

No Group is a Team on Day One

Activities to Help You Get Started

© 2011 Don Gray

The agile training class for a newly formed team was almost complete. We'd covered values, practices, roles, the product backlog, done simulations teaching the Scrum process and I could see the end of training. A little team building activity, and we could start tomorrow with building the backlog, story sizing, then start the first sprint. Forging ahead, the team selected a name, came up with a list of team norms, and they became a team with me as their ScrumMaster.

Or did they?

Over the next few weeks, I noticed cracks appearing. One member tried to find ways to avoid the daily standup. Another member only checked in code toward the end of the sprint. Yet another happily worked on what he was assigned, but never volunteered to take tasks from the task board. At times, the entire team struggled to understand how its activities blended with and supported the activities of the other teams on the project.

There didn't seem to be a unifying focus, something with which all the team members could identify. What could I have done differently?

Forming a Team

Over the years, I've seen teams formed many different ways. There's the Five You process where the manager goes "You, you, you, you, and you. You're now a team!" Often, these team members were selected because they were between projects, they possessed approximately the right skills, and it was time to get a new project started.

Matrixed teams form somewhat differently. Team members get assigned by skill level to projects. The greater (or rarer) your skill, the more teams you get assigned to on a part-time basis. This way, everyone has 100 percent of his time assigned to project work. I once met a QA "resource" (which in itself says a lot about the company) assigned to three Scrum teams. For some reason, he didn't seem to get much done but go to meetings. Matrixed team assignments guarantees all projects get delivered in the maximum amount of time possible, if at all.

In the case of the team I worked with, the developers became a team because they all worked in the same office. I believe in collocation, but being together, in itself, doesn't provide impetus to become a team. So what might?



Don Gray

DELIVERING CUSTOMER VALUE

MENTORING AND TRAINING
for EXECUTIVES, MANAGERS & TEAMS

don@donaldegrey.com
336.414.4645
www.donaldegrey.com

What Makes a Team?

In *The Wisdom of Teams*¹ we find the conditions needed to exist for teams to form:

- A compelling work goal
- Interdependent work
- Stable membership
- Shared history
- Five to nine members

Interestingly, a compelling work goal often gets overlooked. The team had work to do, but I don't believe "We're part of a project that will save the company a bunch of money" constitutes a compelling work goal. Don't get me wrong, helping the company save money is a good thing, but does it call forth the urge to slay dragons or at remove impediments from the development process?

We did have interdependent work, both within the team and between the teams. Other than some members not checking code in regularly, a lot of thought and effort went into making sure the teams' code didn't break other parts of the application. The code management system and continuous integration worked quite well when developers did check in their code.

This team had sort of worked together prior to my arrival. The team members knew each other but divided the work so they had the minimum amount of interaction. So, while they had stable team membership, how they handled work created knowledge silos and prevented them from learning from each other.

Shared history initially seems to indicate the team has to work together for some time. I've learned since my work with this team to use experiential activities to actively start creating shared history instead of letting the shared history happen and hope for the best.

Teams generally need five members. This allows for sufficient skill, experience, and knowledge to generate good decisions and actions. When teams get more than nine members, the communications paths tend to become centralized and the larger group informally splits into two smaller groups, losing a coherent whole.

Creating a Team

Creating a team involves more than picking five to nine people then having them pick a name (although choosing a name for the team can be both fun and enlightening). Managers usually set the initial team membership and pick the project on which the team will work. This means managers need to align the team members' skills so they can fulfill the project's vision. If the project doesn't

have a vision statement how can the team coalesce around a compelling work goal? It doesn't exist.

Team formation activities can range from creating a formal team charter to the Five You process. It may make sense for some teams to go through the formal process of documenting the purpose; summarizing their structure, customers, and stakeholders; listing the members; identifying roles; and defining the team's authority—but I don't usually see this in agile software development teams.

With the team members selected we can do specific activities to help create team identity, highlight the team's purpose and establish ground rules.

Team Identity Activity – Design a Box

This activity lets the team define itself and create an identity to share with the rest of the organization.

Team members need to design a shrink-wrap box that represents their team. They design the box front and back. This involves:

- Coming up with a team name
- A picture
- Three to four key bullet points on the front to "sell" the team
- A detailed feature description on the back
- Operating parameters

It also tends to be a lot of fun.

Team Purpose Activity – Elevator Statement

This activity focuses on the “compelling work.” It aligns the team's actions with the product vision and corporate mission statement. It's the team's reason d'être.

Using the “elevator speech” format helps the team succinctly pull these ideas into a sentence they can share with others.

For [target customers] who want/need [solution to a problem] <your team> provides <what compelling reason to buy> Unlike <competitors/other teams> <the team key differentiator>

Working Agreements

Working agreements specify the team's operating parameters. They list expectations between the team members. They serve as guide rails as the team builds its shared history. They might include

meeting information, team norms, a definition of done, and other pertinent items.

When working with the team to create its working agreements, I use the expand-then-collapse facilitation process. After explaining the goal for the working agreements, I solicit suggestions from the team members about what they feel will be important to them. I use stickies for this. Then we post the stickies, having a conversation about what the words mean. After we have everyone's input, we look for duplicates, suggestions that form part of another suggestion. We aim for seven to nine agreements. Anything more results in a list no one can remember.

I'd be lying if I told you these activities automatically create great teams. Nothing can automatically create great teams. But, by focusing on your team's purpose, identity, and working agreements you can help team members answer the question "Why am I here?" and set the team up for success.

¹ The Wisdom of Teams, Katzenbach and Smith, Harper 2003, ISBN 978-0060522001 pp 43-64.

You can find a number of my articles published at Better Software magazine, StickyMinds.com, AgileConnection.com and techwell.com.

This article was originally posted on techwell.com on March 9, 2011