

Scalable Team Structure

Description

The early stages of a tech startup are often panic-filled days with all hands on deck. In those early days, “all hands on deck” often only means a handful of people, and that’s why it works.

The “Mythical Man-Month” is a concept put forward by Fred Brooks in 1975. Intuitively, one would think that, if it takes one person a month to do a task, you should be able to add another person to that task and it take half as long. Add another person, and it will take a third the amount of time. One more, and it should take a quarter the time.

default watermark

What unexpectedly ends up happening is that you reach a point of diminishing returns with each person you add. Very quickly, you’ll find adding more people will actually *slow down* a project. So how does one scale an organization?

The problem of adding more and more people is that it exponentially increases the number of communication channels needed to accomplish things. In technical work, it also increases the likelihood of complexity creeping into a project, which can have costly consequences down the road.

By having very small teams within your organization focusing on very specific mandates, you can regain the benefits of a larger workforce without suffering from the communication overhead.

Team Composition

Depending on who you ask, **the ideal Agile team size is somewhere between 5 and 11 people**. The members of that team includes everyone needed to deliver the work. That includes product management, software development, platform engineering, QA, scrum master, etc. Keep the team size to the minimum needed to get the work done. It encourages minimalistic thinking which maximizes the chances of a fast and quality product delivery which can be iterated on.

Remember that the teams should be as cross functional as possible to be able to deliver as much as they can with the least amount of outside involvement. One arguable exception to this in smaller organizations is platform engineering. While the development teams should be empowered to

implement simpler platform engineering initiatives directly related to their team, it is still reasonable to have a separate team solely focused on general server and network maintenance beyond the scope of any individual team.

Do your best to allow the teams to maintain ownership over their assigned domain. While there will likely be instances where a project would benefit from work done on another team, that work must be a collaborative effort. No matter who actually does the work, nothing should be merged into it without the owning team's approval.

Internal Team Leadership

With a team being cross-functional and cross-discipline, it is critical that everyone takes ownership for the output of the team. They all have an equally valuable voice in all planning endeavours. The individuals on the team may specialize, or they may have a variety of hats. There are still three critical leadership aspects that need to be owned by someone within a team.

Scrum Lead

This person owns the Agile process for the team. They lead daily standups, sprint planning, sprint reviews, sprint kickoffs, and any other Scrum ceremonies your team adopts. This means they adhere to the relevant agenda and scope for those calls, moderating the conversation and ensuring everyone stays on topic. They would even own coordinating the timing of those meetings.

default watermark

Aside from meetings, they also own the commitment of the team. This means they would analyze past sprint velocity to determine the capacity of the upcoming sprints.

Tech Lead

This is the owner of technical quality and efficiency. They lead collaborative discussions about solutions to technical problems, using their expertise to guide the conversation to the simplest solution. A good Tech Lead has enough knowledge and experience to solve problems themselves, but enough humility to accept that other team members, no matter their experience level, could come up with more efficient solutions, or adjustments that lead to a better outcome. There are several talented people on a team, and a good tech lead takes advantage of everyone's knowledge.

Product Manager

Acting as the voice of the customer, the Product Manager guides the customer-facing goals of the team. They collaborate with the team to determine what deliverables they will bring to the customer. Being customer-driven means they should also be interacting with customers as directly as they can, identifying opportunities to solve the challenges the customers face.

Whether these three roles are filled by one person or multiple depends on the resources available to your organization, but if possible, distribute the roles to different people. These are excellent opportunities for career advancement within your organization, and distributing the work means each role is getting the focus it needs.

In software development, Tech Lead and Product Manager are almost always separate roles. Scrum Lead is often a responsibility owned by one of those other leaders instead of an separate individual, but is still best split off if resources allow. Tech Leads and Scrum Leads still typically do actual development alongside their other responsibilities.

Org Leadership

Each of the previous roles outlines an area of domain leadership rather than people leadership. When it comes to things like personal development, salary discussions, cross-team collaboration, higher level planning, someone outside the narrow scope of a single team is needed. For a very small organization, this responsibility could be owned by the **Chief Technology Officer** themself. As the company grows, that responsibility needs to be split if there's any hope of scaling.

default watermark

The CTO could potentially handle one or two teams, but to provide the appropriate attention to the individual teams and cross-team initiatives, more help will inevitably be needed. The first layer to add would be **Engineering Managers**. These individuals would handle the people management aspects of work, including interviewing, training, cross team collaboration, and generally being the voice of upper management and HR to the teams they lead.

As the organization continues to scale, the next layer contains the **Engineering Directors**. Typically assigned a specific product scope, these directors handle higher level coordination. They must juggle the product and technical aspects of the work that impact their assigned domain. They may need to make far-reaching decisions, like changing the technical stack to better suit the needs of fakeywatch their project. They take on the hiring responsibility for their teams, having the Engineering Managers available to assist.

The next layer would be your **Vice Presidents**. Like Engineering Directors, they typically own a specific domain and own high-level and high impact decision making. They also act as assistants to the CTO, interacting with vendors or high-value clients as needed. They are often involved in management-level hiring and mentoring as well.

Measuring Success

If implementing this in an existing org, you should notice a few areas of improvement.

- Cycle time reduced (time from task creation to delivery)
  - Since the teams are smaller and more focused, they're forced into a minimalistic mindset, resulting in simpler solutions delivered faster.
- Defect rate reduced (number of bugs generated over a given period)
  - Defects are often found hidden in complex solutions. Thanks to the "simpler solution" benefit of this structure, those defects should be reduced.
  - The amount of siloed knowledge is also reduced since the communication of that knowledge is less work with a small team. That means relevant institutional knowledge is owned by everyone, making it less likely to cause issues in the first place.
- Costs reduced
  - Poor communication is costly. Whether it's people making bad assumptions, or simply misunderstanding direction. Small teams simplify communication, reducing costly errors.
  - The mythical man month problem is solved by segmenting the work. Now the value gained by each added member is preserved, as long as the team sizes are kept to a minimum.

Each of these points are actually measurable key performance indicators (KPI) and I highly encourage you to track them as you tweak your team structure.

default watermark

On top of the easily measured indicators, you should see morale be preserved if not improve, as everyone is empowered and set up for success. This structure also has plenty of growth opportunities for the team members. Opportunities like that are proven to improve employee retention.

Keep in mind that you should try to keep the structure as flat as possible, maintaining a balance between minimum distance from CTO to developer, while still allowing everyone to focus on what they do best. Be mindful of the well being of your team, and the needs of your customer, and everything will work out!

Category

1. Leadership Principles

Date Created
January 11, 2023
Author
steveday