Agile Teams Orchestration

Working together with Agile Teams

Aneesh Dalvi
CIDSE
Arizona State University
Tempe, Arizona, USA
adalvi1@asu.edu

Gangadhara Matti
CIDSE
Arizona State University
Tempe, Arizona
gmatti1@asu.edu

ABSTRACT

Traditionally, the software was developed as a whole in linear sequential phases with a fixed set of requirements where each phase depends on the deliverables of the previous phases. However, the requirements for new-age software changed frequently and needed a continuous delivery with a customer-centric approach. This led to an introduction to 'Agile methodology' where the requirements and solutions evolve iteratively through collaboration between self-organizing cross-functional teams. As per the Agile Manifesto, this approach is efficient for a small co-located team (less than 5-15 people). Effective coordination between teams and issues with design and architecture is of major concern with an increase in the complexity of software projects and teams.

In this paper, we discuss a solution to these problems. By following the same principle of agile, we could divide this large team into groups of smaller agile teams which is called Scrum of Scrums. Care must be taken to effectively manage communication and coordination between the teams, which is discussed in sections 2, 3 and 4. Design and Architecture is another point of concern, which may easily get complicated and can lead to failure of the project, which is discussed in section 5. It is at most important to perform project management in the best possible way for the smooth transition of the large-scale project.

CCS CONCEPTS

• Software Engineering • Software Development Life Cycle • Agile Methodology • Scrum • Scrum of Scrums • Design and Architecture

KEYWORDS

Agile, Teams Orchestration, Scrum of Scrums, Teamwork, Team Communication, Design and Architecture, Team coordination.

ACM Reference format:

Aneesh Dalvi, Gangadhara Matti. 2020. Agile Teams Orchestration: Working together in agile teams. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 3 pages.* https://doi.org/10.1145/1234567890

1 INTRODUCTION

In the past two decades, we noticed the tremendous expansion of the internet, which led to the growth of software companies from small-scale to large-scale. In this information age, due to enormous competition in the market, changes are frequent to any software project under development. Conventionally, the entire software was developed by a small team. However, with an increase in the complexity and requirements, led to an increase in people involved in building software. As a result of this, the team is responsible for developing a single module, which is a part of the large software. Agile works well when applied on smaller teams as it gives more flexibility, transparency and coordination between the team members. [1] Although, to meet the requirements for large scale projects a globalized approach with outsourced teams is becoming common, where different teams must work together cohesively flexibly and rapidly to deliver a product that conforms to [2] the requirement specification. For agile principles to be effective for larger teams, careful considerations must be given to the collaboration between the teams, overall design and architecture, the project falls on track, resolving dependencies and duplicity of work. [3] In section 2 we discuss various issues of teamwork that should be considered while working on large scale. In section 3 we discuss the scrum approach that should be followed for large scale development. Section 4 addresses the challenges in inter-team communication over large scale projects. Section 5 highlights how to design and architectural effects in large projects.

2 TEAM WORK

As Helen Keller said, "Alone we can do so little, together we can do so much". [4] Teamwork is the driving force to success for any project. Ideally agile is effective for team size ranging from 5-15 members. But in large-scale projects teams consists of a single scrum to several hundred people. [5] These large groups of people are divided into smaller teams. A dedicated module of the system is developed by each small team. These groups can be categorized

[*] Article Title Footnote needs to be captured as Title Note
[†] Author Footnote to be captured as Author Note

WOODSTOCK'18, June, 2018, El Paso, Texas USA

into scrum teams, distributed scrum of scrums and totally integrated scrums. Scrums team is usually a collection of motivated self-organizing cross-functional teams (typically 5-15 members) working together cohesively to deliver the product increments. The communication between the scrum teams is mainly face to face meetings. The main problem arises when the team is distributed globally. The communication issues include differences in work style, time zone and cultural differences. As the teams are distributed, the teams may not use an agile approach. [2] Every team needs to follow the same methodology in order to minimize the issues addressed in communication between teams.

A productive successful scrum team needs to have the right balance between technical skills and experience. Every project relies on good teams. Effective teamwork helps drive the project towards success. Team players should be committed to the goal, being reliable and responsible, support and respect others, be an active listener and a good problem solver. [6] The next section highlights the necessary scrum approach for large scale projects.

3 PROPOSED APPROACH

Jeff Sutherland in 2007 proposed that out of the three approaches discussed above scrum teams and totally integrated scrum are less efficient than the distributed scrum approach for large scale projects because this is because teams have no way for regular communication. [2] Sutherland and Folk favored the distributed Scrum approach for large scale because of regularly scheduled scrum of scrum meetings. Scrum of scrum is a method to use scrum for large scale teams. [7] The group of people are divided into size of 5 to 15, an ambassador is elected for every team with who participates in the daily meetings with other ambassadors of other teams called scrum of scrums. Scrum master in each team takes part in scrum of scrums. A product owner showcases the work across all the team. Team who is responsible for deliverable should attend this meeting. This approach is slightly inefficient when creating architecture while working in parallel teams.

