Solutions for Effective Cross-team Collaborations

Bingrui Feng  
 CIDSE  
 Arizona State University  
Mesa, AZ, USA  
 bfeng16@asu.edu

Huijing Liang  
 CIDSE  
 Arizona State University  
Mesa, AZ, USA  
 hliang36@asu.edu

ABSTRACT

Nowadays more and more companies start adapting using agile methodology to manage their software development process. Agile methodology provides tons of benefits such as fast delivery and flexibility compared to other methodologies. For the small size of companies, it only requires a small size of team to work on one single project, however for larger companies, cross-team collaborations become more and more common. Lack of effective collaboration across the team slowly becomes a growing pain. So how can we make different agile team work together efficiently? Based on our research we think solutions should at least include: clear objectives and separation, no departmental silos, collaborative architecture and design guidelines [1]

CCS CONCEPTS

• Insert CCS text here • Insert CCS text here   • Insert CCS text here

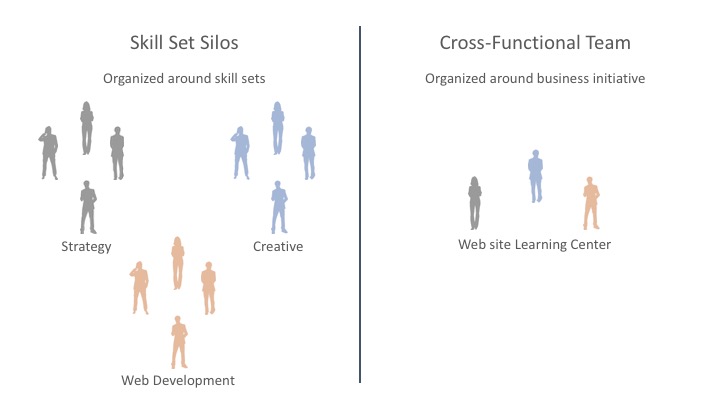
KEYWORDS

Agile, Collaborations, Sprint, Cross-team, Design, Architecture

**1    Clear objectives and separation**

Typically, agile teams are interdisciplinary and consists of programmers as well as test engineers and documentation experts. Every team works separately with their own sprint or together but separate teamwork. Then all the work needs to be done in a uniform way. The functions of every agile team need to be considered functionally relevant in an agile development environment [2]. Each team member should know the goals of the team and separation of work clearly. The clear separation of work among multiple teams also creates independence for every team to reduce the reduplicative work and enables the team to identify and plan dependencies across teams. The best method is developing and maintaining a team product backlog for every team to make sure every team is working on solutions towards the final goal.

Meanwhile, cross-team daily stand-ups are necessary. each team should choose one representative to attend daily meeting, it is convenient to share team updates and remind every team to adjust their work plan. So that the responsibility of every team will be much clearer, which is crucial to the delivery of the whole picture. 



**2 No departmental silos**

In software development industries of al size, it is very common for multiple departments being involved in the same project and working together. However, under these situations, departmental silos sometimes become obstacles of achieving collaboration. It is the duty of the executive leaders and management to prepare and equip their teams with the proper mind-set and vision to break down these destructive organizational barriers [3].

Proper Communication and willing to share necessary information are the key to solve departmental silos. Each department usually have their own perspective regarding special terms. In a big project, lack of communication or information sharing can be very harmful for the project, many times may even not be done purposely. Another reason that could result departmental silos might be design flaws of the project. During designing process, it should enhance the interaction and necessary information shared between departments, and all of those should be also be documented on file, knowing who oversees what, and who to seek for information is very important. If in the beginning of the design process, all of those are clear then the communication will be more efficient and better for the final delivered product

**3    Collaborative architecture and design guidelines**

According to the 11th principle in the Agile Manifesto, “The best architectures, requirements, and designs emerge from self-organizing teams.” Therefore, the organization should provide guidelines for better collaboration between teams, and some rules should be formulated, such as daily stand-up meetings between teams, Multiple Product Owners Check-in, Cross-Team Retrospectives and so on. The goals of every team will be more clear and repetitive work will be reduced by these methods.

At the same time, these methods also help ensure the timeliness of communication. The agile teams are also based on discipline all the time. Although the agile team advocates self-organization and self-management of the team, it must also follow guidelines. The team guidelines in the agile team are stricter than the disciplines of other R & D teams. For example: Whether in the initial stage of product demand or the final gray test, this requires the team to make frequent reviews and necessary adjustments; frequently deliver workable products; timely planning of iteration time; attention to the rhythm of research and development; Information visualization; continuous integration and automated testing.

**4**

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

**5**

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

**6**

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

ACKNOWLEDGMENTS

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

REFERENCES

[1] Anon. GSA Tech Guides. Retrieved January 20, 2020 from <https://tech.gsa.gov/guides/Collaboration_Across_Agile_Teams/>

[2] C.M. Tartaglia and P. Ramnath, 2006. Using open spaces to resolve cross team issue [software development] DOI:

<https://ieeexplore.ieee.org/document/1609818/authors#authors>

[3] Brent Gleeson. 2017. 5 Ways to Destroy the Pesky Silos in Your Organization. (June 2017). Retrieved January 20, 2020 from [https://www.inc.com/brent-gleeson/5-ways-to-destroy-the-pesky-silos-in-your-organization.htmlConference](https://www.inc.com/brent-gleeson/5-ways-to-destroy-the-pesky-silos-in-your-organization.html) Name:ACM Woodstock conference

[4] <https://www.thoughtworks.com/insights/blog/collaboration-techniques-large-distributed-agile-projects>

[5] <http://www.scaledagileframework.com/agile-architecture/>

[6] <https://www.scalablepath.com/blog/manage-distributed-teams/>

[7] <https://agilevelocity.com/team/blogimprove-cross-functional-collaboration/>

[8] <https://www.scaledagileframework.com/agile-teams/>

[9] <https://www.disciplinedagiledelivery.com/agility-at-scale/large-agile-teams/>

[10] <https://agilemarketing.net/crossfunctional-teams-part/>

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00