How can agile philosophy maximise learning in an academic game development project?

COMP150 - Agile Development Practice

1707981

2017-11-16

Game development courses promise to equip students with the skills to develop videogames, now teaching beyond the specialist skills traditionally taught by institutions – such as art and programming – into territories that are typically learned within the work environment, including communication and team management. In practice, with a small team of students, a project and a scrum workflow, many issues arise on a personal basis when learning and adapting specialist skills whilst concurrently producing a project. However, with some minor changes the agile philosophy may be adapted to maximise the learning of these skills in a positive and motivated environment.

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 academic agile game production environment incorporating the scrum framework. While this is a deliberately academic context, significant overlap with game industry practice is expected; as learning, reflection and teamworking are widely acknowledged as vital skills [4, 5].

2 Why scrum?

One of scrum's characteristics is the scrum board, wherein team members choose the tasks they plan to do [6]. Ideally, choosing one's own tasks boosts motivation by promoting autonomy, which is a vital source of motivation [7], leading to deeper engagement with the team.

A project owner may be tempted to choose an agile workflow not to learn, but to improve the actual game. A finished game will often be very different to its initial design, due to its dependence on unpredictable qualities such as look and feel [4]; as a result, most game developers adopt a similarly iterative workflow as an essential part of the early process [8, 9, 5]. Evidence suggests that most game industry workflows are "agile" by nature, at least in the early stages [9]; although interestingly, game development companies often create their own personalised workflows, not self-proclaimed as "agile", and many use no particular workflow at all [5]. This suggests that workflows are popularly adapted to their environment, and when used, agile workflows should also be adapted where necessary.

3 Learning pitfalls

A paper by L. J. Barker and Kathy Garvin-Doxes [3] highlighted a startling concern that student group projects, despite their practicality, can in fact inhibit learning. With the pressure to deliver an equal work standard, being in a team of more experienced developers can lessen an inexperienced students' own desire to contribute.

Psychology and motivation are at the core of these issues. One influential study [7] 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 inexperienced team members, and can introduce the problem of 'unrealistic scope' often shared by the industry [10], wherein a project becomes too demanding for a team to reasonably deliver.

However, it is noteworthy that agile already attempts to address the above pitfalls through interaction [2], and that they would likely be worse in a conventional 'waterfall' workflow: where teams are typically fixed to a single plan for an extended period with minimal in-production reviews [11]. Contrary to the above, this lacks individual autonomy, is focused on an extrinsic goal (the plan), and does not accommodate self-reflection; whilst a strong focus on producing the work introduces the key stresses in the study.

4 Going upwards

4.1 Intrinsic goals

Boosting peer 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. 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 [7]; and perhaps more importantly, and give more knowledgeable team members an opportunity to help.

4.2 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, are typically more successful in their endeavours [12, 13]; similarly, reflection is already incorporated into most agile workflows for virtually the same reason [1, 6], though with a greater focus on the product. 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.

4.3 Peer learning

Peer learning should be encouraged team-wide [4, 7]. Firstly, it is actively practiced in industry, where openness to learning [5] and willingness to help others [4] are vital elements to a team member's personality. Furthermore, the act of knowledge sharing itself is known to have a positive effect not only on the learner, but the teacher as well [14, 7]. Encouraging the application of additional individual-oriented support, such as pair programming between peers of both similar or varied skill gaps, would help to emphasise the essential goal of sharing knowledge, and aid troubleshooting in situations where an unknown problem is hindering production [4, 7], particularly for the inexperienced.

Finally, [15] also found working in pairs to be a considerable solution to the skill gap.

5 Conclusion

The agile workflow highly accomodates learning in a group environment: improving peers' motivation, ability to learn in the workplace, communication skills, all of which are believed to be vital skills to a game developer's employability. However, this is impacted 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.

There remains much to explore in creating an ideal learning environment for aspiring game developers. While agile workflows have positive benefits to motivation and therefore learning, it remains inconclusive whether highly motivated individuals would better learn through group work or personal study, and/or whether this can be accommodated with further group workflow adaptations. However, the importance of collaboration and learning is well-established in industry, which the agile philosophy does well to accommodate.

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] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "What went wrong? a survey of problems in game development," *Computers in Entertainment (CIE) - SPECIAL ISSUE: Media Arts and Games*, vol. 7, Feb 2009.

[11] 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.

[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.

[15] 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.