Challenges when scaling the Agile Methodology and how to Overcome them

COMP150 - Agile Development Practice

1604281

2017-11-16

The Agile Software Development methodology was initially designed to be used in small teams that could quickly develop products with flexible requirements. As Agile practices have gained popularity in the software industry, companies have found the need to employ Agile methods in larger teams and on more complex projects. This paper will be examining some of the challenges associated with up-scaling the Agile methodology and discussing some of the solutions that are used in industry.

1 Introduction

One of the most common criticisms of the agile methodology is the difficulties encountered when up-scaling the size of Agile projects. [1]

Although this paper is addressing general Agile practices and problems, the Agile methodology is also used in the games industry [2] which can range from small independent studios to massive teams working on multiple pieces of software. It has even been shown that many games development companies who do not intent to use Agile development still implement many of its core practices [3] and as such, the potential to scale Agile development to suit the nature of the game development environment could be beneficial to the games industry as a whole.

2 Challenges

Though there are many issues involved in scaling an Agile project [4], the two that will be addressed in this paper are Team Size and Project Complexity.

Project Complexity

As a project becomes larger and more complex, several issues arise when approaching it with the Agile methodology. Firstly, the Agile principle of "Welcome changing requirements, even late in development." [5], while feasible to implement in a small project, becomes increasingly difficult as the software becomes more complex. For example, if a project has many constituent parts which each have many dependencies on each other, changing one aspect of the project late in development may cause problems in other parts of the project. This can be compounded by another aspect of project complexity - mixing of Agile and Non-Agile teams. [6]

A large project will inevitably have multiple teams working on different aspects of it concurrently, often leading to both Agile and Non-Agile groups collaborating. This can cause clashes in design approach and lead to some groups not fully understanding the design of the project.

Team Size

Similar issues exist regarding larger team sizes, particularly when regarding communication. Communication is key to any development project but it is given special focus in an agile context. [7] As teams become larger, usually when consisting of over fifteen people, [8] communication within the team can break down, as Scrum meetings become unwieldy and long winded, wasting time that could be spent working.

3 Solutions

There have been numerous proposed solutions to the difficulties of scaling Agile, however this paper will be focusing on just three of them. A brief outline of each solution has been provided below.

Scrum of Scrums

The Scrum of Scrums is a common method for dealing with large Agile teams which enhances the standard Scrum model. Large teams are divided into smaller ones and an Ambassador is appointed for each. Individual teams conduct normal Scrum meeting, while the Ambassadors meet for a Scrum of Scrums where they relay important issues discussed in their teams' scrums.

Lean Governance

Lean Development Governance is a governance method proposed to create a stable framework to allow Agile, Lean or similar software development methods to be implemented into a company or department.

See this paper for details [9].

Multiple Backlogs

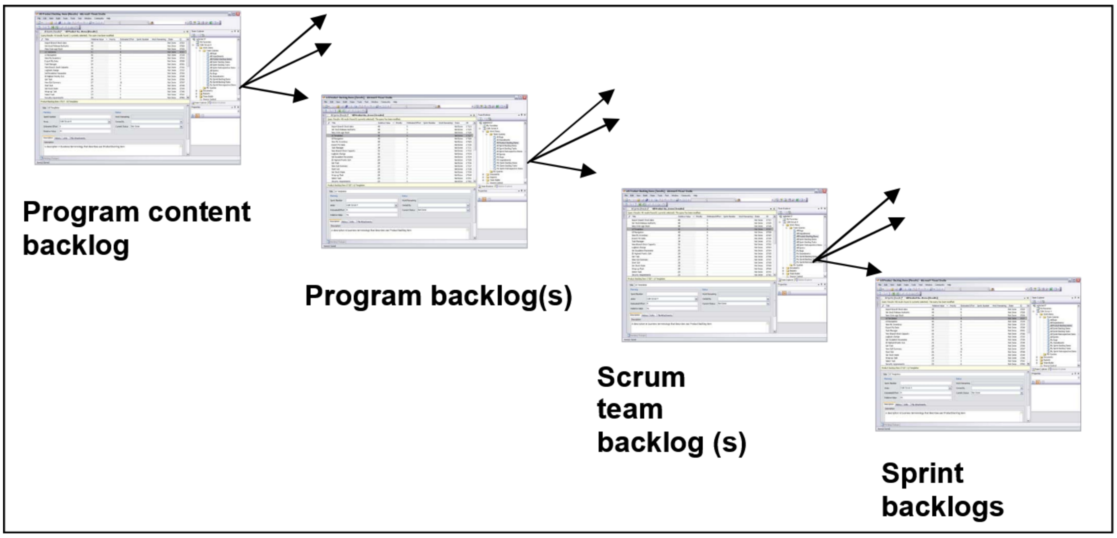

Figure 1: Backlog Structure: Taken from [10, p. 1384]

The Nokia Corporation took an interesting approach on one of their projects, utilising four different types of backlog to organise tasks across multiple teams.

