Collaborating Agile Teams

Trupti Khatavkar
tkhatavk@asu.edu
Arizona State University
Tempe, Arizona, United States

Srajan Gupta
sgupt182@asu.edu
Arizona State University
Tempe, Arizona, United States

ABSTRACT

Agile development model has its own advantages such as, flexibility, transperance, early and predictable delivery and it focuses on users. Taking this into consideration, large projects can be developed using agile techniques by dividing them into different components and distributing the components to different teams. This is generally termed as scrum of scrums. In such large project, each team should take the full responsibility of their own component and then the components are interfaced to build the whole system. The main challenge is to maintain minimum dependencies among teams as possible. This paper suggests methods to achieve successful large agile project. Also, methods to overcome architecure or design related risks are also put forward.

KEYWORDS

agile, components, dependencies, collaboration, communication

ACM Reference Format:
Trupti Khatavkar and Srajan Gupta. 2019. Collaborating Agile Teams. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

2 MAIN

2.1 Methods to achieve successful large scale agile project

The main challenges for working on a large project using multiple agile teams is communication, dependencies between components and interfacing those components. To achieve a successful project, following methods can be useful.

- Separate Product Backlog for each team
  As shown in Figure 1, it is very important that every team should have a strong product backlog in agile when working in a distributed envirenment. But, clear separation of work is equally important. Every day to day operations should be separated among the teams. Overlapping situations should be planned during the cross team meetings.

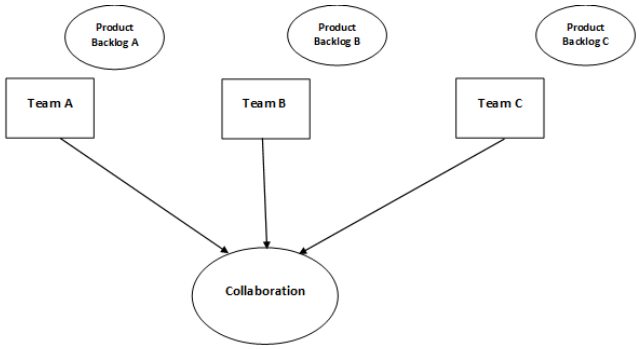

Figure 1: Collaboration of teams and their backlogs.

- Tools for Interaction
  Choosing the correct tool for communication matters when working in distributed agile environment. Slack and GitHub are very effective for this.
- Agile Practices
  All the important agile practices such as standups, planning, deliverables and retrospectives should be defined by each team as they think is suitable.
- Face to face communication
  Using tools for communication is effective, but many times there are broken communications due to various technical issues such as poor connections, fault in the devices, etc. To overcome these, regular cross teams face to face communications are also important.

2.2 Architecture/Design for distributed agile project

Architecture defines the structure of the project. A well defined architecure makes a base for a succesful project. As multiple teams work on a single project, it is important to have a fixed architecture so that different teams can work on different components of the architecture. If there is no fixed single plan, interfacing of the components would be a difficult task. Also, the architecure can get messed up if multiple teams update it. The integrity of the whole project can be affected if architecture is messed up.

To overcome this problem, there should be a single Chief Architect (or two, working in a pair) who will guide the teams and ensures they do not stumble upon the architecure. Chief architect works on a high level architectural issues that involves all the components and inerfaces between them.

Another method would be to crete the Architecture group. It is composed of each team's most skilled person. This group handles all the arcitectural issues rather than one person handling them. The group should not exceed more than 20 people, else communication

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

problem arises. The group understands the subsystem's relationships and the architecure as a whole. Thus, the group decides how each team works and handles the project.

3 CONCLUSION