Collective Work Across Cross Functional Agile

Teams

Amit Pandey Mayank Batra

Computer Software Engineering Computer Software Engineering

Arizona State University Arizona State University

[apande36@asu.edu](mailto:apande36@asu.edu) [mbatra3@asu.edu](mailto: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. Tools