Assignment 1

Writing a paper

Abhishek Haksar  
 CIDSE  
 Arizona State University  
 Tempe AZ USA  
 ahaksar@asu.edu

Amanjot Singh  
 CIDSE  
 Arizona State University  
 Tempe AZ USA  
asing225@asu.edu

ABSTRACT

In this Paper we are going to study how different Agile teams can collaborate with each other. Furthermore, we will also be answering two main questions;

* What is important when an agile team works together with other teams?
* Why is Design/ Architecture so important when doing this?

For our research we will be looking at different research as well survey papers. The expected result after conducting the research is to find and write, what are the possible obstacles faced when different agile teams work together and what can be done to solve these obstacles.

1  What is important when an agile team works together with other teams?

When Agile is performed at a small scale to develop a software the process is rather easy as there are less variables to deal with. Software development at a small scale using Agile thus is recommended and makes the process more streamlined and efficient. This is also makes each and every team member accountable and tracking everyone’s progress is also easier.

While Agile at small scale is easy to maintain scaling this same process at a much larger industrial or business level is very difficult. The reason for that being the number of variables that come into the picture. With such a large scale a single SCRUM team cannot be formed and multiple teams are required. While developing a separate team for a particular task (for e.g. UI design) might sound simple enough and more approachable this also has a downside to it. The more the number of teams in the development the harder it is to manage them.

By this we mean that each team has its own SCRUM master and now all the SCRUM masters have to work in collaboration with each other to make sure a seamless development of the software. Sometimes parts of development can be interdependent (for e.g. The Front end team may be dependent on the UI design team to provide a basic skeleton frame for the product to work with initially). Such interdependence on each other and a decentralized structure of the development teams results in deadlocks. Also it becomes a lot harder to track each team members progress and the general progress of a team as a whole. With each team having it’s own SCRUM master and it’s own targets for a sprint communication becomes vital as miscommunication or improper relay of information can result in cascading failure.

Another aspect of this is that in current dynamic of software development collaborations between different teams in a business are highly recommended by business owners but rarely do we see a proper plan to follow for the development and how different teams would interact with each other. Businesses tends to over rely upon org charts and individual SCRUM boards and mistake it for an “Agile” approach.[1]

To further discuss this issue there is a lack of proper point of contacts between different teams. By that we mean a person or an entity who can relay information between two different teams, who can understand both the teams well enough to properly manage them. For example while developing a speech therapy software there needs to be a proper point/s of contact who can relay information between the Neural Network team and the Speech therapy team.[1]

Another issue is creating departmental silos. This results in a team with members solely expert in one field. Although this might work in a project development on a smaller scale with fewer objectives to achieve. At large scales each team needs a diverse and multi-disciplinary team.[2]

One more way the efficiency of a collaborative Agile process can be increased is by having at least bi-weekly team meetings. In which team members from all the teams can sit together and discuss with each other the problems faced by them and also develop possible solutions. Also, this allows for more clear and effective communication between teams as now teams know each other and recognize each other thus promoting a better collaborative environment.[2]

Building on this point studies such as Santos Goldman and Souza (2015) mentions the importance of intra -team as well as inter team Knowledge sharing for a more streamlined and efficient Agile Software Development (ASD). Although they acknowledge its importance it was noted that this is still an ongoing issue as there aren’t any strategies or frameworks that support inter team Knowledge Sharing. As such they discuss two different approaches namely Codification and Personalization. They discuss the shortcomings of all the different strategies developed in these approaches. While Codification doesn’t really work in the current scenario there are some interesting strategies in Personalization that provide at least part of the solution. A common term SCRUM of SCRUMS keeps popping up in all these strategies. All of them while focusing on different aspects of it do concur that for an efficient ASD at a larger scale a proper communication channel between different teams, SCRUM masters and business owners is required. Thus treating the whole system as a SCRUM team and different SCRUM teams working together in it.[3]

In Paasivaara, Maria, Behm, Benjamin, Hallikainen and Minna (2018) a case study is done on the Agile Transformation at Ericsson and also Dikert, Paasivaara and Lassenius. Mentions the obstacles faced in such large scale transformation. These were Unwilling to changes, Lack of guidance from literature, reverting to old way of working and a general misunderstanding of agile concepts. [4] [5]

The same case study found that with effective coaching, proper support, customization of the agile approach based on the team and work as you learn were some of the strategies employed by Ericsson that resulted in a successful transformation of such scale. [4] [5]

In Paasivaara, Lassenius, Heikkila (2012). The conclusion drawn is that Scrum of Scrums works poorly if the meetings have too many members in it. Which brings us to the initially points pointed out by us that there is a strong need of proper points of contact between different teams who can communicate and relay information between two teams properly.[1]

2  Why is Design/ Architecture so important when doing this?

The next question is what is the role of Design/ Architecture in a large-scale Agile development?[7]

Now our findings suggest that having a proper Design/ Architecture, a proper plan to follow for the software development is equally if not more important than proper communication channels in the Agile Software Development.

Planning and designing of SCRUMs in a small-scale agile software development is much easier. Unfortunately, the same cannot be said about the same process at a large scale. With such a large-scale planning and designing phases can become long and arduous and can also result in teams ignoring this process altogether. This can result in a mismanaged approach in a development process that requires proper planning and highly strategized.

It was noted that such attempts without proper planning failed[6]. In Batra, Xia, VanderMeer and Dutta (2010). A detailed research was conducted in this specific issue where this problem was looked at from several perspectives.

Finally they developed a framework that was best suited for designing and planning of a software development in real life large scale agile software development.

It basically divided the project characteristics in two main categories;

* That require structure.
  + Strategic impact
  + Large
  + Complicated
* That require agility
  + User requirements that keep changing.
  + Unanticipated issues or problems during development.

This process involved creating a proper strategy for the first category while the later one kept evolving and changing as per needs. This resulted in software development that was structured as well as Agile. It met the User requirements (that were evolving or changing) and also the budgetary constraints.

REFERENCES

[1] How to Make Sure Agile Teams Can Work Together Alia Crocker Rob Cross Heidi K. Gardner May 15, 2018

[2] <https://tech.gsa.gov/guides/Collaboration_Across_Agile_Teams/>

[3] Fostering effective inter-team knowledge sharing in agile software development Viviane Santos, Alfredo Goldman, Cleidson, R.B. de Souza 2014

[4] Large-scale agile transformation at Ericsson: a case study Maria Paasivaara1 ·Benjamin Behm1 · Casper Lassenius1 · Minna Hallikainen2

[5] Challenges and success factors for large-scale agile transformations: Asystematic literature review Kim, Dikerta, Maria, Paasivaaraab, Casper, Lassenius 2016

[6] Inter-team coordination in large-scale globally distributed scrum: do scrum-of-scrums really work? Paasivaara, Lassenius, Heikkila (2012)

[7] Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry Batra, Xia, VanderMeer, Dutta 2010.

Conference Name:ACM Woodstock conference

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