As shown in Figure 1, the system they developed consisted of two program level backlogs and two at team level. See this paper for details [10].

4 Analysis and Comparison

The widespread usage of Scrum of Scrums suggests that it becomes a crucial element of Agile Development as soon as team sizes start increasing as it facilitates constant communication without meetings becoming long and cumbersome.

Lean Governance itself is not specifically tailored to improving scalability of Agile development but many of the principles of Lean Software Development overlap or mesh well with Agile. In addition to this, it has been suggested that many of the practices suggested by Lean Governance provide opportunity to employ Agile at a large scale. [11] One practice of Lean Governance that applies strongly to Agile development is Staged Program Delivery as it ensures constant delivery of working software, even if it lacks some features.

Nokia's approach was unique and provided a simple way to separate tasks between teams. However, while this method appears easy to set up, it is important to note that Nokia had many veteran program managers who were able to coordinate between teams and to help keep a long term goal in mind when planning sprints. Nokia also had problems with undisciplined practices, such as Scrum Masters acting as managers. This supports the idea that the effectiveness of Agile development is strongly affected by the presence of supportive management practices. [12] This reinforces the suggestion that a Lean Governance framework is crucial to providing the basis for Agile at Scale, as it promotes governance that supports the Agile values. Despite this, implementing Lean Governance is a major task that requires training management staff and reorganising company work-flow and therefore may not be suitable for all companies.

Even with these proposed solutions, it is likely that compromises will have to be made. [6] Allowing for some initial design work to outline the long term vision of the project can be useful when working with Agile and Non-Agile teams as it provides something permanent to work off, while still allowing room for iterative design. This can also be of use in complex projects where the finished product may be years away, as it keeps the developers focused on the final goals. This may be necessary in situations where a company does not have the experienced managers needed to keep a project on track, as Nokia did.

5 Conclusion

Despite the challenges faced when scaling up the Agile Methodology, it has been proved feasible [13] and many solutions exist to overcome the issues. Whatever method is employed when scaling, it is critical to address communication issues due to their potentially widespread impact on a project. Scrum of Scrums is an essential starting point if large teams are used, as it keeps teams in contact while requiring only a single extra meeting session for the ambassadors. This may also be of particular importance in the Games Industry, where design choices heavily effect mechanics programming, narrative writing and art-style.

Communication aside, the exact method of scaling will have to be tailored for the individual company, project or team, as different requirements may exist.

References

[1] A. Begel and N. Nagappan, "Usage and perceptions of agile software development in an industrial context: An exploratory study," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on.* IEEE, 2007, pp. 255–264.

[2] C. Politowski, L. Fontoura, F. Petrillo, and Y.-G. Guéhéneuc, "Are the old days gone? a survey on actual software engineering processes in video game industry," in *Games and Software Engineering (GAS), 2016 IEEE/ACM 5th International Workshop on.* IEEE, 2016, pp. 22–28.

[3] F. Petrillo and M. Pimenta, "Is agility out there?: agile practices in game development," in *Proceedings of the 28th ACM International Conference on Design of Communication.* ACM, 2010, pp. 9–15.

[4] D. Turk, R. France, and B. Rumpe, "Limitations of agile software processes," *arXiv preprint arXiv:1409.6600*, 2014.

[5] A. v. B. Kent Beck, Mike Beedle, “Agile Manifesto principles of agile,” http://agilemanifesto.org/iso/en/manifesto.html.

[6] D. J. Reifer, F. Maurer, and H. Erdogmus, “Scaling agile methods,” *IEEE software*, vol. 20, no. 4, pp. 12–14, 2003.

[7] M. Stoica, M. Mircea, and B. Ghilic-Micu, “Software development: Agile vs. traditional,” *Informatica Economica*, vol. 17, no. 4, p. 64, 2013.

[8] S. W. Ambler, “The agile scaling model (asm): adapting agile methods for complex environments,” *Environments*, 2009.

[9] S. Ambler and P. Kroll, “Applying agile and lean principles to the governance of software and systems development,” *IBM Software Rational*, 2009.

[10] M. Laanti, “Implementing program model with agile principles in a large software development organization,” in *Computer Software and Applications, 2008. COMPSAC’08. 32nd Annual IEEE International*. IEEE, 2008, pp. 1383–1391.

[11] S. W. Ambler, “Scaling agile software development through lean governance,” in *Proceedings of the 2009 ICSE workshop on software development governance*. IEEE Computer Society, 2009, pp. 1–2.

[12] L. M. Maruping, V. Venkatesh, and R. Agarwal, “A control theory perspective on agile methodology use and changing user requirements,” *Information Systems Research*, vol. 20, no. 3, pp. 377–399, 2009.

[13] S. Bick, A. Scheerer, and K. Spohrer, “Inter-team coordination in large agile software development settings: Five ways of practicing agile at scale,” in *Proceedings of the Scientific Workshop Proceedings of XP2016*. ACM, 2016, p. 4.