Self-Organizing Teams: What and How

Nitin Mittal, *Accenture*, 7 January 2013

Do you have a self-organizing team? If so, half the battle is already won. But if not, beware: Creating a self-organizing team is far more challenging than we had ever imagined. This is especially true in today's dynamic world, where the focus of building a team can get sidelined due to changing business demands.

Defining self-organizing teams

A group of motivated individuals, who work together toward a goal, have the ability and authority to take decisions and readily adapt to changing demands. Let's look at some important ingredients of a self-organizing team:

- They pull work for themselves and don't wait for their leader to assign work. This ensures a greater sense of ownership and commitment.
- They manage their work (allocation, reallocation, estimation, re-estimation, delivery, and rework) as a group.
- They still require mentoring and coaching, but they don't require "command and control."
- They communicate more with each other, and their commitments are more often to project teams than to the ScrumMaster.
- They understand requirements and aren't afraid to ask questions to get their doubts clarified.
- They continuously enhance their own skills and recommend innovative ideas and improvements.

Five essentials of self-organizing teams

- Competency: Individuals need to be competent for the job at hand. This will result in confidence in their work and will eliminate the need for direction from above.
- Collaboration: They should work as a team rather than as a group of individuals. Teamwork is encouraged.
- Motivation: Team motivation is the key to success. Team members should be focused and interested in their work.
- Trust and respect: Team members trust and respect each other. They believe in collective code ownership and are ready to go the extra mile to help each other resolve issues.
- Continuity: The team should be together for a reasonable duration; changing its composition every now and then doesn't help. Continuity is essential for the team.

Creating a self-organizing team

Who takes responsibility for making a self-organizing team? Is it the ScrumMaster, who is also focusing on timeline and delivery; or senior management, who are concerned about financials; or the organization itself, which has bigger challenges to ponder?

In fact, it takes a combined effort from all three entities to help a team emerge as self-organizing. The ScrumMaster should act as a coach and ensure that the team gets job-specific training and coaching. He or she is primarily responsible for ensuring a cohesive and soothing working environment, which is a must for the blossoming of a self-organizing team. Coaching individual team members is important so that they understand the principles of self-organizing teams and trust each other. The ScrumMaster should also look at a variety of ways to improve collaboration within the distributed team (using application lifecycle management tools, video conferencing, and so on).

Senior management should ensure that they don't get in the way of the team's work; they need to act as supports rather than distractions. Team goals should be set and individual heroism should be discouraged — it only acts as a block to self-organizing teams. It's essential to allow team members to fail before they're expected to deliver.

The organization as a whole should ensure that it provides the necessary infrastructure, training, and incentive system to keep employees motivated at work. These are hygiene factors that constitute the first barrier to cross before you can expect your team members to be self-organized.

A three-step process: **Training, coaching, mentoring**

Training
First step: We need to groom employees to get the desired skill set. At the end of this phase, you can assume the team has the capabilities to exhibit self-organizing behaviors. Provide any needed classroom and on-the-job training to make each employee competent in a particular domain/technology. Behavioral training is also helpful.

Coaching
Second step: Once the team starts working together, adopt a coaching style to see if the members are facing any difficulties. They may require more support and guidance at the beginning. As noted earlier, some indicators of a self-organizing team are: Scrum ceremonies are productive, the team enjoys the work and members help each other, new ideas are forthcoming, and teams are pulling work for themselves. By the end of this phase, you know the team is self-organizing. However, keep your eyes open to observe the team's behavior and provide need-based coaching. This is the phase that will result in innovative ideas and improved results from the team.

Mentoring
Third step: Once the team is in self-organizing mode, the key is to sustain this for the longer run. Assign mentors who can help the team go to the next level. Job rotations can be an important aspect of keeping employees involved and of encouraging continuous learning. As mentioned earlier, a self-organizing team doesn't need "command and control," but it does need coaching and mentoring.

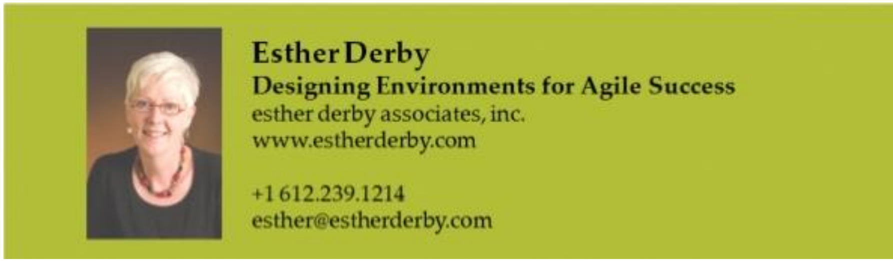
Teams aren't static; they change over time. Building a self-organizing team is an ongoing process, and we're really never done. Whenever a team's composition changes, we need to repeat the whole process.

- See more at: https://scrumalliance.org/community/articles/2013/january/self-organizing-teams-what-and-how#sthash.O9eeu2dF.dpuf

---

Misconceptions about Self-Organizing Teams
July 19, 2011, Esther Derby

At a recent conference, I over-heard three managers talking about self-organizing teams.



