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 product. However, with some minor changes the agile philosophy could be adapted to maximise the learning of these skills in a positive and motivated environment.

1 Introduction

The key principle in education is learning, whilst a key principle in the Agile philosophy is iteration [1]. Iteration is described as a continous improvement based on reflection of previous results [2]. Such reflection is hugely beneficial to learning new skills in a personal context [**?**] [CITEMORE?]; however, iteration in the context of agile development is focused on producing an ideal product in an ideal timeframe, stating that working software is the primary measure of progress, [3] contrast to learning how to do it with good practice. This can raise several issues in a traditional group environment, particularly where skill gaps arise [4].

This study is interested in discovering possible solutions to incorporating the learning of production skills into production itself. Its goal is to explore potential methods and techniques to improve learning, motivation and productivity in a hypothetical agile game production environment incorporating the scrum framework–this will be the primary context for the remainder of the study. Specifically, it will be critically address 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's suitability

Agile software development, contrast with a traditional waterfall approach, makes several promises of key interest. Namely it promises a healthy, motivated and continually improving work environment.

Of particular interest is the scrum framework–an agile framework characterised by the scrum board, wherein team members choose their own task [**?**]. Choosing one's own tasks has proven benefits in both education and industry by promoting autonomy, which is a vital source of motivation. [5]

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 heapings between peers is a measurable practice in the game industry, where openness to learning and willingness to help others [6] 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. This adaptive peer-learning approach can accomodate areas where some individuals don't know what to do[CITE].

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, although fails to address some key pitfalls. However it is arguable that many of the pitfalls are common far beyond game development itself, and that they would be further aggravated by the adoption of a more traditional waterfall workflow[CITE], wherein teams are typically fixed to a single plan for an extended period with little room for reflection. This lacks autonomy, reflection and is focused on an extrinsic goal–the plan.[5]

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

3 Addressing issues

A paper by L. J. Barker and Kathy Garvin-Doxes from 2003 [7] 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 [7].

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.[7] [4] 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 [6], [8].

These problems are centred not strictly on the workflow, but psychology and motivation. A study detailed in [5] 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 [5], 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 [5]. 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 [5]. [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 [10]

"Only when group processes are made explicit can group activities can lead to enhanced learning" [4] 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. [7] 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 with several strengths; improving peers' motivation, ability to learn in the workplace, communication skills, all of which are proven to be 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 can create tensions in the team. Agile philosophy mitigates these through the promotion of autonomy and strong communication of work progress, but they could, dependent on personalities, be further mitigated by implementing a learning-oriented tailoring of the Scrum framework, promoting more peer learning, more awareness and discussion of learning goals and achievements, and deliberate application of pair programming.

As a relatively new teaching area, there is much to explore in creating 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, and it is worth recommending that agile student team workflows continue to be improved over time through iteration and self-reflection for some cool meta agile shoite.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[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.

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