilst using SCRUM?

COMP150 - Agile Development Practice

1605240

November 26, 2016

The use of agile is becoming more popular with teams of all sizes within the games industry, however when distanced they face challenges. This essay will be looking into the key challenges game developers face whilst applying agile in a remote context and searching for ways to make this process as efficient as possible. In order to accomplish my task I will be sourcing papers from academics and authors on what problems they have faced whilst using agile in a remote context and also papers which talk about solutions and I will discuss my views on whether or not I think they are suitable.

1 Introduction

This essay will be reviewing the adoption of the agile methodology and the key challenges game developers face when it is being used in a remote context with distributed teams. The agile methodology is a method that focuses on

individuals and interactions, working software, customer collaboration and responding to change [1] . Most teams of game developers use SCRUM which is an agile development method. SCRUM is an iterative approach where teams do two to four week sprints where they prioritise certain user stories they have in a product backlog [2]. They hold daily stand ups where every team member answers three questions, what did you do yesterday? What are you going to do today? And what problems did you encounter/encountering?[2] To make SCRUM as effective as it can be the main thing is good communication between team members, yet when teams are distributed they encounter challenges, the main one being communication and time zones which I will be looking into and trying to suggest alternatives to them.

2 Main Body

From my research Ive concluded that communication is the biggest issue encountered by game developers using agile in a remote context, however communication is very broad and can be split up into many different smaller issues, the first of these issues is how do they host the scrum if everyone is split up They use some of the many important tools they have available to them such as web conferencing and they host the daily scrum meetings online [3]. It is extremely important to game developers so they can discuss how they are getting on, what they are going to do, if theyve got any problems and also it helps put a face to an online name [3]. I think this is important because everyone knows what each other is doing so nobody does the same task and its a lot more friendly when you see someones face instead of speaking them online. This in turn helps build up trust and friendships making you care more about the team. The lack of communication between game developers can lead to other game developers on the team not getting the same vision

of the project, and not knowing the task at hand. This can lead to team members having to ask multiple times to understand clearly what they had to do [4], this takes up more time than it should as the daily scrums should be 15 minutes maximum. To fix this communication between team members the instructions should be clear and precise so no questions are needed.

Another pressing issue that links into the last issue is when teams are based in different parts of the globe their time zones can vary making it hard for teams to sync up working times [5] [6] or more importantly share work with each other. When sharing work with each other teams are sometimes break the code making it hard for the other team to figure out what theyve done wrong, what most teams have implemented to fix this is they dont leave the office until the last code that was built didnt break the build they were working on [3]. This works well meaning no time is wasted for the other team having to try and fix the code.

One other issue that can have quite an impact and difficult to avoid is language barriers, because if people speak a different language [?] teams have to get tools to translate and makes communication more of an issue. Part of this issue is also when people have different cultures [6] as the quality of the work can be harder to manage as expectations can sometimes be culturally influenced [2] this can be a problem that can be solved with the daily scrum meetings being online so every team member can get a vision of the game and know the quality it should be.

As mentioned above sharing work is a big part of how smoothly the project runs, the teams should set up shared repository [3] so they can all access the work and help each other if people cant manage to do a task, teams also have programs that allow multiple people editing the code at once, this benefits the team massively as it saves the team having to send around the latest

versions of code to each other, it just gets synced with everyones computer.

Although you would think working together in the same office would have nothing but advantages youd be wrong. Research has shown that multitasking and being distracted can lower your IQ by ten points when performing difficult tasks [7]. This can lead to poorer work being produced by the game developers which is not good when trying to make the best game they can. Other advantages of being distributed are it is cost efficient, you have access to talents all across the world and productivity is higher [8]. Employers love this as they can hire game developers from all across the world which gives them a lot more choice to pick the best employees and with them being in a different location it means they get more work done.

3 Conclusion

To conclude my essay I think that if game developers use the agile method by following the guidelines, teams have constant, clear and precise communication and make use of all the tools they have available then I believe distributed teams can create projects just as good as or even better than when the team would be collocated. I believe this because the papers show that communication is the biggest issue and if they manage to break that one key issue down then the projects should go smoothly.

References

- [1] J. Highsmith and M. Fowler, “The agile manifesto,” *Software Development Magazine*, vol. 9, no. 8, pp. 29–30, 2001.
- [2] L. Pries-Heje and J. Pries-Heje, “Why scrum works: A case study from an

- agile distributed project in denmark and india,” in *2011 Agile Conference*, Aug 2011, pp. 20–28.
- [3] A. Danait, “Agile offshore techniques - a case study,” in *Agile Development Conference (ADC'05)*, July 2005, pp. 214–217.
 - [4] D. S. Cruzes, N. B. Moe, and T. Dyb, “Communication between developers and testers in distributed continuous agile testing,” in *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, Aug 2016, pp. 59–68.
 - [5] M. Paasivaara, S. Durasiewicz, and C. Lassenius, “Distributed agile development: Using scrum in a large project,” in *2008 IEEE International Conference on Global Software Engineering*, Aug 2008, pp. 87–95.
 - [6] S. Berczuk, “Back to basics: The role of agile principles in success with an distributed scrum team,” in *Agile 2007 (AGILE 2007)*, Aug 2007, pp. 382–388.
 - [7] H. Kaur, H. M. Haddad *et al.*, “Distributed agile development: A survey of challenges and solutions,” in *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015, p. 42.
 - [8] C. E. Matthews, “Agile practices in embedded systems,” in *Proceedings of the compilation of the co-located workshops on DSM'11, TMC'11, AGERE! 2011, AOOPES'11, NEAT'11, & VMIL'11*. ACM, 2011, pp. 249–250.
 - [9] A. M. E. Hamid, “Upgrading distributed agile development,” in *2013*

*INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL
AND ELECTRONIC ENGINEERING (ICCEEE)*, Aug 2013, pp.
709–714.