Agile Teams Orchestrations

Working together with Agile Teams

|  |  |
| --- | --- |
| Aneesh Dalvi  CIDSE  Arizona State University  Tempe, Arizona, USA  [adalvi1@asu.edu](mailto:adalvi1@asu.edu) | Gangadhar Matti  CIDSE  Arizona State University  Tempe, Arizona   gmatti1@asu.edu |

ABSTRACT

Traditionally, software was developed as a whole in linear sequential phases with a fixed set of requirements where each phase depends on deliverables of the previous phases. However, the requirements for new age software changed frequently and needed a continuous delivery with 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 10-15 people). Effective coordination between teams and issues with design and architecture are of major concern with increase in complexity of software projects and teams.

∗Article Title Footnote needs to be captured as Title Note

†Author Footnote to be captured as Author Note

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*WOODSTOCK’18, June, 2018, El Paso, Texas USA*

© 2018 Copyright held by the owner/author(s). 978-1-4503-0000-0/18/06...$15.00

https://doi.org/10.1145/1234567890

CCS CONCEPTS

• Insert CCS text here • Insert CCS text here   • Insert CCS text here

KEYWORDS

Insert keyword text, Insert keyword text, Insert keyword text, Insert keyword text

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK’18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1 INTRODUCTION

In the past two decades, we noticed tremendous expansion of the internet, which led to 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 increase in the complexity and requirements, led to the increase in people involved in building software. As a result of this, team is responsible in 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. 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 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, project falls on track, resolving dependencies and duplicity of work. In section 2 we discuss various issues of teamwork that should be considered while working in large scale. In section 2.1 we discuss the scrum approach that should be followed for large scale development. Section 2.2 addresses the challenges in the inter-team communication over large scale projects. Section 2.3 highlights how design and architectural affects in large projects.

2 TEAM WORK

Ideally agile is effective for team size ranging from 5-15 members. But in large-scale projects teams consists of single scrum to several hundred people. These large group of people are divided into smaller teams. Dedicated module of the system is developed by each small team. These groups can be categorized into scrum teams, distributed scrum of scrums and totally integrated scrums. Scrums team is usually 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 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 agile approach. Every team needs to follow same methodology in order to minimize the issues addressed in communication between teams. **(Totally integrated scrum)**

A productive successful scrum team needs to have a right balance between technical skills and experience. Every project relies on good teams. Effective team work helps drive the project towards success. Team players should be committed to the goal, being reliable and responsible, support and respect others, be active listener and a good problem solver. 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. 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. 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. 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. 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. 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. 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. 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.

In the below paragraph, it is explained how alt-txt value is placed in **MS Word 2010**. To add alternative text to a picture in Word 2010, follow these steps:

1. In a Word 2010 document, insert a picture.
2. Right click on the inserted picture and select the **Format Picture** option.
3. Select the **Alt Txt** option from the left-side panel options.
4. In the "Title:" and "Description:" text boxes, type the text you want to represent the picture, and then click "Close".

Below are steps to place alt-txt value in **MS Word 2013/2016**. To add alternative text to a picture in Word 2013/2016, follow these steps:

1. In a Word 2013/2016 document, insert a picture.
2. Right click on the inserted picture and select the **Format Picture** option.
3. In the settings at the right side of the window, click on the "Layout & Properties" icon (3rd option).
4. Expand **Alt Txt** option.
5. In the "Title:" and "Description:" text boxes, type the text you want to represent the picture, and then click "Close".

*1.1.1 Heading Level 3.* Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

*1.1.1.1 Heading Level 4.*Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

ACKNOWLEDGMENTS

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

REFERENCES

[1] Patricia S. Abril and Robert Plant, 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan, 2007), 36-44. DOI: <https://doi.org/>10.1145/1188913.1188915.

[2] Sten Andler. 1979. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774

[3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.

[4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00