Designing an Agile Collaboration: How to Succeed in Cross-Team Interactions

Paul Horton
pahorton@asu.edu
Arizona State University
Mesa, Arizona

Ruby Zhao
qrzhao@asu.edu
Arizona State University
Mesa, Arizona

Abstract

Agile models within the field of software engineering are widely used to facilitate flexibility and ease of communication in order to solve dynamic engineering challenges. For agile teams to succeed on a complex project, they must be able to work not only with their own members but also with other teams. The key to collaboration in an agile setting is strong architectural planning that allows for multiple teams to contribute simultaneously. By creating a valid architecture that allows for collaborative programming, all participants in the development are given immediate feedback. This ultimately leads to better software products developed in a more efficient manner.

Keywords agile, teamwork, design, architecture, project management, software engineering

ACM Reference Format:
Paul Horton and Ruby Zhao. 2019. Designing an Agile Collaboration: How to Succeed in Cross-Team Interactions. In *Proceedings of Mesa '18: SER 574 (Mesa '18).* ACM, New York, NY, USA, 2 pages. https://doi.org/11.2222/3333333.4444444

1 Introduction

Agility is currently the byword of the software engineering field. Software engineers must respond to customer requirements with speed and flexibility, and for many projects, development teams have turned to Agile methodology as the answer to these demands. A good agile team is one that communicates, but is also responsive to, changing requirements to and from other teams. Understanding the communication techniques used on best practice agile teams reveals optimal development strategies for future projects. This work primarily focuses on cross-team interaction methodology for agile development.

The section "Important Aspects" will discuss which qualities allow teams to work together successfully. This section lists actionable qualities of a healthy team dynamic that can be incorporated into any team making use of agile processes. The following section, "Design and Architecture", discusses the role of design and architecture in cross-team communication efforts. This section highlights the importance of architectural design as a tool to assist cross-team development as well as how design planning and agile can coexist.

2 Agile Collaboration

2.1 Important Aspects

2.2 Design and Architecture

Architectural planning is a vital part of successful cross-team agile development despite the perceived conflict between the two practices. A key aspect of Agile is high adaptability to change over following strict plans [**?**]. This seems to directly contrast the traditional viewpoint for architecture which provides the "floor plans" as framework directly tied to the requirements of the project [4]. At first glance, the two are at odds on the fundamental level with agilists concerned with Big Up Front Design (BUFD) and You Ain't Gonna Need It (YAGNI) features and architecturalists seeing the Agile methodology as amature [2]. The problems in this dynamic arise when an architectural team design a team for an implementation team to create. When changes arise in requirements, "the architecture team do not always fully understand the repercussions their design changes will have on existing and new product features," [1]. By integrating the architecture and implementation teams into agile scrum teams, all developers have a stake in the final product and understand the impacts of changes requirements [1].

The key to agile architecture surrounds the sprintable nature of scrum. With up-front design that decomposes the architecture into sections with architecturally signifigant boundaries, teams are able to distribute the systems and implement in sprints [3]. Newly designed parts of the architecture can be rolled out in parallel to existing architecture to build up the planned design [1]. By producing the parts in increments, developers are able to "learn and adjust along the way," [1]. To account for change, the architecture can be

Mesa '18, Jan 20, 2019, Mesa, AZ

ACM ISBN 111-2-333-4444-1/20/19…$15.00
https://doi.org/11.2222/3333333.4444444

designed such that a range of options are considered so that modifications can be made as the system is being built [3].

3 Conclusion

References

[1] Mark Isham. 2008. Agile architecture is possible you first have to believe!. In *Agile, 2008. AGILE'08. Conference.* IEEE, 484–489.

[2] Philippe Kruchten. 2010. Software architecture and agile software development: a clash of two cultures?. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2.* ACM, 497–498.

[3] James Madison. 2010. Agile architecture interactions. *IEEE software* 27, 2 (2010).

[4] Dewayne E Perry and Alexander L Wolf. 1992. Foundations for the study of software architecture. *ACM SIGSOFT Software engineering notes* 17, 4 (1992), 40–52.