Group EN04:

In this blog post, we will discuss Team Topologies, in what they consist and the main principles behind them. We will go through their origins and address why they are important to consider in the world of software architecture. After that, we will explore the different strategies and types of teams and groupings that follow the Team Topologies approach, as well as other metrics to take into consideration. Finally, we will go through how we could implement Team Topologies, some of the common pitfalls that comes with it, and how to manage teams in this context.

Team Topologies is, as it is defined by Matthew, the leading approach to organizing for a fast flow value in team-of-teams organizations. It is a concept that came out of the IT space, but it's applicable to all knowledge work. Team Topologies can be seen as an approach to knowledge work.

The model is based on four team names and 3 interaction modes.

This approach started around 2008, influenced by DevOps, which combines practices and tools to deliver applications and services faster. These practices allow software development to proceed more rapidly. The tools also provided developers with a greater variety of options to work more quickly.

The principles of Team Topologies are the following: To generate separate fast flows of value. Deliberately decouple the flows (explained later). To be mindful of team size (small teams, high trust on them). To think about the cognitive or mental load of the team (and its members). To think about the kind of system and organizational architectures. To take care of Conway's Law (communication pathways and system design or architecture). In addition, to expect your technology to become obsolete or disappear in a short period of time.

Cognitive load theory suggests that our working memory has a limited capacity, and new information being presented to the worker comes with a cognitive load because of it. When applying this theory to team topologies, you aim to measure how much cognitive load a team is in, so we can avoid a loss in value flow.

It is important to note that cognitive load can come from different sources in the workplace. To measure it, we use a metric called team temperature. Whenever there is something that introduces complexity, the temperature goes up, and your goal is to keep the temperature as low as possible while maintaining a level of productivity. To accomplish that, you must structure your teams accordingly.

Deliberate desynchronization is a technique used in Team Topologies, that limits synchronization between teams to enhance product value, especially in large projects. It reduces cognitive load on team members. Desynchronization also cuts waiting times and costs, and production value appears sooner. However, excessive desynchronization can lead to fragmented value.

“Elephant carpaccio” exercise is used to illustrate the division of workflows in Team Topologies. Starting from an initial development team in a pilot project, fast flow values are created as their needs arise. Other decisions are taken in parallel as well, allowing to maintain the generation of value active while improving the condition of the development team (reduce of mental load, increase in team facilities).

The 4 types of teams:

Stream-Aligned teams

These are the most common along the companies. Each tea mis attached to a end-to-end value, so a single product or specific scenario. They are small groups but large enough to have the skill to provide value independently of other groups

Enabling teams

These groups are in charge of supporting and helping Stream-Aligned teams. They try to upfill their skills and capabilities. Aswell, they can help them by suggesting new tools or organization wide training. Once they finish helping the team they detach and go on with the next team.

Complicated-Subsystem team

A Complicated-Subsystem Team handles complex systems within a Stream-Aligned Team, reducing its cognitive load. This specialized team is more efficient than placing individual experts in each stream-aligned team, as it prevents distractions from value delivery. It becomes necessary when a Stream-Aligned Team struggles with a complex problem that the Enabling Team cannot resolve.

Platform Grouping

Platform Grouping is a structure of multiple Platform Teams, which each one build and maintain internal platforms by integrating third-party, near-complete platforms, and internal services. It reduces the cognitive load on Stream-Aligned Teams and scales as platform complexity grows.

Focusing solely on team types can lead to misunderstanding the objectives of Team Topologies. It's crucial to grasp the purposes and principles behind it, such as creating platform groupings and different team types, driven by flow and cognitive load. Effective implementation requires reorienting towards value stream approaches, as a flow-based way of working is more effective and humane.

References

https://en.wikipedia.org/wiki/DevOps
https://teamtopologies.com/key-concepts
Conway's Law
Elephant carpaccio
An introduction to cognitive load theory

Complicated-Subsystem team:
(goodreads-quote) (itrevolution-article)
Platform Grouping: (martinfowler-post)
Altassian-team-topologies