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

COMP150 - Agile Development Practice

1707981

2017-11-16

Today game-centric courses promise to equip students with the skills to develop videogames. These extend beyond the specialist skills traditionally taught by institutions–such as art and programming–into territories that are typically learned through the work environment, including communication and team management. Practical application of those skills is established with a team of students, a project and a workflow; but workflows are designed for the product, and it is unclear how it can be used to effectively teach the specialist skills. This essay specifically explores the agile philosophy, and how it can be adapted to maximise the learning of these skills.

1 Introduction

The key principle in education is learning. A key principle in the Agile philosophy is iteration [1]. Iteration can be described as a continous improvement based on reflection of previous results. [2] However, iteration in the context of agile development is focused on the goal of producing an ideal product in an ideal timeframe, stating that working software is the primary measure of progress. [3], contrast to particularly learning how to do it well and with good practices.

While the agile manifesto is promising overall with regard to production and teamwork, this study is interested in discovering how learning about production can best be incorporated into production itself. Its goal is to explore potential methods and techniques to improve opportunities to learn effectively during agile game production. Specifically, we will be critically approaching these three questions:

* What benefits does Agile, ideally, offer to learners?
* What compromising issues arise in learning through group project work?
* How can, or should the Agile workflow be improved to further promote the learning of new skills?

2 Agile

Agile software development is summarised by the following philosophies as described in the Agile manifesto:

1 *Individuals and interactions over process and tools*,

2 *Working software over comprehensive documentation*,

3 *Customer collaboration over contract negotiation*

4 *Responding to change over following a plan* [1]

These principles are driven by one of various workflow frameworks, such as Scrum, wherein a team will stand and meet up at a particular time in a working day and state what they have done, will do, and what may block them from doing it. Tasks themselves are written in on a task board by the Scrum Master in an order of priority, and the team members select the task they are prepared to do. By the end of the meeting all members are aware of everyone's roles for the day, this keeps everyone on track and motivated etc [CITE]

Ideally, this links well with the concept of education. Allowing team members to choose their own tasks promotes autonomy, which is a vital source of motivation. [4] By prioritising interactions and response to change, particularly as games are rarely expected to go as planned[CITE], peer learning is promoted between team members. Teaching and learning in small [amounts] amongst peers is a measurable practice in the game industry, where openness to learning and willingness to help others [5] are vital elements to a team member's personality. Types of peer learning include pair programming, often applied in educational institutions[CITE] and industry itself[CITE], wherein one programmer takes the keyboard and another observes, offering tips and pointing out potential errors, before the two swap later on.

Iteration by nature is a highly respected concept in the game industry, owing to the fun factor, playtesting, and reviews continuously impacting the direction the project takes. Some even consider the iterative process to be enjoyable[2].

Agile offers many benefits to group work, but not without its pitfalls. However it is arguable that many of the issues are common far beyond game development, and that they would be further aggravated by the adoption of a more traditional waterfall workflow.

Teamwork, communication, collaboration etc Masters of these skills are highly desired by game and software industry employers.[cite]

3 Group work

A paper by L. J. Barker and Kathy Garvin-Doxes from 2003 [6] highlighted a startling concern that group projects, despite their practicality, can in fact

inhibit personal learning. In particular, being in a team of more experienced developers can lessen some inexperienced students' own desire to contribute. In some cases, the burden of delivering an equal standard of work can sour motivation, and students drop out entirely [6].

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.[6] [7] The lack of learning is contrary to the interests of an educational institution, and in fact the game industry itself, where learning is considered a vital skill [5], [8].

These problems are centred not strictly on the workflow, but psychology and motivation. A study detailed in [4] observed that students 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. A high quality game is an extrinsic goal, which may risk diluting the focus of learning and teaching the skills, and redirecting it to the game's quality. This puts pressure on students, especially the inexperienced, and a common issue in the game industry manifests–unrealistic scope [**?**], wherein a project is too demanding for a team to reasonably deliver on time.

To combat these issues, perhaps the biggest focus should be on improving motivation. In a hypothetical Scrum environment, awareness of the intrinsic goals, as described in [4], could be applied by adding a 'what did I learn yesterday' part to the stand-ups. This may perhaps already overlap too much with 'what did I create yesterday', but simply raising the intrinsic goal itself could result in the same positive results as it did in the study [4]. 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 [9]. This extra awareness may also spark voluntary discussion among peers about the newly learned material in specific, and bring to attention others who are also interested in learning it–providing the benefits of peer learning and autonomy at the same time.

The issue of less inexperienced team members finding their place in the project still requires additional attention. The application of additional individual-oriented support among peers, such as pair programming, would help to emphasise the essential goal of sharing knowledge, and troubleshoot in situations where an unknown problem is hindering development [4]. [CITE] To promote this, another particular addition could be made to the stand-ups–'what do I want to learn today'. This would briefly shift focus from the project to the individual, and gives more knowledgeable students to an opportunity to autonomously volunteer to help.

Despite a burden of teaching, such opportunity to teach others may be taken more often than one would expect. Active teaching is shown to make the teacher feel better about themselves [**?**]

"Only when group processes are made explicit can group activities can lead to enhanced learning" [7] Scrum promotes a healthy combination of

tasksetting and autonomy

Learning doesn't stop at university [CITE], learning is ongoing etc [CITE]. A survey conducted between 2014-2015 discovered that of the top required skills for hiring a new employee, 'ability to learn while working' [8] was the most prevalent, with 48% of game developers citing this in their Top 3 priorities.

Furthermore, despite the extrinsic goal's negative reputation so far, the game project itself has can often bring teams together by its ambition, uniqueness, etc[CITE]. This [puts Agile in a good place] where it says 'is the primary indicator of progress'.

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, heterogenous teams–teams varied in skill–are often seen among the most successful, but this depends on the team's ability to communicate. [6] 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].

4 Conclusion

The agile workflow highly accomodates learning in a group environment and has several strengths, improving peers' employability, ability to learn in the workplace and communication skills. However, this is subject to vital elements of human psychology, particularly motivation and attitude, as well as skill gaps which can create tensions in the workplace. Agile philosophy mitigates these through the promotion of autonomy and strong communication of work progress, but these effects could, dependent on personalities, be further mitigated by implementing a slightly more education-specific tailoring of the Scrum framework, promoting application of pair programming, and accomodating a self-critical focus on bettering oneself throughout the process.

As a relatively new teaching area, there is much to be explored in the making of an ideal learning environment for aspiring game developers. Outstanding questions particularly include whether there are better, more creative ways to further benefit the learning environment. There are many possible answers and no single solution. Regardless it stands that agile student team workflows will continue to improve over time through further iteration and self-reflection–very much agile in themselves.

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."

[4] 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.

[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] 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.

[7] 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.

[8] 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.

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