

Self-organizing Scrum teams - Challenges and Strategies

Background

Self-organizing teams is a challenging concept, especially when moving from traditional project management to Scrum. Unlike traditional teams, the self-organizing empowered teams are not directed and controlled from the top; rather they evolve from team members participating actively & collectively in all the Scrum practices and events.

This challenge is not only for the teams themselves, but even for the management who finds it difficult to reconcile need for predictable outcomes with the demand for greater freedom and autonomy.

Benefits of self-organization mainly arise from enhanced sharing and learning. This learning can happen from peers & seniors during frequent interactions encouraged by Scrum. Learning also takes place by observing and exposure to cross-functional teams with diverse skills and backgrounds. The frequent interactions also lead to close team bonding and identification with project commitments. Thus the most essential requirement of Scrum teams is to have active participation by all team members in all Scrum events.

Challenges

As mentioned above, we need to look at two distinct sets of challenges; those that arise while assembling the team during the initial adoption and later while sustaining the self-organization to attain successful transformation.

The set of challenges associated with initial adoption can be summarized as,

- **Team Inertia** – Initially the teams are quite enthusiastic about agile adoption due to its novelty as also because of the freedom it provides. However, soon the challenges start surfacing and the common response is to fall back to the old ways of working. Result is to cut corners and adopt only what is convenient. This is a short-sighted approach because the results are even worse than where we came from.
- **Personality types** – In the workplace, we meet different kinds of people. Some are loners, who are happy pattering away on their keyboard all day without surfacing for any meetings or social interactions with their co-workers. On the other hand, there are socializers who can spend endless hours interacting, discussing and comparing notes. Similarly we have aggressive people, as well as passive people in one single team. In traditional project management, we have the project manager to manage these differences. In a self-organizing team, we expect the team to manage itself. While diversity is good, having incompatible people together is a sure recipe for failure.
- **Stakeholders' mindset change** – One of the toughest challenges for the stakeholders is to understand, accept and practice a different behavior. As per Scrum, once the sprint has started you can't change anything unless it is going to help the sprint goal. This is essential because the team is held accountable for the product backlog that they have agreed to take up. Any deviation may have dire consequences, such as lower team

morale or sprint failure. In the traditional methodology, the top executives and stakeholders were able to request changes at any time as the need arose. Since, Scrum takes away this freedom; this will lead to serious conflicts, unless proactive steps are taken right from the beginning.

Scrum provides a well-protected and secure environment for the team to work in. It says that the self-organizing teams choose the best way to work. And though people outside the team may share their well-meaning advice, the team ultimately has to decide what makes sense to them and why. Though this is highly empowering for the team members, it may also lead to these external stakeholders feeling a loss of control over the team as they are pretty much used to telling the project teams what to do, by whom and by when.

- **The Big Shift** - Scrum places low importance on the top down hierarchy as it gives greater autonomy and freedom to the team. Scrum Master is considered a servant-leader by Scrum. There is no corresponding role in traditional management. This is a big shift, not just for the person chosen to be the Scrum master, but for all those involved. Any error in choice of the person to play this role can have serious consequences. Similarly, it is critical to prepare all others to understand & accept this role and adjust their interactions with him accordingly.

The second set of challenges associated with successful transformation can be summarized as,

- **Personality Clashes** – Though a project team can be quite large in size, each of the Scrum teams is small to facilitate interaction and collective participation in all Scrum events. These teams are also cross-functional, that implies that the developers, testers, technical writers, etc. have to work closely as a team. The team members who have previously worked on non-Scrum projects may bring with them their old prejudices and notions. We may get developers, testers and technical writers into the scrum team hoping they all will work in close coordination with each other. However, the developer may feel greater affinity with another developer, and a tester with another tester. This segregation was partly reinforced by the hierarchical structures. This can potentially create clashes. We all have seen that the developers and testers in a Scrum team find it difficult to work with each other as they still expect the work to be handed over to them in a formal manner. But the intimacy and speed that Scrum insists on just does not leave room for such luxuries.