4 TEAM COMMUNICATION

One of the main issues in distributed development is the inter-team communication. Lack of communication in team, generates a lot of problems. Failure to communicate with team can create delays in production and several code conflicts. [6] Conventional Agile methodology emphasis self-organizing highly skilled cross functional team members in a team who take care of all the phases in the software development cycle. However, as the size of the project increases, the teams need to face the issues of time zones, languages and culture. [1] The teams can also be distributed across the globe where companies outsource the team to work on a particular module of the large project.

Bjornson always believed that the shared mental model, closed looped communication and trust among the team members are the most required techniques for a project to become successful. These techniques are co-related, so to have them in a distributed environment is difficult. Dingsoyr and Moe stated that the communication between teams can be better if several inter team workshops are conducted. The end goal of these workshop would be to get used to each other's work culture and develop new methods to communicate with each other. [8] There is one more way to increase the communication between teams by building knowledge which is beyond the team scope. This aides to self-management of individual teammates across teams.

5 DESIGN AND ARCHITECTURE

Design and Architecture is of major concern while dealing with large scale projects. [9] This could easily get complicated and can go off track since many teams will be working parallelly on various modules of the project. Agile projects will usually not rely on sound architectural design upfront, instead will rely on code refracturing as and when the design changes. Conversely, refracturing is a poor substitute for a large-scale project. In a large agile team, there will be an architecture owner team. [9] As these teams are sub-divided into smaller sub teams, each sub-team will have an architectural owner who is a member of architectural owner team. This helps in educating sub-teams on overall architecture of the project and increases the chance that the overall architecture strategy will address the full needs of the overall situation. A Chief Architecture Owner who is responsible for the overall architecture of the project will be facilitating the group. This role could be a rotating one. [10]

4 CONCLUSION

Agile aims at making best use of individual talent in a given team. There is an improvement in the growth and learning curve of every individual in an agile team, because agile team are formed by motivated self-organizing cross-functional teams. Everyone in the team, gets to work on each part of the project. Common vocabulary and work culture are the two major aspects of a good team. Problems can be solved quickly when different comes together. Successful implementation of the project also depends on where and how each team is formed and placed. Good system design and Architecture decides the right formation and placement of teams. This requires some modification in organizational structural and management level. These changes cannot be done instantaneously, and it takes time. It may not yield the best results in the very beginning and it might be a bit hard to accommodate and adjust to these changes. Having said that, in the long run agile and scrum of scrums implementation for a large-scale project is the best proven methodology for the smooth software development process. Hence, a collaborative environment with the right organizational structure, is the building block of companies, which leads to drastic improvement in team's performance and organization success.

ACKNOWLEDGMENTS

We would like to thank Professor Michael Findler for assigning us this interesting research work. We got to learn a lot about the advantages and disadvantages of Agile Methodology. We also

gained knowledge of how the teams handle Agile when working with other teams and are able to show that in this paper. We would like to continue this research ahead and use this in a practical setting for our project which we will implement in our SER 574 Advance Software Design class.

REFERENCES

[1] N. Singh, "Agile Software Development: An Overview," vol. 2, no. 4, p. 6, 2016.

[2] A. V. J. B. N. P. Jeff Sutherland, "Distributed Scrum: Agile Project Management with Outsourced Development," Proceedings of the 40th Hawaii International Conference on System Sciences, p. 10, 2007.

[3] M. Clews, "7 Agile development problems Large-Scale Scrum helps you overcome," Equinox IT, 16 07 2015. [Online]. Available: https://www.equinox.co.nz/blog/agile-development-problems-large-scale-scrum-overcome. [Accessed 22 01 2020].

[4] V. Botsvadze, "10 Teamwork Quotes From Great Leaders," Medium, 05 07 2017. [Online]. Available: https://medium.com/@VladimerBotsvadze/10-teamwork-quotes-from-great-leaders-3d76bb44c1f3. [Accessed 22 01 2020].

[5] D. Carlson, "Debunking Agile Myths," Cross Talk, p. 36, 2017.

[6] P. F. K. C. M. J. Jeff Schwalb, "Cross Talk Beyond Agile Manifesto," The Journal of Defense Software Engineering , vol. 29, no. 6, p. 12, 2016.

[7] E. S. Dick Carlson, "Scrum is Simple," Cross Talk, p. 02, 2017.

[8] N. B. M. Torgeir Dingsoyr, "Towards Principles of Large Scale Agile Development," p. 2, 2014.

[9] P. M. P. T. V. P. Mirjana Maric, "Documenting Agile Architecture: Practices and Recommendations," p. 17, 2016.

[10] S. W. Ambler, "Agile Architecture: Strategies for Scaling Agile Development," [Online]. Available: http://agilemodeling.com/essays/agileArchitecture.htm. [Accessed 20 01 2020].