Tim Leslie · Follow

5 min read · Dec 4, 2019

1

There’s a big difference between a team and a working group.

Unfortunately, creating a team isn’t quite as easy as gathering a group of people and making sure they all have well-defined specific tasks. It’s difficult to overstate the productivity gains and employee happiness you’re missing out on if this is all you think there is to a team.

The distinguishing feature of a real team is mutual accountability. Members of a team genuinely care about their collective goals whereas members of a working group care more about their own tasks. Members of a team will jump to help each other out and work collectively to deal with any problems. Really strong teams also have members who are deeply committed to each other’s personal growth and success.

If a working group fails to achieve its goal the typical internal monologue is "I did my bit and I deserve recognition for that. It wasn't my fault we failed. The goal wasn't really achievable anyway and I always knew the way we were going about it was all wrong." Whereas, if a *team* fails the typical internal monologue is more like "We tried really hard and I thought we had a good shot at success but we didn't foresee problem X."

Teams are particularly beneficial in software engineering where the complexity of the work can be extremely high. It's not possible for any one engineer to have a perfect understanding of a typical commercial codebase — it's simply too big, with too many moving parts. Knowledge is necessarily dispersed and communication is essential. It brings huge benefits if helping your colleague isn't a "distraction", if you can bounce ideas around, if you're able to learn from each other every day, if you can share your experience of the codebase and if there isn't a massive hole left in your institutional knowledge every time someone moves on.

—

Sounds great but how do we actually make it happen? Real teams are rare and require a lot of nurturing. Two excellent books*, The Five Dysfunction of a Team* and *The Wisdom of Teams* were seminal in my own understanding of the problem and I'd highly recommend them to anyone with an interest in the topic.

If members of the group are going to commit and become mutually accountable for a goal then they can't go into it thinking there are obvious flaws in the approach. If they're not given real input to the plan and a genuine opportunity to challenge each other then they're not going to get past this. This is the environment that needs to be in place for a team to form:

- A common goal
- Autonomy to decide how to achieve the goal
- Group members that are able to challenge each other in a constructive way.

Of these, constructive challenge is the most mysterious and hardest to achieve. There are two things to watch for that indicate that constructive challenge *isn't* happening.

Firstly, some members of the group become entrenched in a position. They refuse to listen to counter arguments. Being right is very important to them.

In extreme cases, they may even sabotage the team to prove their point. In these cases there may be a personality problem which is why we consider humility to be fundamental for a successful career at Float. We think it's incredibly important to avoid hiring big egos — no matter how good they are technically.

However, it's not always about personality and you should look deeper. Very often this is a symptom of an environment where people don't feel comfortable being vulnerable. They think they'll be judged for being wrong and that they'll lose influence. The other members of the group are seen as competitors instead of allies and friends. In other words, a lack of trust often underlies this behaviour.

Secondly, other members of the group may not speak up at all. They may be too shy to say what they really think and too ready to say they agree even when they don't. They don't want to look stupid by saying the wrong thing. What's interesting is the root cause of this behaviour is often the same. They're not prepared to show vulnerability because they lack trust in the other members of the group. They're scared they'll be judged if they say something silly and they don't feel like the rest of the group has their back.

There's no such thing as a team made up of people who don't trust one another. More specifically the members of a team need to trust each other so they can show vulnerability. They need to show vulnerability so they can challenge each other in a constructive way. And they need to challenge each other so they can commit and become mutually accountable for the collective goal.

—

Unfortunately, instructing the members of a working group to trust each other probably isn't going to help! (Though explaining some of the theory in this article certainly won't do any harm.) Fostering trust is difficult. I don't have all the answers but I do have some advice.

Firstly, lead by example and show vulnerability yourself. Whether you're in a position of authority or not, showing that you trust the people around you is a really good way to earn trust. Part of this is honest and open communication, part of it is admitting your mistakes and part of it is speaking up and exploring half-formed ideas with the rest of the group.

If you're in a position of authority it's incredibly important to treat the team as a unit. It seems obvious, but if you're obsessed with judging and measuring individual performance you'll create an environment where individual performance is more important than team performance. If a team is performing well after you've given them a chance to settle then call the team up on any problems that do arise and give them a chance to work together to find a resolution. You should hold the *team* accountable and the members of a team should hold *each other* to account. However, if the team is consistently underperforming then it's time to dig deeper and start questioning whether it's specific individuals that are getting in the way, or a different issue.

Most importantly, it takes time for trust to build. Team members need to go through forming, storming and norming before they start performing (Tuckman's stages of group development). Don't constantly change the group or they'll never form a team!

Leadership Cto Software Development

Written by Tim Leslie

Follow

39 Followers

CTO at Float (www.floatapp.com).

More from Tim Leslie