

Draft of Paper for SER - 574

Ruben Acuna
Arizona State University
Phoenix, USA
ruben.acuna@asu.edu

Sarthak Tiwari
Arizona State University
Phoenix, USA
sarthak.tiwari@asu.edu

Abstract

With the boom in the use of agile process model, imperfect implementations of agile are frequently being seen, out of which the problem of inter-team communication in agile teams is one of the most common issue. An agile team by its definition is a group of people who are self-sufficient to bring their responsibilities to closure, the interaction between different teams is thus considered minimal in most projects implementing agile. This causes problems when the integration of end-products of different teams is carried out. In this paper we have taken a detailed look at this problem and how we can mitigate this.

CCS Concepts • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Keywords Agile, Software Engineering, Process Models, Inter-team, Management

ACM Reference Format:

Ruben Acuna and Sarthak Tiwari. 2018. Draft of Paper for SER - 574. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/1122445.1122456>

1 Introduction

As mentioned in the Agile Manifesto [2], Individuals and Interactions are given more value than the processes and tools. At the same time, an agile team is supposed to contain all that is required for them to do their work, thus interactions no matter how valuable usually are done intra-team only. Thus, when a situation arrives where two or more agile teams need to work together to achieve a common goal the usual approach taken by companies fail. The agile approach of interacting in-person fails due to a few reasons such as the vertical

structure of the company organization and the existence of middle managers [3]. This large number of intermediaries in inter team communication causes the process to crumble and fall apart as it takes longer than usual time for messages to reach their destination then what agile can afford. This problem is harder to fix as this is so embedded in the working culture of the current organizations that changing them will take a long time, thus an immediate patch that works is required. And that's where the approaches listed in the article comes in. We have collected a couple of approaches which if implemented efficiently results in a more stable and functioning agile process model. Thus, in the following sections we will be explaining couple of approaches we came across.

In Section 2, we will be explaining the improvements that can be made to the process and team structures to get better collaboration out of the team involved.

In Section 3, we will be explaining the enhancements that can be done in the technology stack being used and in the different ways in which technology can help in improving the process.

In Section 4, we will be describing the importance of architecture and design in resolving this issue of inter-team communication in agile.

2 Process Improvements

There are many small changes [1] that we can incorporate in our process model to make sure that parallel developing agile teams do not run into problems when they reach the integration phase. These changes are to be made part of the entire process and are not to be implemented only in the end.

2.1 Changes in standups

One of the most critical aspect of agile process model is to have daily standups which are the platform serving the purpose of letting each team member know the work being done in the other parts of the team and correlate it with the work being done by them. This results in escalation of differences between the development early in the process and prevents end moment discovery of mismatches in interfaces and such. In the case of multiple agile teams this problem is compounded as usually a daily stand up is a closed activity of the team itself, thus preventing other teams from knowing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Woodstock '18, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9999-9/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

the results or discussions of each other. This can be resolved by having a representative of each concerned team being present in the daily standup thus letting each team know the status of other teams.

2.2 Changes to Product Owner

Though in usual implementations the product owner is responsible for agile teams under his supervision, in large projects with multiple agile teams where a number of product owners are present sometimes over time the vision of the owners may get too distinct from each other thus pushing the development track in different directions. This can be limited by having regular meetings of product owners where the scope and vision of the project could be synced again. This can be a bi-weekly or monthly meeting depending on the size of the project.

2.3 Changes in Planning Sessions

All the initial, intermediate and final planning sessions should be made such that all the teams which are or could be impacted by that part of the project are part of the meeting. This can assist in early agreement on high-level requirements and standardization of inter-team interfaces.

3 Technology Improvements

A good technology stack can be a powerful tool in maintaining a widely distributed team. Good communication and management tools can help the teams in keeping track of things that they need to do so that other teams can work as intended. The following are some of the areas where the technology can assist the teams in making a more efficient agile process environment.

3.1 Communication Tools

As agile focuses on personal interaction with highest importance given to face-to-face conversation, conferencing tools such as video conferencing and WebEx etc. can help the team interactions become more fluid and clearer. They also make the communication real-time thus removing the lag in process due to time spent in communicating ideas across teams.

3.2 Integration Tools

Continuous integration tools can go a long way in finding out inter-team problems early in the process as every time any team makes a change, its impact on the entire project can be seen.

4 Importance of Architecture/Design

As we know that the product owner in an agile process model is the person with the vision of what the project will look like and if it is a small team the product owner can clearly pass this to each and every member of the team and even each

member can query the product owner directly when in doubt. But in large scale projects where multiple agile teams are working together in supervision of a few product owners it is practically impossible for the owner to keep doing what they did in a small team, that's where a formal definition of their vision comes in handy as it enables the teams to look up to something when encountering a design decision. The design or architecture in this case acts as a common vision for not only the teams but also for the communication between all the product owners. The design acts as a "deadlock breaker in decisions" [4] as when the teams can't come to a common consensus the design shows the path to take.

5 Conclusion

As we have seen, the issue of inter-team communication can be a big problem in agile teams working on the same thing as intermediaries in communication and lack of common personal causes things to go haywire if left unchecked. To reduce this problem a number of small additions can be done to the process as specified but the most important part of the change is a good architecture / design which is available to all the teams and thus provides a common vision to all the teams involved.

6 CCS Concepts and User-Defined Keywords

Two elements of the "acmart" document class provide powerful taxonomic tools for you to help readers find your work in an online search.

The ACM Computing Classification System — <https://www.acm.org/publications/class-2012> — is a set of classifiers and concepts that describe the computing discipline. Authors can select entries from this classification system, via <https://dl.acm.org/ccs/ccs.cfm>, and generate the commands to be included in the \LaTeX source.

User-defined keywords are a comma-separated list of words and phrases of the authors' choosing, providing a more flexible way of describing the research being presented.

CCS concepts and user-defined keywords are required for all short- and full-length articles, and optional for two-page abstracts.

References

- [1] [n. d.]. Collaboration Across Agile Teams. https://tech.gsa.gov/guides/Collaboration_Across_Agile_Teams/
- [2] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Melor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. 2001. Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>
- [3] Nicolas Frankel. 2016. Making Sure Inter-Team Communication Doesn't Work. <https://dzone.com/articles/making-sure-inter-team-communication-doesnt-work/> [Online; posted 13-October-2016].

- [4] Scott W. Ambler. [n. d.]. The Architecture Owner Role: How Architects Fit in on Agile Teams. [http://agilemodeling.com/essays/](http://agilemodeling.com/essays/architectureOwner.htm/)

[architectureOwner.htm/](http://agilemodeling.com/essays/architectureOwner.htm/)