How cross-functional should my Team be?

In a Lean/Agile world teams should be cross-functional. Period.
Or maybe it's not that easy? What does cross-functionality actually mean?
And is it true that the more cross-functional, the better?

By Stefan Roock and Arne Roock

What do Ocean's Eleven, The A-Team, The Avengers and the group around Manny the Mammoth in Ice Age all have in common? It has to do with interdisciplinary Teams – or „cross-functional Teams", if you will. Each team member brings different skills to the table, all of which complement the skills of the others, making it possible to cope with a variety of tasks and find innovative solutions.
Just like in the movies, this principle also applies to the business world. Like Takeuchi/Nonaka describes in his 1986 article (see [1]), Cross-functional teams in business offer two advantages:
1. Speed
2. Innovation through the mutual exchange across disciplines („cross-fertilization").
Cross-functional Teams are faster because they do not rely on the legwork of other teams, and tend to tackle and solve problems right away themselves. And this is exactly where innovation tends to arise – when thinking steers down a different path than it normally takes. This out-of-the-box thinking is more likely to occur, the more different disciplines that work together.

In the world of Agile, especially with Scrum, cross-functional teams play a very important role. Ken Schwaber and Jeff Sutherland wrote in their Scrum Guide [2]: „Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment" (see [2])
.

Don't lose sight of the Target

But when is a team considered to be cross-functional? The question isn't that simple to answer – you must consider what it is that the team is actually trying to accomplish. With Scrum, the minimum goal of the team is to provide a finished product increment with every sprint. This product increment is finished according to the Definition of Done, which is set by the Scrum Team. It may be that it is enough to have developed some new functionality and to present this into a test environment. Then, for this Scrum Team, you would need developers and probably testers. However, if the Definition of Done is further-reaching and establishes that the new functionality must also be deployed on a live server, then Admin-skills will also be required. But maybe even this is not enough, and the team also needs text-translations or clarification on legal issues. Then, its possible that an editor or an attorney should be included in the team. Aside from the expected result, the Input-side (in Scrum often referred to as Definition of Ready) plays another important role: Does the Team get really specific requirements that they should implement – along with the respective designs? Or, at the beginning of the Sprint, is the design not yet set in stone and the requirements still a little vague? In the second case, if the team is only made of developers and Testers – then there will probably be problems. Team members with Design-know-how are needed – and it may even be a good idea to have a business analyst. Potentially you could need more and more members on the team, until sooner or later, the team can actually no longer function properly as a team. The important thing to note here, is that we don't need a specialist for every possible skill in the team. The necessary skills must be present, and when a skill is only necessary every now and then, then a less-skilled person would ▶

perform a task when needed. In this context, we're talking about T-shaped Skill-sets.
Every team-member has his specialty and this is also his preferred type of work. He has, however, skills in other related areas – but in these areas, he would not work as efficiently as a specialist. It is still more efficient than doing nothing in his own specialization when he's not needed. So a developer could be a Java programming specialist and at the same time be able to program HTML to create database tables and test- may just not quite as efficient as a specialist in this area.

Alternative Team Target

We'll leave the Scrum world for a moment and just have a look at teams in general. It is by no means always the case that every team's goal is to deliver product increments. What would happen if a team received one of the following orders:
a) organize an economically successful conference
b) within 120 days, build a prototype for a new phone that has the potential to replace the iPhone
c) rob three Las Vegas casinos at the same time
This short list can show how completely different cross-functional teams look, depending on the goal they pursue. In the first scenario, you would definitely need extensive marketing experience in the team, and it would perhaps be useful to also have an accountant. To successfully complete contract b, hardware and software experts are definitely needed, with different specializations. And as we learned from „Ocean's Eleven", for mission C , it would be handy to have an explosives expert, a pickpocket, at least two drivers, a safe-cracker, a computer hacker, and – oh yeah – a contortionist.

> So to break it all down so far: A team is then cross-functional if it has all the skills necessary to achieve its goal. And depending on this goal – the composition of a team can differ considerably.

Team Membership

To move forward, we should briefly address the question of what exactly it means to be „in a team". One possible answer is: If you belong to the team, the team goal is the top priority at all times – meaning you may need to drop everything if you are needed by the team. Belonging to a team is then primarily a matter of prioritization. This is why it is so problematic when employees participate in more than one team. They are often caught in a conflict of priority.

> Our second learning then: To be a member of a cross-functional team, means that all tasks relating to achieving the team goal, automatically have priority.

Who is the Team?

