Principles of Distributed Agile Development

Aditya Samant
ansamant@asu.edu
Arizona State University

Sanay Devi
svdevi@asu.edu
Arizona State University

Abstract

Agile teams embrace change, companies today have agile teams running the software development process. Inner-team and cross-team communication is critical in the development of software. Team effectiveness depends upon trust between the small agile teams not only within the company environment but also between teams across different geographical locations, time zones and cultures [3]. As John Maxwell said "Team work makes the dream work", this same principle applies for agile teams within an organization. The teams should be motivated, aware and informative throughout the software development process.

ACM Reference Format:
Aditya Samant and Sanay Devi. 2019. Principles of Distributed Agile Development. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Agile teams are most effective in small projects (<50 people) who have easy access to user, business experts and are developing non-safety critical projects [2] .However, to meet requirements for large scale projects a globalized approach with outsourced teams is becoming common [1],where different teams must work together cohesively, flexibly and rapidly to deliver a product that conforms to requirements specifications. In order for small team based agile practices to be effective on a larger scale, careful consideration must be given to the collaboration between teams and the overall architecture. Section 2 is entirely dedicated to highlight the various facets in which teamwork affects large scale agile projects. Subsection 2.1 informs the reader which scrum approach of the three scrum approaches [1] is the best to use in a large scale scrum project, subsection 2.2 highlights the challenges in inter team communications over large scale projects and subsection 2.3 highlights how architecture and design affects teams in large projects.

2 Team Work

Most companies today consist of project teams ranging from two to several hundred people. These teams are further split into groups and each group is then responsible for developing a part of the system. The teams can be grouped into three categories. Isolated Scrum Teams, Distributed Scrum Of Scrums and Totally Integrated Scrums [1]. The isolated teams are usually on site teams and communication between them is usually by in-person meetings and "elevator meetings".The main communication problem occurs when the teams are distributed overseas since teams will portray differences in work styles, in the worst case outsourced teams may not use Scrum and be productive using the waterfall approach. This issue needs to be addressed, and all teams should in fact use the same methodology and stick to it.
To form a group of professional individuals, the critical management task is to find a right balance of technical skills, experience and personalities. However, for a team to be actually productive and successful, the team should be cohesive and have team spirit [8]. Each team member should motivate others and be loyal to each other. When problems arise or when sudden changes are required, the group as a whole and not individually should be able to adapt to the changes and overcome the problems.
The idea of teamwork encapsulates a set of values that encourage listening and responding constructively to views expressed by others, giving others the benefit of the doubt, providing support, and recognizing the interests and achievements of others [6]. The subsequent subsections highlight what is important when Agile Teams have to work together.

2.1 Scrum Approach

Jeff Sutherland et al. in 2007, conducted a case study and provided three different approaches to conduct Scrum in large scaled processes of which Isolated Scrum methodology was considered the least efficient since the teams had no mechanism for regular communications, the other methodologies were viewed favorably by Sutherland and folk and largely that was due to the presence of regularly scheduled Scrum of Scrums meetings, where a Chief Scrum Master and Chief Product Owner will discuss the overall shape of the project with the individual team leads [1]. The scrum of scrums meeting greatly increases the efficiency of the project and is considered by the authors to be the second best scrum approach for distributed development.However, it is still a bit inefficient when trying to create a proper architecture that is workable with parallel teamwork [2].The best approach for distributed development which further increases communications between the teams is by dispersing team members through different sites. By doing so team members visiting the sites are exposed to a different culture and work environment and thus get a better understanding of the project as a whole.

Conference'17, July 2017, Washington, DC, USA
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.2 Team Communication

Inter-team communication is one the six issues in distributed development [1], since a failure to communicate properly with teammates will lead to production delays and conflicts in code implementation. Traditional agile methodology emphasizes self-management, in a small team a team member is cross-functional and responsible for every phase in the agile development of the project [2]. Due to the eclectic nature of modern large-scale projects the teams can be spread across time zones, cultures and languages. Many a times to make a project within the financial, quality and time constraints provided by the client, companies will outsource portions of the project to other companies.

