How can Agile philosophy maximise skill learning in a small game development team?

COMP150 - Agile Development Practice

1707981

2017-11-16

Game development is more than a process of creating games. It is a process of learning, collaboration and discovering what is required to create a unique and interesting experience for the player. The effects of learning from a product's development are inspirational to a team, but it also calls to question how much they can learn how to produce, whilst producing it. An exploration of learning tendencies reveals Agile development's striking ability to inspire a team's skill learning throughout the collaborative development process, and proposes some potential techniques to augment this positive effect.

1 Introduction

A key principle in the Agile philosophy is iteration [1]. The iterative process of continually improving according to previous mistakes virtually matches the definition of learning. However, Agile philosophy declares that working software is the primary measurement of progress [2], and is focused on producing an ideal product in a short timeframe – contrast to learning how to do it with good practice. This can raise several issues in a traditional group environment, particularly where skill gaps appear [3].

This study aims to discover methods to improve the learning of production skills during the production process itself, taking place in a hypothetical small game production studio incorporating the Scrum framework.

2 Why Scrum?

A project owner may be tempted to choose an Agile workflow primarily for the game's sake. Due to its dependence on unpredictable factors such as look and feel, a finished game will often be very different to its initial design [4].

As a result, most game developers adopt a workflow that is iterative [5, 6, 7], matching and justifying Agile's 'responding to change' [2] clause.

Interestingly, game development companies as a majority don't use an Agile workflow. While evidence suggests that most game industry workflows are "Agile" philosophically (at least in the early stages [6]), many create their own personalised workflows, and others use no particular workflow at all [7]. This calls to question whether Agile development is truly appropriate for games, or whether adjustments should be made. That is beyond this study's scope, but Agile's benefits to team autonomy can't be ignored.

3 How to successfully not learn

A paper by L. J. Barker and Kathy Garvin-Doxes [3] highlighted a particular concern that threatened learning in a university project team context. Under the pressure to deliver an equal work standard, being in a team of more experienced developers can lessen an inexperienced student's own desire to contribute. This negatively impacts motivation and productivity [3, 8].

Psychology and motivation are the core of this type of issue. One influential study [9] observed that peers who were encouraged to learn by an extrinsic goal, such as money, learned less than those who were encouraged with an intrinsic goal, such as understanding themselves better. The game is an extrinsic goal, which could direct focus to the game's quality at the expense of learning. This puts pressure on less experienced or junior team members.

These pitfalls call to question how Agile philosophy could accommodate them. It is, however, arguable that Agile already does this through the 'interaction' clause [2], especially compared to, for example, a more traditional *waterfall* workflow. In a *waterfall* workflow, teams are typically fixed to a single plan for an extended period with minimal process review [10]. Much unlike Agile, this would lack individual autonomy, would be focused on an extrinsic goal (the plan), and would not accommodate self-reflection. Meanwhile, the strong single-track focus on producing the work would introduce pressure as noted in [8].

4 How to improve

4.1 Autonomy

This is an existing characteristic of Scrum that is worthy of mention: One of Scrum's characteristics is the Scrum board, wherein team members choose the tasks they plan to do [11]. Choosing one's own tasks could boost motivation. This is because it promotes autonomy [9], leading to better learning and recall, contrast to being set specific tasks by a leader.

4.2 Intrinsic goals

Boosting team discussion of these key issues could improve the situation. To recover motivation, awareness of the intrinsic goals could be gently raised by adding a 'what I want to learn today' clause to the Scrum stand-ups, particularly for juniors. This may overlap with 'what I plan to do today', and so may seem redundant. However, simply raising one's intrinsic learning goals could itself reproduce the positive results shown in the study [9]; and perhaps more importantly, give more knowledgeable team members an opportunity to help.

4.3 Reflection

The same could go for a 'what I learned yesterday' clause: It is well-documented that groups who reflect on their learning process, rather than just the task they achieved, are typically more successful in their endeavours [12, 13]. Reflection is already incorporated into most Agile workflows for similarly educational reasons [1, 11], except with a greater focus on the product. For a developing or junior team, educational focus should be added. When communicated clearly, reflection may also spark discussion among peers about the learning topics in specific, and bring to attention those who understand it and those who wish to learn.

