



AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

DevOps

Henrik Bærbak Christensen

Expert People

- From 2015 Master of IT thesis
 - “A typical development team includes UI designers/developers, backend developers, and database experts. [...] and many other companies split the development process into several components, such as UI, backend, database and mobile applications.”
- Why does this make sense?



UI designers

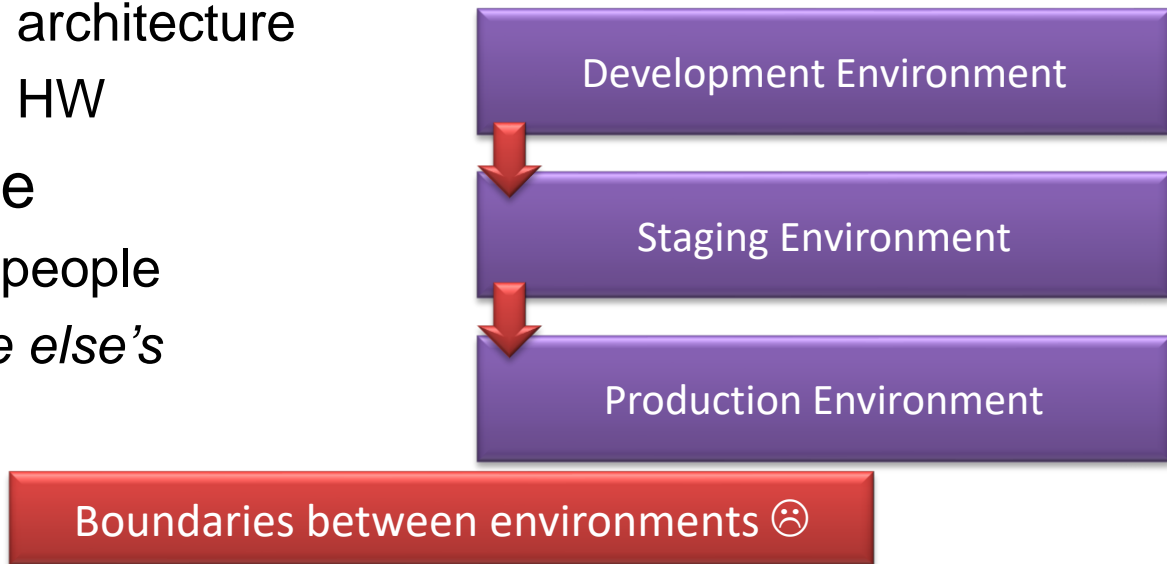
Application Server
developers

Database designers

Boundaries between people ☹️

Moving Code

- Crossing the *boundaries of environments* for the full development-testing-deployment cycle is costly...
 - Reconfigurations
 - Setup
 - Changed architecture
 - Changed HW
- Even worse
 - Different people
 - *Someone else's problem*



- The problems
 - Boundaries between *people*
 - *Is it the fault of the UI guys, or the DB guys*
 - Boundaries between *environments*
 - *But it works on my machine ???*



The DevOps movement, for example, emerged from one of the classic stumbling blocks in a lot of organizations. Developers build code and applications and ship them to the operations people, only to discover that the code and applications don't run in production. This is the classic "it works on my machine; it's operations' problem now."

Definition (?)

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

Bass, Weber, Zhu, "DevOps", 2015

- Key Points
 - Quality (of code) is important. "Suitability" is key quality

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Definition (?)

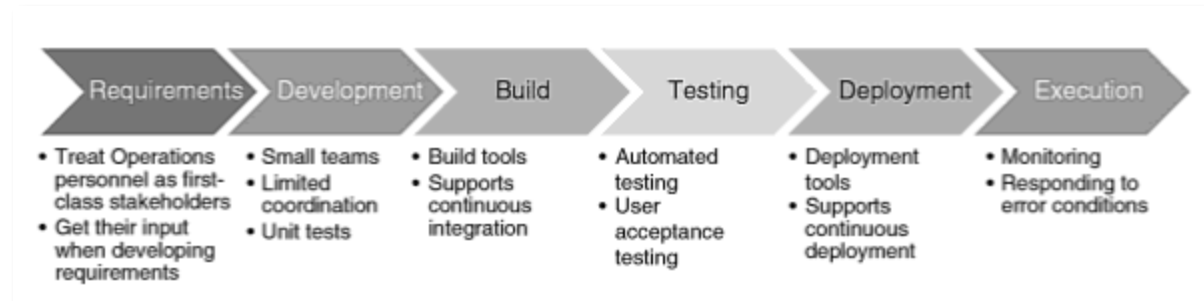
DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

- Key Points
 - High quality *delivery mechanism* – tooling to put change in prod.
- Practices
 - Ops is *first-class citizen*
 - Logging and monitoring coded as part of Dev
 - Dev also involved in incident handling
 - Dev is usually responsible for initial deployment & monitoring

Definition (?)

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

- Practices
 - Enforce deployment process
 - No ad-hoc stuff here... Automation is the name of the game.
 - Continuous delivery
 - Infrastructure-as-code



- Wikipedia

DevOps is a set of practices that combines **software development (Dev)** and **information-technology operations (Ops)** which aims to shorten the **systems development life cycle** and provide **continuous delivery** with high **software quality**.^{[1][2]}

-



Rouan Wilsenach

The primary characteristic of DevOps culture is **increased collaboration** between the roles of development and operations. There are some important cultural shifts, within teams and at an organizational level, that support this collaboration.

Dev & Ops Collaboration



Team culture

No silos

Autonomous teams

Organizational culture

Jabbari et al. defined DevOps as follows:

DevOps is a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance, and delivery with automated deployment utilizing a set of development practices.

² R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is DevOps?: A Systematic Mapping Study on Definitions and Practices," in *Proceedings of the Scientific Workshop Proceedings of XP2016, XP '16 Workshops*, (New York, NY, USA), pp. 12:1–12:11, ACM, 2016.

DevOps Helps to

- ... ensuring that...

- Boundaries between *people*

- *Is it the fault of the UI guys, or the DB guys?*

The same people:
small **DevOps** teams

- Boundaries between *environments*

- *But it works on my machine?*

The 'same'
environment:
small **containers**

Microservices: About getting the *right*
containers...



Discussion

- Have anyone experience
 - Organizing in DevOps
 - Pro/cons