

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. The communication within the particular team is of utmost importance however how the agile team communicates with other agile teams is one of the most critical parts of developing a 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.

Keywords datasets, neural networks, gaze detection, text tagging

ACM Reference Format:

Aditya Samant and Sanay Devi. 2018. Principles of Distributed Agile Development. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/1122445.1122456>

1 Introduction

Agile teams are most effective in small projects (<50 people) who have "easy access to user and 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 in larger scale careful consideration must be given to the collaboration between teams and overall architecture. The following section 2.1 in Main side will cover the information about inter-team collaboration and

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Woodstock '18, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9999-9/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

section 2.2 will cover the importance of architectural design and architects in large-scale agile 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 viz. 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 group to be actually productive and successful, the group should be cohesive and have team spirit[7]. Each group member should motivate others and should 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 Selecting Team Members

The main responsibility of the manager of a project is to create a cohesive group of teams and organize them so they can work together effectively. Companies due to budget constraints and lack of specific skills, outsource teams from different regions and managers do not have the complete say in the process of team selection. The problem arises when individuals are motivated by their own work and who have their own ideas about how technical problems should be solved. The manager should select teams and their members based on having similar complementary personalities and not rely solely on technical capabilities. Individuals who are interaction-oriented help facilitate communication within the team and are the key in holding the sanity of the entire

team together[7].

When building a team, each participant's potential contribution to the process has to be evaluated. The ideal team member is a part of the team and identifies himself/herself with the team's goals, represents colleagues and has the time to participate through the entire process[5].

The key factors to select the team members are education, training, application domain, technology experience, communication ability, adaptability and problem solving skills[7].

2.2 Team Organization

The way the team is organized affects decision making process, the way in which information is exchanged and the interaction between the development teams and the project stakeholders. For a big project with many interconnected teams a system architect is required who is usually a senior engineer. The critical decisions should be made by the system architect, project manager and all the team members. To make the most effective use of skilled programmers, the teams should be built around an individual chief highly skilled programmer. In the end, all these teams should work together in harmony to put forth a deliverable product.

2.2.1 Plan Together

The teams should work together towards a common vision and road map, and collaborate on ways to achieve the objectives. All Agile Teams should participate in a common approach to estimating work. This provides a meaningful way to help decision-making authorities guide the course of action based on economics.

2.3 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 [2], in a small team a team member is "cross-functional" and responsible for every phase in the agile development of the project. 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. Bjørnson 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. Dingsøyr 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"[2]. 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].

2.4 Team Work and Scrum

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 through scrum is the Integrated Scrum approach which incorporates the group scrum team approach of the Scrum of Scrums approach but further increases communications between the teams 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.

Another reason why the Integrated approach used in the case study was regarded the best was the strong planning and strong authority of the central command in the project. There were tools for collaborations, protocols for Configuration Management, Use Cases and Testing [1]. The strong central command added a bit of certainty to the interactions between the teams as everyone was following the same agreed upon procedures and using the same technology. This prevented needless conflicts arising from contradicting tech stacks or testing protocols as may have taken place in Isolated Scrum.

2.5 Visualizing and documenting the team's efforts throughout the process: Design and Architecture

Acknowledgments

References

- [1] Jeff Sutherland Anton Viktorov, Jack Blount and Nikolai Puntikov. 2007. Distributed Scrum: Agile Project Management with Outsourced Development Teams. (2007).
- [2] Torgeir Dingsøyr and Nils Brede Moe. 2014. Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda. (2014), 6. https://brage.bibsys.no/xmlui/bitstream/handle/11250/2472769/Towards+principles+of+large-scale+agile+development_postprint.pdf?sequence=2

- [3] Siva Dorairaj and James Noble. 2013. Agile Software Development with Distributed Teams: Agility, Distribution and Trust. (2013), 10. <https://ieeexplore-ieee-org.ezproxy1.lib.asu.edu/stamp/stamp.jsp?tp=&arnumber=6612873&tag=1>
- [4] Christoph Johann Stettina Finn Olav Björnsen, Julia Wijnmaalen and Torgeir Dings. 2018. Inter-team Coordination in Large-Scale Agile Development: A Case Study of Three Enabling Mechanisms. (2018), 16. https://www.researchgate.net/publication/325188087_Inter-team_Coordination_in_Large-Scale_Agile_Development_A_Case_Study_of_Three_Enabling_Mechanisms
- [5] Oystein Gutu Gautam Ghosh. 2004. Agile, Multidisciplinary Teamwork. (2004). <http://www.methodsandtools.com/archive/archive.php?id=17>
- [6] Nils Brede Moe, Torgeir Dings, and Tore Dybå. 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. 52 (5 2010), 480–491. Issue 5. <https://doi.org/10.1016/j.infsof.2009.11.004>
- [7] Ian Sommerville. 2009. *Software Engineering* (9 ed.). Addison-Wesley.