Suppose for a moment that we are dealing with a team that is developing new features for a web-platform, and releasing them live. Occasionally, a lawyer will be needed

because of legal issues that may need to be resolved before a new features can really go live. Does a lawyer really need to sit around in a team the whole time, waiting to answer questions? Does he need to attend all of the team meetings? And when he's not needed, is he able to, instead, do something else – even if it does not directly serve the goal of the team? This is certainly an extreme example – and also certainly not very economical. In this case, the lawyer would probably not be a team member. However, the team must have the ability to quickly obtain this legal know-how – or have some immediate access to a lawyer. Problems could arise in this situation, however, when many teams are all depending on one person at the same time.

> Here we find our third nugget: Cross-functionality is not an "either-or" decision, but instead a consideration. On one side we have speed and innovation, on the other, efficiency.

Cross-Functionality and Innovation

As we briefly touched on in the beginning of this article, one of the great advantages of cross-functional teams is that they tend to be more innovative than teams in which only one or a few different disciplines are represented. In general, we might say: The more innovation that you want, the more cross-functionally composed your team should be. There is only one problem: increasing cross-functionality, sure, increases innovation – but at the same time, increases inefficiency. Coming back to our example: Lets say a team needs, not only a lawyer, but also an accountant (even if only for once a year), and an Online-Marketing expert, and a specialist in encryption technology – the team members would often have either nothing to do, or spend most of the time working outside of their actual work specialty. Not to mention, the team is getting bigger and bigger, which also means inefficiency in most cases.

The relationship between innovation and inefficiency can be illustrated using this simple graph:

On the far left could be a purely mono-functional team, for instance, a team of graphic designers. The goal of the team certainly wouldn't be the development of product increments, but instead creating graphics. Because the team members would only work within their specialization, this team would be very efficient and could implement many ▶

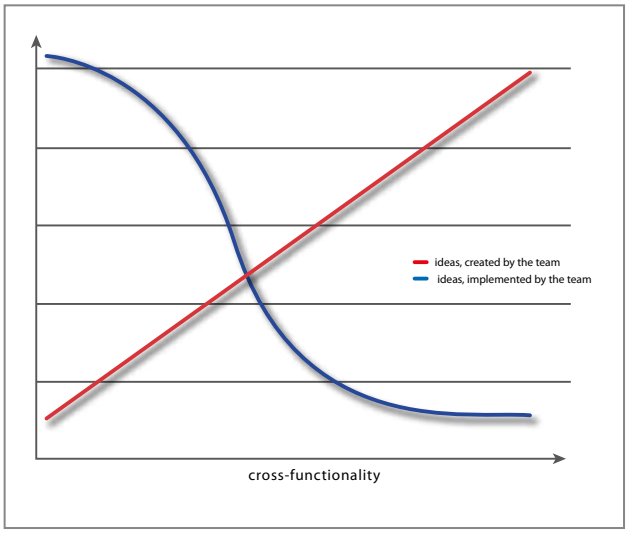

Figure 1. The tension between innovation and efficiency

externally predetermined ideas. But this would not be particularly innovative, and wouldn't create many new ideas. Tim Brown, the head of the design firm IDEO says „Design's too important to be left to designers". Typically, a lot of like-minded people are not especially creative.

At the other extreme of the scale would be the team which has all of the skills you could ever imagine. In addition to graphic designers, programmers and testers, there would also be brain surgeons and freedivers. On the far left could be a purely mono-functional team, for instance, a team of graphic designers. The goal of the team certainly wouldn't be the development of product increments, but instead creating graphics. Because the team members would only work within their specialization, this team would be very efficient and could implement many externally predetermined ideas. But this would not be particularly innovative, and wouldn't create many new ideas. Tim Brown, the head of the design firm IDEO says „Design's too important to be left to designers". Typically, a lot of like-minded people are not especially creative.

At the other extreme of the scale would be the team which has all of the skills you could ever imagine. In addition to graphic designers, programmers and testers, there would also be brain surgeons and freedivers.

This diversity of the team members would have a huge potential for innovation, however would be extremely inefficient in implementing the ideas generated (just imagine the brain surgeon pair-programming with the freediver to develop an iPhone App.). We buy innovation through inefficiency. The question is no longer: „Do we want cross-functional teams", but „How much innovation do we need?" or „What are we willing to pay for additional innovation?". Even though we'd really like it, we can not be simultaneously super-innovative and super-efficient. And the curve raises another interesting question: Should our team, as a first priority, implement finished ideas (wherever these ideas come from)? Then it tends not to be necessarily cross-functional. Or should the team generate new ideas? Then there should be cross-functionalty – but at some point they won't be able to implement all of these ideas themselves.