Clashes can also come up due to the work profile. The testers can test only after the developers are done with code. In case of short sprint cycles, any delay from the developers can put undue pressure on the test engineers.

- **Geographically Dispersed Team** – In case of multi-location teams within the same organization, the main problem is of reduced interactions. Though high quality communication systems do help to some extent, it is not same as being face to face at the same location. As the saying goes, "out of sight out of mind".
- **Multi-organization Team** - In offshore software development, sometimes the Scrum team may comprise multiple vendors. In this case, the team is distributed not only across locations but also through organizations, cultures and time zones. Scrum says that the Product Owner is the only person accountable and responsible for managing

the product backlog. This of course, necessitates his close interaction with the team. The challenge comes when the Product Owner and the team belong to different organizations or cultures.

- **Reduced measurements & status reports** - Traditionally, measurements and status reports form the major source of information on which we act to take corrective actions and keep things under control. Actually lot more information is available and in a far more up-to-date state. However, it is tough to change our ways and we continue to expect the information through the old channel and feel frustrated when it does not come.
- **Perceived Drop in Efficiency** - One of the most common arguments against having all the team members participate in the important Scrum events is that it could be time consuming, thereby impacting the productivity of individual team members. Both the team and the stakeholders believe in this misconception. This soon translates into reduced participation and involvement with only one's own work. This is completely at odds with the needs of self-organizing teams and is enough to kill the spirit of self-organization.
- **Absence of individual accountability** - Scrum says that the selection of product backlog for the sprint is solely up to the development team, and not on any particular individual or external stakeholders. It also mandates that the Scrum Master should ensure that the development team has a daily standup meeting; but should not conduct the same. So essentially, the Scrum Master's role is very limited in terms of accountability. It is the team who is accountable and therefore, there is no way you can have one particular person accountable for any tasks or goof-ups. And this can potentially develop into a very serious problem for those who are responsible for the success of the project.

Solutions

The key to the above challenges is provided in the latest Scrum guide 2011, where Ken Schwaber and Jeff Sutherland make an important distinction between rules of the game and strategies for success. Similar to the game of chess, where there are few rules which all players must abide by and there are many strategies developed and shared by individual players. This approach can help us to successfully manage the initial adoption and subsequent transformation so that we can derive full benefits of Scrum. Let's try to see which of the rules specified in the Scrum guide can help us in relation to the self-organizing teams. For rest of the situations, I would propose some strategies which I have found to be quite useful in my work as the Scrum coach for a variety of our clients and project teams.

To successfully manage the initial adoption, it is important that the essential rules of Scrum articulated by the authors of Scrum guide should be clearly made known to all concerned and they should be enrolled to understand & accept the importance to follow those rules. Some of these rules are,

- Scrum roles, artifacts, events and rules are immutable.

- The Product Owner is one person, not a committee. The Product Owner is the sole person responsible for managing the Product Backlog and he remains accountable. For the Product Owner to succeed, no one is allowed to tell the Development Team to work from a different set of priorities, and the Development Team isn't allowed to act on what anyone else says.
- Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule.
- The Scrum Master is a servant-leader for the Scrum Team.
- The work remaining and date are the only variables of interest. The Development Team tracks this total work remaining at least for every Daily Scrum.

Some examples of strategies found useful to meet the challenges during the initial adoption are given below,

- Though the organizations may be tempted not to follow all the practices mandated by Scrum and thereby adopting a watered-down version of Scrum, the leadership should not give in to such temptations. Intensive & extensive exposure to all concerned about benefits of following the Scrum rules will help to get them enrolled. It is also useful to identify persons who are difficult to convince and work on their specific areas of resistance to get their buy-in.
- Choose the Scrum master carefully. Main quality or attitude to look for in the person is whether he enjoys working with people and helping them. An egoist would not help. A person with deep-rooted beliefs in command & control culture may also prove to be a liability.

