Collective Work Across Cross Functional Agile Teams

Amit Pandey
Computer Software Engineering
Arizona State University
apande36@asu.edu

Mayank Batra
Computer Software Engineering
Arizona State University
mbatra3@asu.edu

Abstract

The Agile approach has been incorporated in many small to mid-sized organizations. Agile methodologies have been a great success, but still tagging along with many different teams together is a difficult task. Fortunately, there exist many result-oriented strategies to make interconnected agile teams work together. Communication, negotiation, challenging assumptions, having a willingness to change, learning from others, team structuring, tooling, and having clear designs are all important for the success of many Agile teams.

CCS Concepts

• Agile Process; • Team Success; • Inter Communication

KEYWORDS

Agile process, team-collaboration, design, architecture,

ACM Reference format:
Amit Pandey, Mayank Batra. Collective Work Across Cross Functional Agile Teams. pages
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1. Introduction

Agile project management is an iterative approach to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration. [1] Agile manifesto states that “Individuals and interactions over processes and tools” [2] Which helps teams realize what is important. With roles clearly defined, regular communication and feedback, members from different departments - developers, testers, business analysts and product managers can seamlessly work on a single product with the same intent, incentive and vision. The most intuitive part of Agile is its closed feedback loop. This feedback loop allows customer to be involved in the development of product, which in turn helps the resulting deliverable. Most importantly, Agile practices allow for changes in requirements without a huge cost to development. Despite the success and great features of Agile, it can be difficult to productively implement it across teams. There are many strategies that allow multiple Agile teams to still be successful together, such as creating a collaborative environment, maintaining communication channels, and choosing the right tools. Having a clear architecture and design of a system can also point teams towards the same goals. This
paper dives into the proper techniques for managing large, multi-team Agile projects.

2. Team Success

A project's successful delivery is heavily guided by the Team's structure. In traditional businesses, teams are split up based on their role. There is a clear demarcation between testers, developers and business analysts. This role-based separation can create void within a company. If this void is not filled carefully with interleaving communication, then successful delivery of a software product may get impacted. These void block team members from naturally collaborating with each other to fix a problem. While working on a multi-team Agile project, it is important to eliminate such voids, as it allows for more organic collaboration. [3] For success in multiple Agile teams, there must be a balance between keeping team members together and creating a **collaborative environment** across teams.

3. Interactions

Inter-communication is one of the most important aspects when working on a multi-team Agile project. Similar to how communication can make or break individual teams, it can have even greater effects on multi-team projects. When you can’t strike a perfect balance in the team layout, alternative forms of communication can help. Even physical siloes between developers can be overcome by having good communication channels. These events can come in the form of cross-team stand-ups, multiple project owners communicating, and cross-team planning. [2]
By making communication convenient, teams can easily work together to move the project forward.

4 Clear Goals

Communication alone will not ensure a successful project, it is important to have a set of clear goals for each team. These goals should be publicly known, so each team knows exactly what every other team is working on. With each team having a defined and unique part of the project, they’ll be able to work independently, and "...plan for cross-team dependencies" [4]. Similar to convenient communication, having clear goals helps teams organically collaborate. By knowing what each team is working on,

teams with similar goals can make sure they develop something that's concise
and can work when combined.

5. Using the apt tooling

It is necessary to focus on the tools used to implement agile methodology. Tools like GitHub facilitate continuous integration and collaboration in such a way that the teams work exclusively on their part without modifying other team's effort inadvertently. Tools of automated testing like selenium also help and reduce the testing work, this facilitates more efficiency in the team.

6. Agile Architecture

Notion Behind the architecture-

One of the main reasons behind this architecture is that the code needs to be maintainable i.e. the existing code should be operational for many years as well as adapts to the changes that is added after the code is deployed. If the code is again architecture, the cost that is involved in doing that increases exponentially. Moreover, it delays the functioning of the code. From the customer's perspective, the code should be always in running state mode i.e. the software built by the developer must be always in a working and in a ready state. It basically avoids the "start-stop-start" nature of the system. Another thing to keep in mind while building the system, is that the code needs to be both testable as well as deployable. Take for example, micro services architecture is an architectural style which is a collection of loosely coupled services which has many benefits such as highly maintainable, testable and also independently deployable.

Role of Architecture in Agile System

The duties of an architect are maintaining and declaring the structure of the solution of the system and also making sure that it meets the needs of all the stakeholders those are involved in it. The architect needs to ensure maximum optimization of all the stakeholders who have different emphasis on different parts of the solutions so that their requirements are met. There are many factors that impacts the approach to agile architecture. Firstly, the uncertainty of requirements deters early feedback from the stakeholder. Secondly, Technical risk also effects if a challenging requirement related to architecture is inclined to more up-front architecture. Big design up front means that the team would need the full specification and requirements and architecture design before the beginning of the development. The architecture may change or evolve during the phase of development.

Agile architecture allows incremental value delivery by balancing between emergent design and intentional architecture.

Applying architecture to agile teams

In agile architecture, each individual in a team must contribute significantly in providing the inputs to the architecture. Feedbacks should be given regularly by the customers so that there is a scope of improvement.
There should be proper communication among the team members so that they convey their perspective and what they are performing and how they are moving forward with it. Not to mention the least, there must be active participation of the stakeholders as well. They should be able to put forth their complaints, intents, views. The system should be built which is best in the interest of the customers and the company.

CONCLUSION

Agile methodologies came into life as there was a need to incorporate changes in requirements environment. It requires communication between the developers and stakeholders. The developers must be able to cope up with the changes at any time and also maximize the investments of the stakeholders.
One of the reasons of agile implementation is that it makes the life of developers, testers easier. The system which you are developing should be flexible and incorporate Agile methods. The transition from plan driven approach to agile decreases the extra cost and time and reduces the work of your development team. Secondly, there is a misconception that processes, tools, documentation, contract negotiation are not used in agile methods, it is just that over emphasis is avoided. The main agile methodologies that are used commonly Extreme Programming, Agile modeling, and SCRUM.

Agile methodologies are not best suited for every project. Sometimes, the communication between the developer and the stakeholder is difficult, when the team does not have knowledgeable developers, agile is not appropriate.
Where the development team has many skilled people, and the deadlines are tight, agile is apt.

References

[1] "Agile Project Management," [Online]. Available: https://www.atlassian.com/agile/project-management.
[2] "Agile Manefasto," [Online]. Available: https://www.atlassian.com/agile/manifesto.
[3] "Collaborations across agile teams," [Online]. Available: https://tech.gsa.gov/guides/Collaboration_Across_Agile_Teams/.
[4] [Online]. Available: https://hbr.org/2018/05/how-to-make-sure-agile-teams-can-worktogether.
[5] http://agilemodeling.com/essays/agileArchitecture.htm

[6]https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/

[7]http://www.agilemodeling.com/essays/architectureOwner.htm

[8]https://hackernoon.com/agile-architecture-the-rise-of-messy-inconsistent-and-emergent-architecture-e6801ab25b61

[9]http://agilemodeling.com/essays/agileArchitecture.htm

[10]https://www.scaledagileframework.com/agile-architecture/