Design, Architecture, and Communication in Agile Development

David Lahtinen[*]
Ruihao Zhou[*]
dlahtine@asu.edu
rzhou21@asu.edu

Abstract

Individuals and interactions play an important role in Agile development process and face to face communication is the best way to discuss the progress and convey information across different Agile teams. However, due to the some uncontrolled factors, collaboration between different teams poses problems. For example, each team could have different working space about their own work. In another words, some team's work are not catching up with the current progress. Therefore, it's worthy to think what is important when agile teams have to work together with other agile teams and this paper will show the answer to you. This paper does not discover anything new, so there are no results section.

CCS Concepts • **Software and its engineering → Programming teams**.

Keywords Agile, Communication, Collaboration, Architecture, Design

1 Introduction

"Agile software development is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s)."[2]Now, it's popular to have multiple agile teams to work on one big project. One of the benefits of having Agile teams is that

---

[*]Both authors contributed equally to this research.

they closely correspond to modular architecture. Because communication between teams is somewhat restricted by the realities of the large teams, but communication within teams is encouraged, the engineers on the teams are encouraged to have very simple interfaces with engineers working on other teams or in other departments which increases the maintainability of your software. Another more subtle reason behind encouraging cross functional teams is the sense of ownership that is automatically enforced as team members start to associate themselves more with the product that they are working on than their individual job titles.[3] Each Agile team is formed with 5 to 11 people who follow the Agile development process like define, test, deploy, and working towards the common goal. All Agile teams are all on the same train, collaborating with other teams to contribute its vision and roadmap. Although it sounds great, it can be difficult to implement successfully in reality. For example, each team has large number of people and locate in different places which cause stand up daily much more difficult. Follwing are some important attrubutes each Agile teams should be adopted and what's the relationship between architecture and design.

2 Background

Agile teams are an established practice in the industry. There are 3 things that Agile workfloow standardizes: Goals, Standards, and Communications

2.1 Goals

"Agile Teams do not operate independently; they power the Agile Release Train by collaborating and building increasingly valuable increments of working solutions. All teams operate within a common framework that governs and guides the train. They plan together, integrate and demo together, deploy and release together, and learn together."[1] Companies should organize cross-team meetings every sprint to craft a clear idea about the final objectives and separation of work for the sprint. If each team has its own ideas about project, it's going to be really hard to continue the project. They should spend their efforts towards same goal and work for it. Therefore, product backlogs are popular among developers which can help them identify and plan for cross-team dependencies.

2.2 Standards

Each company should agree on the standards they would adopt and follow before the project starts and apply this accross all teams. For this kind of communication, meetings between teams are not necessary, but it may make sense to hold a meeting with them all anyways to give them all the same message at the same time, and make sure they are all being held to the same standards. For example, which kind of development procedure will teams use and how they will create testing for each phase? If the company decides that test-driven development will be the way their sprints are run, they may have an engineering all-hands meeting to make sure everyone is on board. Of course, standards can also include patterns, code style and more. The whole work efforts across mean it's a huge product and it's appropriate to create checking points in order to check errors more easily and decrease developer's burden. Also, all teams should have full access to the whole project repository and contribute at anytime.

2.3 Communications

Each team should have communication quickly, frequently, quickly as soon as possible. It's all about teamwork. Using some project management tools such as Slack, Git and more to convey information. Daily and face to face meeting are two core ideas in Agile development. If you need to communicate ideas to the whole engineering team, hold a meeting with representatives from all teams and be there to answer any questions the members have. Create slides, outilines, diagrams, or whatever tools can aid you in communication. Before taking actions, you should get their agreement. "Communication improves, teamwork improves and "Us vs them" is reduced. And if the distance is not very long, seed visits infrequent intervals across the length of the project, it builds trust between the client and the team and between the team members themselves. Don't forget, that the highest performing teams are those based on trust!"[3]

Feedback, each team should incorporate feedback quickly from other teams and come up with solutions to fix it before it's going to deeper development. Take Scrum for example, work is divided into small iterations called Sprint which includes a lists of tasks. Since each Sprint's size is not big, they can work on it with expected time. Once finish one, they can reflect easily whether it's good or not and make some necessary change. And when one Sprint is finished, each team is able to evaluate it and decide which parts should be kept or altered. Changes can be incorporated into next iteration which can save lots of time and increase productivity.

3 Architecture and Design in Agile Teams

Architecture is design at the highest level. At this level, the Agile teams cannot work separately from the rest; there must

be communication and coordination. As the government describes in its article about Agile team communication, they recommend holding regular -daily if possible- scrum meetings consisting of representatives from related agile teams so that each team is on board with the bigger picture. [4] If your project has multiple project owners, they should also meet regularly so that releases and features can be planned, but also so that each team can learn from the mistakes and novel solutions of the others. [4] The closer to the modular level that you examine, however, the less communication to other teams about it is required, as they likely only want to know the interfaces. This lower level of communication no longer corresponds to architecture, but it still corresponds to design as the engineering teams communicate more with each other to design the specifics of their modules.

One of the benefits of having Agile teams is this close correlation to modular architecture. Because communication between teams is somewhat restricted by the realities of the large teams, but communication within teams is encouraged, the engineers on the teams are encouraged to have very simple interfaces with engineers working on other teams or in other departments, which increases the maintainability of software. Another more subtle reason behind encouraging cross functional teams is the sense of ownership that is automatically enforced as team members start to associate themselves more with the product that they are working on than their individual job titles.[3]

4 Discussion

Agile is already being used in the industry to great effect; its focus on flexible requirements is optimal for long-term projects and users with evolving needs.

5 Conclusion

Large projects cannot be completed in a timely manner without large workforces, but communication within that workforce must be managed effectively. The most effective way to do this is to put everyone on small scrum teams. The problem this poses to communication between teams can be overcome by having regular contact via representatives, so that the teams can stay in regular communication without spending all their time in meetings.

References

[1] Online editor. 2018. Agile software development. Retrieved January 31, 2019 from https://en.wikipedia.org/wiki/Agile_software_development
[2] Online editor. 2018. Agile Teams. Retrieved January 31, 2019 from https://www.scaledagileframework.com/agile-teams/
[3] David AYZA ENERO. 2018. DISTRIBUTED AGILE TEAMS: 9 HACKS THAT MAKE THEM WORK. Retrieved January 20, 2019 from https://apiumhub.com/tech-blog-barcelona/distributed-agile-teams-hacks/
[4] Lulit T. 2018. Collaboration Across Agile Teams. Retrieved January 19, 2019 from https://tech.gsa.gov/guides/Collaboration_Across_Agile_Teams/