Another requirement for a Scrum master is deep knowledge & understanding of Scrum. As the Scrum guide says, "Scrum is simple, easy to understand but extremely difficult to master". So a scrum master may not be a master of Scrum to begin with, but he should have an open mind and desire to move towards mastering Scrum.

- The product owner needs to have a free hand to decide what goes in the product and when. There will be intense pressure on him from sales & important customers to include their needs on priority, sometimes at a very short notice. The top management needs to watch out for such interferences and nip them in the bud before they are allowed to become precedence. It is good to review the reporting relationship for the product owner. In one case, where the product owner reported to the sales director, he was continuously under pressure. Just a change of reporting to COO solved the problem.

Apart from the Scrum rules useful for initial adoption, there are others which are particularly helpful to sustain self-organization for the team.

- In the Sprint planning meeting, the plan is created by the collaborative work of the entire Scrum Team. The number of items selected from the Product Backlog for the Sprint is solely up to the Development Team. Only the Development Team can assess what it can accomplish over the upcoming Sprint.

- By the end of the Sprint Planning meeting, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.
- The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Scrum Master enforces the rule that only Development Team members participate in the Daily Scrum. The Daily Scrum is not a status meeting, and is for the people transforming the Product Backlog items into an Increment.
- The work remaining and date are the only variables of interest. The Development Team tracks this total work remaining at least for every Daily Scrum.
- Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis.

Some examples of strategies found useful to sustain self-organization while meeting the challenges during the transformation phase are,

- Review the hierarchical structures for developers and testers. If they are creating silos which result in communication & decisions moving up-and-down the hierarchy, it helps to flatten the structure. This calls for redefinition of roles for upper levels of this hierarchy so that they act more as domain experts & mentors for the team members belonging to their fraternity.
- In case of geographically dispersed teams, apart from the obvious solution of improving the technological facilities for transparent communication, it helps to create opportunities where members from different locations get a chance to connect beyond work topics. This immensely helps to build rapport and a shared language.
- To tackle the problem of non-availability of the product owner to the team as and when required, it is useful to let a senior member of the team play the role of proxy product owner. He may not have the authority to change priorities but he can be an effective bridge between the product owner and the team.
- Sprint planning meeting is the time when the team goes into the details of product backlog items proposed by the product owner for the coming sprint. And though it may appear that it would be more efficient to carry out this activity in a smaller group, the collective participation of all team-members facilitates enhanced knowledge sharing. This should be encouraged as the team can eventually come up with a much better estimation & plan. The same logic applies to all the other Scrum events like daily Scrum, Sprint review and Sprint retrospective.
- The solution to lack of individual accountability really is in ensuring that each team member identifies with the team and considers himself as a part of a unit. He must consider the team's success as his individual success and the team's failure as his own failure. Active and collective participation by all members in all the team events is critical to create this bonding within the team. And though the team is likely to forego this very important practice, the management should firmly support, rather insist on carrying on this activity. The daily standup meetings and the sprint-end review & retrospective meetings should be effectively used to curb the practice of passing the

buck. These blame games should be highly discouraged since building a right culture from the very beginning is crucial. The collective team responsibility means that members question and put pressure on their peers to take ownership of various tasks, The Scrum Master would also need to identify the weakest link within the team and get them in line with the other team members or if required, rotate them outside the team.

Summary

Self-organizing team is an important concept of Scrum. There are many challenges both while assembling the team during initial adoption as well as to continue supporting it during the transformation.

Scrum guide separates the rules from strategies and defines the rules which are mandatory for those using Scrum. One can evolve strategies to deal with specific contexts and situations. Thus the framework of rules & strategies can help us to overcome these challenges.

The session identifies the challenges and explains with examples how the rules and strategies can be used to meet them.