



Workgroups vs. Teams

by **Holger Paffrath** MVB · Apr. 24, 18 · Agile Zone · Opinion

Adopting a DevOps practice starts with understanding where you are in the implementation journey. Download the DevOps Transformation Roadmap. Brought to you in partnership with Techtown.

You know the drill. Your manager or someone higher up says "We are all one Team," but something doesn't quite sit right with you.

You may be all one team, but no one works together. Everyone is working on their own thing. They are working on tasks that are not related to what you are doing. Your task isn't related to what they are doing. There is no coordination. No discussion, just work. Are you really on a team?

The most likely answer to this question is "No." You are not on a team. Despite what your manager says or his manager or anyone higher up. You are not working in a team.

Businesses like to group people into teams, and rightly so. Teams are powerful entities. Teams helped put men on the moon. Managers know this, they are not stupid, but due to decades of misinterpretation, culture, and misunderstanding, the real concept of "team" has been lost.

So, what are you if you are not a team? Quite simply put, you are in a workgroup.

Isn't a workgroup a group of people working together? Isn't a team a group of people working together? Aren't they the same thing?

Well, not quite. All Teams are Workgroups, but not all Workgroups are Teams.

Well, to put it another way, there are some fundamental differences between workgroups and teams. According to Jon Katzenbach and Douglas Smith in their HBR paper "The Discipline of Teams," there are some differences.

Let's go through these one by one.

Strong, Clearly Focused Leader vs. Shared Leadership Roles

With a Workgroup, you basically have your head of the group who is in charge. This is someone who directs those under him or her what to do. They are in this position all the time. Everyone underneath obeys, at least in the extreme case.

In a shared leadership role, everyone collaborates. Sometimes one person directs, sometimes another person. The person most appropriate—the person with the "expertise" at the time—is in charge. Sometimes there is

no one with expertise and discussion ensures. Sometimes the "team" as a whole decides.

DevOps Webinars On-Demand

Either way, everyone has a say.

Continuous Delivery: A Pragmatic

Uncover the Hidden Value of

Crack the Automation Code and

Approach to Mitigating Risks

Your CI/CD Pipeline

Create a Balanced Test Strategy

Individual Accountability vs. Individual and Mutual

Accountability

Watch Now

Watch Now

Watch Now

With a workgroup, you are pretty much on your own with the task you are working on. If you mess up, you are to blame. If you succeed, you get the benefit. Achievements and failures are your own.

With a Team, you are accountable for the work you are currently doing, but the team is there to support you. Failure means that the team as a whole has failed, not the individual. It is the team's responsibility to help those in trouble. If there is success, it is the team as a whole that succeeds. Not an individual. Think of a team sport. It is not the individual that won or lost, it is the team. The whole team.

The Purpose: Organisation's vs. Team's

With a workgroup, the group's purpose is the same as the organizational purpose. In other words, there is no purpose other than the default one or the organization. The group itself just is there to work.

With a Team, the whole point is to accomplish a goal. A goal that the whole team has agreed upon. It is specific to the work the members are currently doing. The best purposes are the ones greater than the team. One that if the members all work together, have a hope of achieving as opposed to one for each individual.

Individual Work Products vs Collective Work Products

In work groups, everyone is working on their own. One member is working on project A, another member is working on Project B and so forth. Each is on their own.

In a Team, everyone is working towards the same goal. So, everyone is working on Project A. Everyone is working on Project B. How they accomplish that is decided by the team.

Runs Efficient Meetings vs. Encourages Open-Ended Discussion and Problem Solving Meetings

For Workgroups, meetings are efficient. This may seem like a good thing, and it can be. But the reason meetings are efficient is that the leader dictates the meeting. Knowledge flows from the leader down. Very rarely does it flow up. Even when it does, the leader still makes the decision. This is not necessarily a bad thing. It can be quite good.

With Teams, problems are discussed. The team as a whole works out the solution through consensus. Each person has an opportunity to speak and be heard. Knowledge is disseminated from all individuals to all other individuals. This also gives the team as a whole ownership.