Esther Derby
Designing Environments for Agile Success
esther derby associates, inc.
www.estherderby.com

+1 612.239.1214
esther@estherderby.com

"You can't just turn people loose and let a team make all the decisions. They'll mess things up. And with all these ScrumMasters, coaches, and self-organizing teams, sounds like I'm out of a job," said one with resignation.

"This time boxing thing is great," said another. "Put them in a room, turn up the heat, and they'll perform," said a second manager.

"Wow, this means I can move people around based on projects, and they'll just form and self-organize. I can have rolling, ad hoc Scrum teams!" crowed a third.

Time to rectify some misconceptions.
# 1. Self-organizing teams are completely autonomous, self-managing, and don't need managers.

2. All you need to do to form a self-organizing team is provide a goal and apply pressure.

#3. Since the team is self-organizing, they can accommodate moving people on and off the team easily.

Misconception: Self-organizing teams don't need managers.
There's a reason we use the term "self-organizing" rather than "self-organized" or "self-managed." That's because it's a process and a characteristic, not something that is done once and for all. Self-organizing, from a social systems perspective only means that the team can create new approaches and adapt to meet new challenges in their environment.

Self-organizing Agile teams do have–bounded—authority to make their own commitments, organize and assign their own work. They craft appropriate strategies to accomplish their goals, and make decisions with (again bounded) economic and organizational impact.

But they are not out there on their own, disconnected from the organization. Self-organizing teams exist to produce a product or service that is valuable to the organization and its customers. They are accountable to make their progress visible, and work within financial boundaries. Self-organizing teams may also be self-managed, to one degree or another.

Managers must create the conditions that enable teams to thrive and continue to self-organize. Manager need to work across the organization to create a work system that enables teams to deliver value to customers and the organization. And managers need to work with the team to set appropriate boundaries and constraints. Managers still act as agents for the corporation. Therefore, they still must be involved where there are legal or fiduciary responsibilities.

Misconception: Time boxing forces any group to become a team. Put a group of people together and hand them a challenge and they'll gel.

I wouldn't bet on it, and neither should you. Teams do need a clear and compelling work goal. Without that, there's no reason to form a team. They also need the technical skills required by the work and interpersonal skills to work as a team. They need resources such as tools and access to information and education. They need a connection to the larger organization.

The pressure cooker method of team formation will more likely burn people out than result in the productivity of a real team. Calling a group a team and turning up the heat, doesn't make it so.

Time boxing is one of the structures than can help teams succeed by providing focus. Working in time boxes creates a natural rhythm of feedback and connection to the team's purpose. But a time box and goal, in and of themselves, don't create a team.
Misconception: Self-organizing agile teams should be able to accommodate frequent membership changes. After all, they're agile aren't they?

Teams need time to develop the strategies and trust that enables high performance. They need time to understand each other's strengths and weaknesses, develop shared knowledge, and learn how to learn together.

When new people constantly arrive and leave, a group may never develop the shared approaches and shared knowledge that permit them to outperform a group of individuals.

Some teams—when they've had time to form and create a strong team culture—do become adept at adding new members. Even then, it's best to limit the number of new members added at any given time. Changing more than 30% of team membership causes the team to reboot. Constant turnover prevents a team from truly forming.

As a manager, you can help by keeping teams together long enough to gel, and by protecting teams from the revolving door syndrome.

Reality
Self-organizing teams are not teams gone mad. Like all teams, they need a compelling goal, skills, information, and enough time to form and perform. And they still need managers to create a supportive context, set appropriate boundaries and constraints and connect the team to the organization.

Scrum and the self-organizing team
http://www.billharlan.com/papers/Scrum_and_the_self-organizing_team.html

Scrum seems to have become the world's most popular Agile development methodology. We find Scrum practices codified in team roles, iterative schedules, and project-management software.

One of the essential features of Scrum is often forgotten -- the self-organizing team. Without it, Scrum is just structure and process, without any agility or productivity gains.

The authors of Scrum have made it clear in all their books that self-organization is fundamental to their design. Below, they emphasize this point in their definition of a Scrum Team.

From the "Scrum Guide -- the official rulebook" (1991-2011) at http://www.scrum.org/scrumguides/ , by Ken Schwaber and Jeff Sutherland:

§ The Scrum Team
The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

§ The Product Owner
The Product Owner is the sole person responsible for managing the Product Backlog... The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a backlog item's priority must convince the Product Owner.

§ The Development Team
The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness. Development Teams have the following characteristics:

- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;
- 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;

- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole; and,
- Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis.

§ Development Team Size

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work. Fewer than three Development Team members decreases interaction and results in smaller productivity gains... Having more than nine members requires too much coordination.

§ The Scrum Master

The Scrum Master is responsible for ensuring Scrum is understood and enacted...

The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.

The Scrum Master serves the [Scrum Team] in several ways, including:

- Finding techniques for effective Product Backlog management;
- Clearly communicating vision, goals, and Product Backlog items to the Development Team;
- Teaching the Scrum Team to create clear and concise Product Backlog items;

- Coaching the Development Team in self-organization and cross-functionality;
- Removing impediments to the Development Team's progress;