4.4 Peer learning

Peer learning is actively practiced in industry, where openness to learning [7] and willingness to help others [4] are vital elements to a team member's personality. Encouraging the application of additional individual-oriented support, such as pair programming between peers of ideally varied skill gaps, would help to emphasise the essential goal of sharing knowledge. It may also aid troubleshooting in situations where an unknown problem is hindering production [4, 9].

In fact, [8] also found working in pairs to be a considerable solution to the skill gap. While it may be viewed as a burden on the surface, the act of knowledge sharing itself is known to have a positive effect not only on the learner, but the teacher as well [14, 9].

5 Conclusion

The Agile workflow can accommodate learning in a team; indirectly improving peers' motivation, ability to learn in the workplace, communication skills, all of which are believed to be vital to a game developer. However, this is affected by individual personalities; and low motivation levels, attitude and skill gaps remain key obstacles to learning. Scrum mitigates these by promoting autonomy and communication, but could further mitigate them by implementing a learning-oriented tailoring of the framework; supporting peer learning, greater intrinsic awareness, and discussion of learning goals and achievements.

Outstanding questions include how personality conflicts could be resolved, whether highly motivated or skilled individuals would better learn through team work or individual study, and/or whether further group workflow adaptations can accommodate all of these issues. However, the importance of collaboration and learning is well-known in industry. It is thus arguable that, through reflection and interaction, Agile philosophy by its nature is a good starting point in supporting the development of the developer.

References

[1] M. A. G. Darrin and W. S. Devereux, "The agile manifesto, design thinking and systems engineering," in *Systems Conference (SysCon), 2017 Annual IEEE International*, (Montreal, QC, Canada), pp. 1–5, Apr 2017.

[2] M. B. et al., "Manifesto for agile software development," 2001. Accessed 16th Nov 2017.

[3] L. J. Barker and K. Garvin-Doxas, "Why project courses sometimes widen the experience gap among students," in *Proceedings of the 8th annual conference on Innovation and technology in computer science education*, (Thessaloniki, Greece), pp. 258–258, Jul 2003.

[4] M. Q. Tran and R. Biddle, "Collaboration in serious game development: a case study," in *Future Play '08 Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, (Toronto, Ontario, Canada), pp. 49–56, Nov 2008.

[5] A. Kultima, "Developers' perspectives on iteration in game development," in *Proceedings of the 19th International Academic Mindtrek Conference*, (Tampere, Finland), pp. 26–32, Sep 2015.

[6] C. Polotowski, L. Fontoura, F. Petrillo, and Y.-G. Guhneuc, "Are the old days gone?: a survey on actual software engineering processes in video game industry," *GAS '16 Proceedings of the 5th International Workshop on Games and Software Engineering*, pp. 22–28, May 2016.

[7] J. Kasurinen, M. Palacin-Silva, and E. Vanhala, "What concerns game developers? a study on game development processes, sustainability and metrics," in *2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM)*, (Buenos Aires, Argentina), pp. 15–21, Jul 2017.

[8] L. J. Barker, "When do group projects widen the student experience gap?," in *Proceedings of the 10th annual SIGCSE conference on*

Innovation and technology in computer science education, (Caparica, Portugal), pp. 276–280, Jun 2005.

[9] E. L. Deci and R. M. Ryan, “Motivation, personality, and development within embedded social contexts: An overview of self-determination theory,” *The Oxford Handbook of Human Motivation*, pp. 85–107, Jan 2012.

[10] W. W. Royce, “Managing the development of large software systems: Concepts and techniques,” *ICSE ’87 Proceedings of the 9th international conference on Software Engineering*, 1987.

[11] K. Schwaber and M. Beedle, *Agile Software Development with Scrum.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 1 ed., 2001.

[12] S. Edmunds and G. Brown, “Effective small group learning: Amee guide no. 48,” *Medical Teacher*, vol. 32, pp. 715–726, Sep 2010.

[13] R. Helyer, “Learning through reflection: the critical role of reflection in work-based learning (wbl),” *Journal of Work-Applied Management*, vol. 7, pp. 15–27, 2015.

[14] C. A. Benware and E. L. Deci, “Quality of learning with an active versus passive motivational set,” *American Educational Research Journal*, vol. 21, pp. 755–765, Dec 1984.