Undergraduate students lack the maturity skills in order to use Agile methodologies, such as Scrum efficiently

COMP150 - Agile Development Practice

1506919

February 25, 2017

Undergraduates lack the skills neccessary to use Agile properly. More specifically why, in a classroom setting, does the allocation of tasks become uneven and biased? Futhurmore there is no financial benefit or maybe there is no interest in the task, which has a negative effect on the team collaboration in Scrum. This paper adresses the topic by looking at case studies based in a classroom setting, in which the students are using an Agile methodology for the first time. This will look in depth at the collaboration between the student groups, comparing how professionals using Agile methodologies compare to amateurs. Hopefully this will find productive methods to teach students how to collaborate in a more professional way.

1 Introduction

This paper will discuss the implementation of Agile methodologies [1] (focusing on Scrum) in non-professional teams, particularly within the classroom setting, when Scrum can quickly become biased [2]. This could refer, for example to one group member who may be given the majority of the workload, or the more complex of the Scrum tasks, while another member is given an easier task to complete in the same time period. It could be suggested that the chances of this occurring is greatly increased if there is a member with an introverted personality. This is in comparison to Agile being used properly in a professional games industry, where there is a more mature view of the working relationship [3]. After outlining these problems, this paper will present possible solutions in the form of practice collaboration tasks, teaching the correlation of daily stand up meetings, positivity and team productivity [3].

2 Agile in the classroom

Agile methodologies have been used exclusively in professional settings for longer than it has been taught in universities [3], but as the use of Agile increases within the software development and games industries, it is important for students to learn and understand these methods before leaving education[4]. Any type of software development relies on group collaboration, including coordination, organisation and motivation [5], this is also a vital part of Scrum, additionally students need to respect, understand and cooperate with every team member to succeed, as well as showing courage and openness. In other words the teams that showed the most maturity worked the best [3][6]. Kropp, Meier and Meteescu state the educating of students in advanced

collaboration skills is crucial for teams to overcome specific challenges, making the students aware of the importance of maturity within the team [2], students are usually unfamiliar with working in large teams of six or seven individuals, in which each person is responsible for different parts of a set goal [7], but with online collaboration tools (e.g. Trello and Github) as well as fun challenges and a tutor acting as project owner encourages student commitment and confidence[3][4]. Devedzic and Milenkovic who take an opposing view, state that it is sometimes best for the students to discover things on their own [8], but provide no proof to backup this claim, this might be true for extrovert students but may make the introverted individuals feel like their unable to speak out. Anslow and Maurer wrote about their experience of introducing Agile methodologies (Scrum and Extreme programming) to computer and information science students, and found that because of the students lack of experience they did not realize the time scale of completing some tasks, therefore the workload became unbalanced in the teams [4], in addition less experienced teams are unlikely to use pair programming and daily stand-up meetings but usage will grow with experience [7].

3 Agile in the professional game studio and other businesses involved in software development

The size of teams needed to produce a triple-A (AAA) game increases every year, creating uncertainty as to which management technique is best for them [9] [10]. The use of Agile in managing software projects is becoming increasingly popular[11] an example of this is in game studios, in which everyone contributes different parts to make a successful game. The Waterfall method of managing software, which is designed by keeping each team within

the same occupation, means that artists in are one group and programmers in another before the game gets tested. Therefore all the problems are logged before the waterfall is repeated again. In the case study at Runescape made by the project leader Conor Crowley, the main problems that occurred was that the notes from one team to the other could be misunderstood [12]. In contrast to the Waterfall method, Agile differs instead by separating the different roles, creating teams with workers from each of the different disciplines, which in affect create small game studios responsible for their own part of the full working game. New tasks are set each month for every small team instead of trying to tackle every task at once. Furthermore the project leader is responsible for setting tasks and making sure everything is ready on time for the release date [13]. In this sense Agile enables fast user feedback during the development process and when used with Scrum which modifies the priority list, decreases the time spent on game development [9]. Companies that use the Agile method, not limited to game studios but software development businesses as well reported that handling changes and the time it took to get the product onto the market improved [14]. Good communication skills are a crucial part of working in the game studio, especially willingness to talk about challenging issues, understanding feedback and general problems without fear of judgement [2]. In a series of interviews conducted by Kropp, Meier and Mateescu with workers from several I.T. businesses one was recorded stating For those that have been part of the team it is not a problem, but new team members have to get there At first you have to force them to talk another interviewee stated With time one can speak openly with each other. From these interviews it seems that lack of maturity is not just limited to students [2].

4 Conclusion

After reading papers that examined how Agile methodologies, such as Scrum were implemented in both professional software development businesses, including game studios, and at universities in which Agile was being taught to students for the first time, it appears lack of maturity doesnt just apply to students, but also to new teams, or new members going into existing teams in a professional environment. Therefore maturity seems to be effected by how long the team members have known each other in their group, but the longer they remain within the same team the more successful the Scrum tasks are, additionally everyone becomes more open and truthful. Perhaps students are more likely to exhibit these behaviours, particularly in their first year of university because everything is new to them.

References

[1] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "Manifesto for agile software development," 2001.

[2] M. Kropp, A. Meier, M. Mateescu, and C. Zahn, "Teaching and learning agile collaboration," in *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2014, pp. 139–148.

[3] M. Kropp, A. Meier, and R. Biddle, "Teaching agile collaboration skills in the classroom," in *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*. IEEE, 2016, pp. 118–127.

[4] C. Anslow and F. Maurer, "An experience report at teaching a group

based agile software development project course," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education.* ACM, 2015, pp. 500–505.

[5] A. Schroeder, A. Klarl, P. Mayer, and C. Kroiß, "Teaching agile software development through lab courses," in *Global Engineering Education Conference (EDUCON), 2012 IEEE.* IEEE, 2012, pp. 1–10.

[6] K. Beck, *Extreme programming explained: embrace change.* addison-wesley professional, 2000.

[7] M. Kropp and A. Meier, "Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight," in *2016 IEEE Global Engineering Education Conference (EDUCON).* IEEE, 2016, pp. 1003–1011.

[8] V. Devedzic *et al.*, "Teaching agile software development: A case study," *IEEE Transactions on Education*, vol. 54, no. 2, pp. 273–278, 2011.

[9] J. Mateos-Garcia and J. Sapsed, "Adopting agileand scrumpractices as organisational becoming: Cases from the uk video games industry," in *Euromot 2008: the Third European Conference on Management of Technology*, 2008.

[10] R. McGuire, "Paper burns: Game design with agile methodologies," *Gamasutra–The Art & Business of Making Games*, 2006.

[11] V. One, "State of agile development survey results," 2012.

[12] M. B. Snapp and D. Dagefoerde, "The accidental agilists: One team's journey from waterfall to agile," in *Agile, 2008. AGILE'08. Conference.* IEEE, 2008, pp. 171–175.

[13] Conor Crowley, "Making release day just another monday," [Online].

Available: http://www.gdcvault.com.ezproxy.falmouth.ac.uk/play/ 1023206/Making-Release-Day-Just-Another, [Accessed: 23-Nov-2016].

[14] M. Kropp and A. Meier, "Swiss agile study 2012," 2013.