I'm reminded of the meeting Waigaya where some meetings are discussed this way. A Waigaya is a meeting where every member leaves their title at the door. No hierarchy; in fact, it is forbidden and frowned upon if someone does use their title. Everyone has equal say. Important points are discussed until a mutual solution

by all parties has been found. Meetings can be quick—5 min or less, or can last a long time. For hairy topics, they can go on and off (and on and off) continuously for years. This may seem like a waste of time, but Honda puts more emphasis on getting to the best solution rather than "a" solution and based on the quality of their cars and motorcycles, and I think it works. With an efficient meeting, a decision may be made quickly, but it may not always be the right one.

DevOps Webinars On-Demand

[Continuous Delivery: A Pragmatic Approach to Mitigating Risks](#)

[Uncover the Hidden Value of Your CI/CD Pipeline](#)

[Crack the Automation Code and Create a Balanced Test Strategy](#)

Measures Effectiveness by Its Influence on Others vs. Measures by Assessing Collective Work-Products

Workgroups measure based on outside factors that. For example, the performance of the company as a whole. How can they do it any other way? Individuals are working on their own. On their own little work areas.

With a Team, everyone works together, so the measurement is based on the collective work that they have done **together**.

Discusses, Decides and Delegates vs. Discusses, Decides and Does Real Work Together

Workgroups, as discussed earlier in meetings, have everything sent down from on high. The members may discuss together, but the leader is the one who makes the decision and then delegates to the members.

With a team, there is no real leader like in the sense of a workgroup. The team collaborates together and does the work together. There is no one person always coordinating activities and delegating.

As you can see, with Teams, there is more of a mutual relationship. There is more collaboration between members. Everyone is working towards the same goal. Be it complete a product, win a game or solve a problem.

With a Workgroup, everyone works on their own thing. It requires coordination. Therefore there is someone in charge to coordinate, direct and delegate.

Is the Team Always Best?

No, Teams and Workgroups do have their places, but the best place for a team is when performance is required.

Now, you are probably thinking that your group or team is somewhere in the middle. That may be the case.

Marie J. Kane categorizes Workgroups and Teams into different levels.

Work Group - As we have discussed above.

Pseudo Team - Not focused on collective performance, or trying to achieve it but it is needed. These are the weakest type of group.

Potential Team - There is a performance need. It is trying to improve, but has no purpose or goals.

Real Team - Its members have complementary skills and are working together towards a common goal. They

hold themselves accountable. The performance impact is significantly higher than a workgroup.

DevOps Webinars On-Demand

High-Performance Team - Like a Real Time, but also committed to everyone's personal growth and success.

Continuous Delivery: A Pragmatic Commitment to Overcome the Hidden Value of Form all **Crack the Automation Code and**
Seamless Deployment **Approach to Mitigating Risks** **Your CI/CD Pipeline** **Create a Balanced Test Strategy**

Now, these differences cannot be taken individually. For example, just because members help each other occasionally, **Watch Now** mean they are a team. Working group members help each other **Watch Now**

When Teams Do Not Work

Ok, so you found out that you are not on a team. That is not necessarily bad. Workgroups are a working form of group that is better performant than a Pseudo Team. A Pseudo Team is trying to be a team, but only doing it half-baked. This actually turns out to be worse than being in a workgroup. This can happen if the "team" doesn't even know what it is doing. If a team doesn't agree at some point, things can stagnate, stall or be paralyzed from working. Getting agreement is a leader's job and having a weak leader, one not willing to take professional risks in the team direction, can make it possible that the team won't do a good job.

Then there is the problem where the team is always in agreement all the time. This creates another type of stagnation. One of lack of innovation. Having some conflict in the team of a technical nature, not a personal one, is good for the team. Alternative views are given. These alternative views must be explored and tested, not shot down, otherwise, there is a lack of trust between team members.

