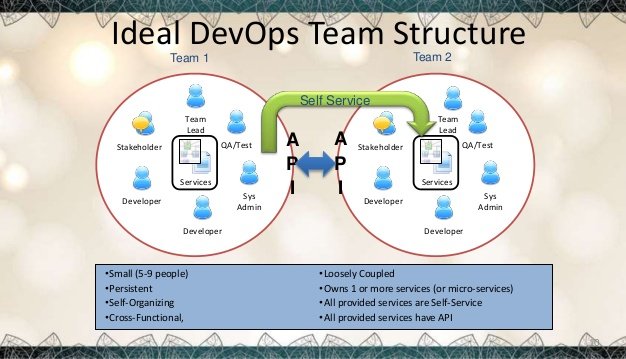
**DevOps Team structure**

DevOps team structure is highly important for any organisation who are implementing DevOps to increase faster and continious delivery for customers.

Conflicting and misleading team structure results into negative outcomes and loss of customers.Hence Heartland and many other companies have realigned their teams around common goals, processes and tools as a necessary step in their successful DevOps transformations.



**Type-1: Collaboration between Dev and Ops Team**

This devops team structure takes once-siloed development and operations functions and redeploys them as tightly woven, highly collaborative teams working side by side.

“Conversations with development went from, ‘oh that’s tough, we don’t know that, we aren’t experts’...to ‘we can change that easily because we know we can run those changes through QA, we can validate them, and treat the infra[structure] as a code base as well,’” Jez says. “And that fundamentally changed what we were able to do as a company, and the speed with which we were able to deliver.”

This approach also maximizes the blend of software engineering skills and deep systems knowledge most organizations need. That combination can be tough to find in a single person, which poses problems for the "NoOps" approach.

**Type-2: Dedicated DevOps Team**

Dedicated DevOps teams are often made up of experienced operations people with a mix of skills including using version control, writing infrastructure as code, and continuous delivery.

DevOps is definitely a team sport, but it's still important to clearly define individual roles and responsibilities for success within the team. Here are some recommendations:

* **IT manager:** Build trust with counterparts on other teams; create a climate of learning and continuous improvement; delegate authority to team members
* **Dev manager:** Build trust with Ops counterpart; bring Ops into the planning process early.
* **Systems engineer:** Automate the things that are painful.
* **Quality engineer:** Provide input into scale and performance; provide feedback on staging environments.
* **Devs:** Plan for deployment as you're planning new features; get feedback from Ops and work with them on deployment process.

Regardless of your organizational structure, DevOps requires collaboration across multiple functions and shared responsibilities for ensuring the best possible quality.