Cross-functional Teams and Speed

Let us now turn to the second great advantage of cross-functional teams: speed. When a team has all the skills necessary to achieve it's goal, then it tends to be very quick, because it doesn't have to wait around for the external legwork. It can instead react immediately to unforeseen events. In this regard, it is useful to have teams be as cross-functional as possible. Unfortunately, here again there is an opposite trend of inefficiency when considering the entire company. Cross-functional teams generate a huge focus on the goal, which however only leads to short-term results. At the same time, however, this focus tends to set everything else aside which doesn't directly serve the achievement of the goal. If our lawyer is always making himself readily available to the team, with their work as the highest priority, then inevitably he would have to let some of his other tasks slide. The same applies here, to find a balance between the two opposing movements, as illustrated in the following graph.

Once again, our mono-functional team is represented on the left in the figure. It is very efficient – however, it con-

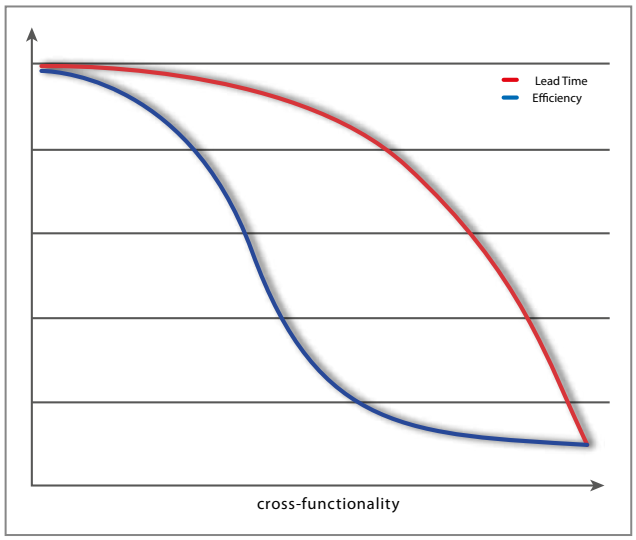

Figure. 2: Cross-functional teams tend to be fast, but inefficient.

tributes very little to the overall value chain. One cycle through the value chain involves many mono-functional teams (Silos) and leads to a very long lead time. On the far right we see again the all-encompassing team. This team never has to wait around for outside support, and therefore has a minimal turnaround time. Then again, this is paid for through inefficiency.

This figure may shed some light on the first problem, but there is another problem to address. Because "efficiency" and "lead time" are measured entirely differently, they cannot be reasonably compared with each other. It is difficult to determine from these conflicting efficiency and lead time just where the appropriate position is, to decide how cross-functional the team should be.

Cross-functional Teams and Cost

Here it is helpful to take a look at the work of Don Reinertsen [3]. Reinertsen discusses in his books these reoccurring problems, where the optimum in two opposing movements can be found. His clever answer entails converting both values into a common unit – and this unit in most cases is „cost". Costs can be directly monetary, in the sense that someone has to write an invoice. In most cases, however, there are also "opportunity" costs: Because we completed tasks later (or never), we missed out on potential revenue.

When considering cross-funtional teams, we are essentially dealing with two types of costs: One cost is a result of the low utilization of team members, or members working outside of their specialization – they are tied to the team and can't actually perform any other tasks in the meantime, and so spend at least part time working outside of their specialty. The other cost the team might incur is when the team has to wait for external groundwork and is therefore blocked. The value that the team is trying to produce is therefore delivered later than it potentially could have been. Besides that, the feedback loop is delayed. The team learns slower than it otherwise could have.

Through this image, we get a more distinct picture. The question of whether we need cross-functional teams or not is no longer the main issue. Instead, the question that comes forth is; how can we set our teams up so that we achieve the minimal total cost? It would be great if we could just assign exact numbers to the curves and easily calculate the cost. In most cases, however, this would end up being too expensive – or even impossible. Fortunately, we don't actually need these exact calculations for the first step. The first ▶

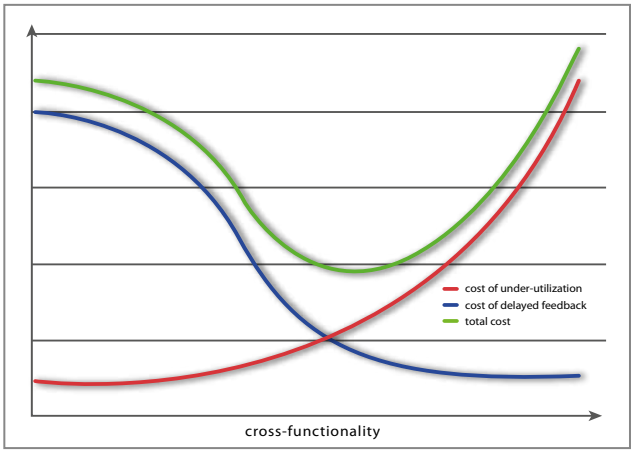

Figure. 3: Cost-curves can be helpful to determine the right level of cross-functionality.