Bjornson et al. studied multiple large scale agile projects and determined that a shared mental model, closed-looped communication and trust are the essential mechanisms through which a project can succeed [4].

These mechanisms are interrelated and as such without effective communication mechanisms the building of a shared mental model and the fostering trust within teams is less likely to take place. Dingsoyr and Moe in 2014 determined that one way to foster a strong communication and understanding between different teams is by conducting inter team workshops, the goal of the workshop is to understand each other's work culture and to develop a shared team language. Another key means through which a shared team language and cohesion can increase is having the individuals have a "knowledge network" to be aware of fields and area beyond the scope of their team which will aide in the self-management of the individual teammates[2].

2.3 Agile and Architecture

Architectural design can easily become problematic in large scale agile projects due to multiple different teams working in parallel on different sections of the product [2]. Many agile teams favor code refactoring instead of sound architectural design decrying it as big architecture upfront [7]. However, in large scale systems refactoring is a poor substitute for sound software and system architecture due to the frequent changes agile is prone to make throughout its lifecycle. Conversely, good architectural design enhances agility in large projects. A large-scale agile process with an early architecture focus defines the implementation structure that divides the organization into small teams focusing on either the infrastructure or on the independent features themselves[7]. A good software and system architecture also allows for the creation of a common vocabulary and common culture, a guide for release planning and configuration management and many other benefits [7]. Architects focus on three separate concerns for agile software development: Architecture of the system, the structure of the development and production infrastructure [2]. For agility to remain over time in a large scale project these three concerns need to be kept aligned over the course of time. Nord et al maintains that architectural requirements must be drawn along with functional and non-functional requirements with respect to changing needs. Architectural tactics such as vertical and horizontal decomposition (which deal with task distribution), matrix and augmented team structures, architectural runways (for better maintainability) and deployability tactics provide a healthy framework to maintain the alignment of the three concerns [7].

3 Conclusion

Individuals learn more and learn better working together rather than in isolation or competing against each other. Being creative and innovative are important factors for teamwork along with having a common vocabulary and work culture. When the teams together can solve problems quickly and effectively, the entire development process benefits from it. The performance of teams is also dependent on how and where each team is placed in the organization. Proper placement of teams is driven by good system and software architectural designs. Being encouraged and accepted within the organization will motivate the individual in teams to perform better and lead to more creative innovation. Therefore, a collaborative work environment with the correct organizational context, is the base on which companies are built, it enhances the team performance and caters to the overall success of a company.

References

[1] Jeff Sutherland Anton Viktorov, Jack Blount and Nikolai Puntikov. 2007. Distributed Scrum: Agile Project Management with Outsourced Development Teams. *Proceedings of the 40th Hawaii International Conference on System Sciences - 2007* 1, 1 (2007), 10.

[2] Torgeir Dingsoyr and Nils Brede Moe. 2014. Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda. *Research Paper* 1, 1 (2014), 6.

[3] Siva Dorairaj and James Noble. 2013. Agile Software Development with Distributed Teams Agility, Distribution and Trust. *IEEE Conference* 6, 5 (2013), 13.

[4] Christoph Johann Stettina Finn Olav Bjornson, Julia Wijnmaalen and Torgeir Dingsoyr. 2018. Inter-team Coordination in Large-Scale Agile Development: A Case Study of Three Enabling Mechanisms. *publication 325188087* 1, 1 (2018), 16.

[5] Oystein Gutu Gautam Ghosh. 2004. Agile, Multidisciplinary Teamwork. *Agile Business Conference* 5, 4 (2004), 10.

[6] Nils Brede Moe, Torgeir Dingsoyr, and Tore Dybaa. 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. *Elsevier B.V.10.1016/j.infsof.2009.11.004* 52 (5 2010), 480–491.

[7] Ipek Ozkaya Robert L. Nord and Philippe Krutchen. 2014. Agile in Distress: Architecture to the Rescue. *Conference Paper 2014021001446847* 1, 1 (2014), 15.

[8] Ian Sommervile. 2009. *Software Engineering* (9 ed.). Addison-Wesley, 4.