Then again, sometimes teams that do have conflicts of a personal nature do thrive. Just look at the Monty Python team. They hated each other, but each member knew that the others had their back no matter what when it counted. If your team is getting to the point where there is no conflict, and there is no innovation, then it might be time to switch things up a bit. Swap members with other teams. Bring in new blood with alternative views—a dissident who will "rock the boat." Get the teams' synapses thinking differently. The dissident also needs your support. They veer from the norm at great personal cost. No one wants to be the outsider, get shut down, or get told to shut up. So rather than let them be silenced, let them speak out and listen because, without them, your team can degrade to the point of being mediocre, but also make sure the dissident has the teams best interests in mind. You don't want delinquency for the sake of it.

Another reason that Teams do not work is that they are not necessarily faster than workgroups. For the simple nature in teams where discussion between team members is encouraged, teams can actually be slower than workgroups. The trade off, though, with teams is that they produce better.

Teams may not work when they are large. Large teams have to communicate with one another. The more members, the more communication channels are required, the longer it takes to disseminate information, to discuss, the slower a team gets. Teams that are small. Between 3-9 members are ideal.

Only having a "Real Team" or "High-Performance Team" will give you the full benefits of having a team. Just saying that a group of people is a "Team" is not good enough to get the benefits. So, I request that if what you really want is a Work Group where you direct and delegate to members, call it a Work Group and not a Team. You are only confusing matters further and perpetuating the confusion with the term.

One more thing DevOps Webinars On-Demand

Just to quote the late great Steve Jobs, there is one document that I have left to last on purpose as Teams are not specific to Agile. That document describes how a team should work and it describes it very well. That document is the Scrum Guide. It mentions that Teams should be Self Organizing. Team members should be cross-functional. In other words, have all the skills to complete the goal.

Teams are altruistic in intentions. There is no hierarchical structure. Every member at least for the Development team has the title of "Developer." There are no sub-teams.

The Scrum Master is there not to direct and delegate the team, but to support the team. Play interference and remove impediments from the team that prevents them from succeeding. No one but the team itself directs the team.

Accountability belongs to the team as a whole, not to individuals.

Jeff Sutherland and Ken Schwaber knew what they were talking about when they wrote the teams section in the Scrum Guide.

References

The Discipline Of Teams

Jon R. Katzenbach and Douglas K. Smith Harvard Business Review

Organisational Work Groups and Work Teams - Approaches ad Differences

EcoForum - Volume 4, Issue 1 (6), 2015

Turn Your Group Into a True Team

Harvard Business Review

How To Distinguish the Important Differences Between Teams and Work Groups

Executive Evolution

WAIGAYA: HONDA'S SECRET TO SUSTAINED SUCCESS

<http://quarterly.insigniam.com/wp-content/uploads/2014/11/IQ-Waigaya-Hondas-Secret-to-Sustained-Success.pdf>

Why Teams Don't Work

Harvard Business Review

The Scrum Guide

Take Agile to the next level with DevOps. Learn practical tools and techniques in the three-day DevOps Implementation Boot Camp. Brought to you in partnership with Techtown.

Like This Article? Read More From DZone

DevOps Webinars On-Demand



Continuous Delivery: A Pragmatic Approach to Mitigating Risks
Watch Now

Uncover the Hidden Value of Your CI/CD Pipeline



Crack the Automation Code and Create a Balanced Test Strategy
Leadership Lessons for Creating High-Performing Scrum Teams
Watch Now



What You Should Know When Building an Agile Team
Watch Now

Watch Now



Free DZone Refcard Agile Patterns
Watch Now

Topics: AGILE , TEAM , WORKGROUP , SCRUM MASTER , AGILE LEADERSHIP , AGILE TEAMS

Published at DZone with permission of Holger Paffrath , DZone MVB. [See the original article here.](#) Opinions expressed by DZone contributors are their own.

Agile Partner Resources

Free e-book: Agile Product Development for Dummies

1 Gang

|

Urban Guide for Scrum Teams

Scrum.org

|

Continuous Integration Boot Camp

Udacity Training

|

Agile at Scale: 3 Must-Have Tools for Enterprise Agile Teams

1 Gang

|