question we should ask ourselves is; „Are we on the left or right of the cost minimum?" The vast majority of the teams and organizations that we know, are very clearly too far left. They are not organized cross-functionally enough. The reason is, that often we only look at the red curve: When teams are poorly utilized, it is immediately apparent. Therefore, we usually optimize the utilization. Most hardly ever even consider the green curve: How many (opportunity) costs are incurred because our lead time is longer than it should be? If we look at the curve of the total cost, we notice that the line only slightly curves to the minimum. This is good for us because it means that we don't have to meet the minimum exactly, but just land in the vicinity. Now that we have our team set up so that we can work at the lowest possible total cost, we have taken the first step. The second step is to reduce the total costs. What measures can we take to drive the blue or the red (or both) curves down? As for the red curve, just put a little thought into possible measures that could be taken to reduce inefficiency:

- Consult a Moderator or Coach: He helps the team members to find the positive factors in the resulting friction in a team. He also provides assistance in organizing the work in the team, so the team can work as efficiently as possible under the circumstances.
- Break down the work into smaller packages: Often a sequential workflow in one work package cannot be avoided. Only after a function is programmed, can it be tested. The result is that in the beginning, the testers may be underutilized and at the end, the graphic designers. If you cut the work down into smaller tasks, you can reach a uniform demand for the existing specializations, which is then a more efficient use of the available skills.
- A second pillar: In the diagram of the T-shaped skill-sets, each team member has a specialization. However, there are also companies which require its team members to have two specializations – you could call it ϖ-shaped (Pi-shaped) skill-sets. So we can also increase the efficiency through targeted training

(although, this measure requires a fair amount of time before the goal is reached).

The way to cross-functional Teams

Scrum makes it easy. Cross-functional teams are simply formed and then, away we go. But, this procedure is not always readily available for some organizations, and sometimes also not even necessary. As with Kanban, there are no initial requirements made, regarding the composition or ▶

even the cooperation of the team. It is rather like the Kanban principle "Start where you are". If, for example, a team made of only testing experts starts using Kanban, then initially it will just continue to work within it's specialty – and use Kanban to improve its process step-by-step. Eventually though, the question is raised about how the team should be assembled and how it should work together. As in, who should be at the Standup or Retrospective Meetings?- and how do new tasks fit into the Kanban-system? Or when is the ticket really finished? The theme of cross-functionality is not only relevant to Scrum, but really in all team contexts.

Conclusion

Cross-functional teams are faster and more innovating than mono-functional teams. However, cross-functionality is not an either-or decision. There is a tension with the work efficiency in the team. The appropriate level of cross-functionality depends on the goal of the team. The more innovation and more learning required, the more cross-functional the team should be. In practice, most teams are not cross-functional enough, because too much emphasis is placed on efficiency. Accordingly, it is not wrong for teams to experiment with gradual steps towards cross-functionality. Parallel to this gradual approach to optimizing the cross-functionality of the team, it makes sense to invest in reducing the associated cost: It is often very useful to include a coach or moderator, and to invest in expanding the capabilities of the team members.

References

[1] Hirotaka Takeuchi, Ikujiro Nonaka (1986): The New New Product Development Game. Harvard Business Review, Januar 1986.

[2] Jeff Sutherland, Ken Schwaber (October 2011): Scrum-Guide.

[3] Donald G. Reinertsen (2009): The Principles of Product Development Flow. Second Generation Lean Product Development. Celeritas Publishing.

About the authors

Arne Roock

Dr. Arne Roock is working as a Trainer (accredited by LKU) and Coach for Lean and Kanban at it-agile in Germany. He trained and coached Kanban in different contexts, such as Web Startups, Major Product Companies and international Enterprises. He wrote several papers on Lean/Kanban and translated David Anderson's Kanban book into German. He runs the Blog www.software-kanban.de and founded the first Limited WIP Society Germany.
email: arne.roock@it-agile.de
Twitter: @arneroock

Stefan Roock

Since 1999 Stefan Roock has participanted in Dozens of projects as a developer, coach, consultant and trainer. He has in-depth experiences with Scrum, eXtreme Programming, Feature-Driven-Development und Crystal. He approaches new projects with a „methodology per project" mindset and combines practices from several methods for the situation at hand. Stefan wrote books about eXtreme Programming and Refactoring and is a Certified Scrum Trainer (CST).
email: stefan.roock@it-agile.de
Twitter: @stefanroock