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], described as a continous improvement based on reflection of previous results [2]. This virtually matches the definition of learning. However, agile philosophy states that working software is the primary measurement of progress [3], and appears to be focused on producing an ideal product in an ideally 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 [4].

This study aims to discover potential methods to improve the learning of production skills during the production process itself. It will attempt to improve learning and productivity 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 [5, 6].

2 Why scrum?

The hypothetical small team adopts the scrum agile framework. This framework is characterised by the scrum board, wherein team members choose their own tasks [7]. Choosing one's own tasks has proven benefits in both education and industry by promoting autonomy, which is a vital source of motivation [8], and explicitly expressing the group's individual tasks enhances learning. [9]

An educational environment may be tempted to choose an agile workflow not just for its learning benefits, but for its relevance in industry. A finished game will often be very different to its initial design, due to its dependence on unpredictable qualities such as look and feel [5]; as a result, most companies in industry adopt an iterative workflow as a vital part of the process [2, 10, 6]. Evidence suggests that most game industry workflows are "agile" by nature [10]; although interestingly, game development companies often create their own personalised workflows, not self-proclaimed as "agile", and many use no particular workflow at all [6]. This large diversity suggests that different workflows should suit different environments, furthering the possibility that the traditional scrum framework is suboptimal for an educational one.

3 Pitfalls

A paper by L. J. Barker and Kathy Garvin-Doxes from 2003 highlighted a startling concern that student group projects, despite their practicality, can in fact inhibit learning. In particular, being in a team of more experienced developers can lessen some inexperienced students' own desire to contribute. In some cases, the pressure of delivering an equal standard of work sours motivation, and students may drop out entirely [4]. Furthermore, in efforts to deliver high quality work, many students prefer to work in their "comfort zone", rather than expanding into areas they have yet to learn [4, 9]. The lack of learning is contrary to the interests of an educational institution, and in the game industry itself [5, 6].

These problems are centred not on the agile workflow, but psychology and motivation. One influential study [8] 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 present 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 itself [11], wherein a project becomes too demanding for a team to reasonably deliver.

It is however noteworthy that many of the above pitfalls exist far beyond game development, and would likely be aggravated by the adoption of the

more traditional waterfall workflow [12], wherein teams are typically fixed to a single plan for an extended period with minimal in-production reviews. This workflow lacks individual autonomy, is focused on an extrinsic goal (the plan), and doesn't promote reflection.

4 Going upwards

To combat these issues, a focus could be placed on promoting spirit and motivation. In the Scrum environment, awareness of the intrinsic goals could be gently raised by adding a 'what did I learn yesterday' clause to the stand-ups. This may overlap with 'what I did yesterday'; however, simply raising the intrinsic goal of learning itself could reproduce the positive results shown in the study [8]. It is, in fact, well-documented that groups who reflect on their learning process, rather than just the task, are typically more successful in their endeavours [13, 14]; similarly, reflection is already incorporated into most agile workflows for related reasons [1, 7], though with a greater focus on the product.

Reflection on learning may also spark discussion among peers about the learning topics in specific, and bring to attention other members also interested in learning it – providing the benefits ([5, 8, 9]) of peer learning in a fully autonomous format. This could be further augmented with a 'what do I want to learn today' clause, which would help an individual express their learning goals, and give more knowledgeable team members an opportunity to help.

In fact peer learning should be encouraged team-wide, owing to its several benefits. It is actively practiced in industry, where openness to learning [6] and willingness to help others [5] are vital elements to a team member's personality. Furthermore, the act of knowledge sharing itself is known to have a positive effect on both the teacher and learner [15, 8]. 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 [5, 8], particularly for the inexperienced.

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 proven as vital skills to a game developer's employability. However, this is impacted by vital areas of individual personality, and low motivation levels, attitude and skill gaps remain key obstacles to learning. Scrum mitigates these with promotion of autonomy and strong communication of work progress, but they could be further mitigated by implementing a

learning-oriented tailoring of the framework, adding focus to peer learning (such as pair programming), greater intrinsic awareness, and discussion of learning goals and achievements.

As a relatively new teaching topic, there is 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 comparatively whether highly motivated individuals will learn better specialist practices through group work or personal study, and/or whether this can be accommodated with further group workflow adaptations. However, the importance of learning through work is well-established by industry; and as agile philosophy suggests, that is perhaps the most important skill to gain.

6 Recycle bin

7 A comparison to industry

As hinted in the earlier findings, a survey conducted between 2014-2015 discovered that of the top required skills for hiring a new employee, 'ability to learn while working' [6] was the most prevalent, with 48% of game developers citing this in their Top 3 priorities.

Identifying technical mistakes is a challenge to those who are unaware, and is mostly achieved purely through trial and error [**?**].

From an organisational perspective, arranging groups into a smaller ranges of past experience is shown to be beneficial from a motivational standpoint. A study from 2009 [**?**] amusingly discovered that while students prefer to learn from the experienced, in a blind experiment they found reviews from lesser experienced team members to be more useful.[**?**] However, conversely, heterogeneous teams–teams varied in skill–are often seen among the most successful, but this depends on the team's ability to communicate. [4] Therefore a focus should be put on communication, something which is already learned during practical application of the agile philosophy. Being in a heterogenous groupalso helps when wntering the game industry as a junior[REVISE AND CITE].

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] 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.

[3] M. B. et al., "Manifesto for agile software development," 2001.

[4] 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.

[5] 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.

[6] J. K. et al., "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.

[7] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. 1 ed., 2001.

[8] E. L. Deci and R. M. Ryan, "Motivation, personality, and development within embedded social contexts: An overview of self-determination theory," pp. 85–107, Jan 2012.

[9] 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.

[10] C. P. et al., "Are the old days gone?: a survey on actual software engineering processes in video game industry," pp. 22–28, May 2016.

[11] F. P. et al., "What went wrong? a survey of problems in game development," vol. 7, Feb 2009.

[12] W. W. Royce, "Managing the development of large software systems: Concepts and techniques," 1987.

[13] S. Edmunds and G. Brown, "Effective small group learning: Amee guide no. 48," vol. 32, pp. 715–726, Sep 2010.

[14] R. Helyer, "Learning through reflection: the critical role of reflection in work-based learning (wbl)," vol. 7, pp. 15–27, 2015.

[15] C. A. Benware and E. L. Deci, "Quality of learning with an active versus passive motivational set," vol. 21, pp. 755–765, Dec 1984.