Software Development Process

Lecture 8

DevOps

Short Releases, Then What?

- Agile promotes quick releases, short cycles
  - New features, new releases every sprint
  - Then what?
- System admin has other issues
  - He has to make sure his system is operational
  - He also has to make sure it scales to grow number of users
  - He also has to fix immediate problems so he does not like new, untested stuff
  - So, deployment has to be done with cautious
  - DON’T FIX A RUNNING SYSTEM!

Your release does not run, why?

- To the system admin:
  - The new release does not run because our system is older or we do not support that version
  - The development and production architectures do not match (storage, network, security)
- To the dev:
  - We have tested. The code runs flawlessly
  - Just upgrade your system (OS, infrastructure, etc.)

Your release does run, what next?

- Okay, your release may not be that problematic and it does run on production
- How do we get feedback from the user or the operation staff?
  - Workshops?
  - Interview the users?
  - Error/Issue reports?
  - System logs?

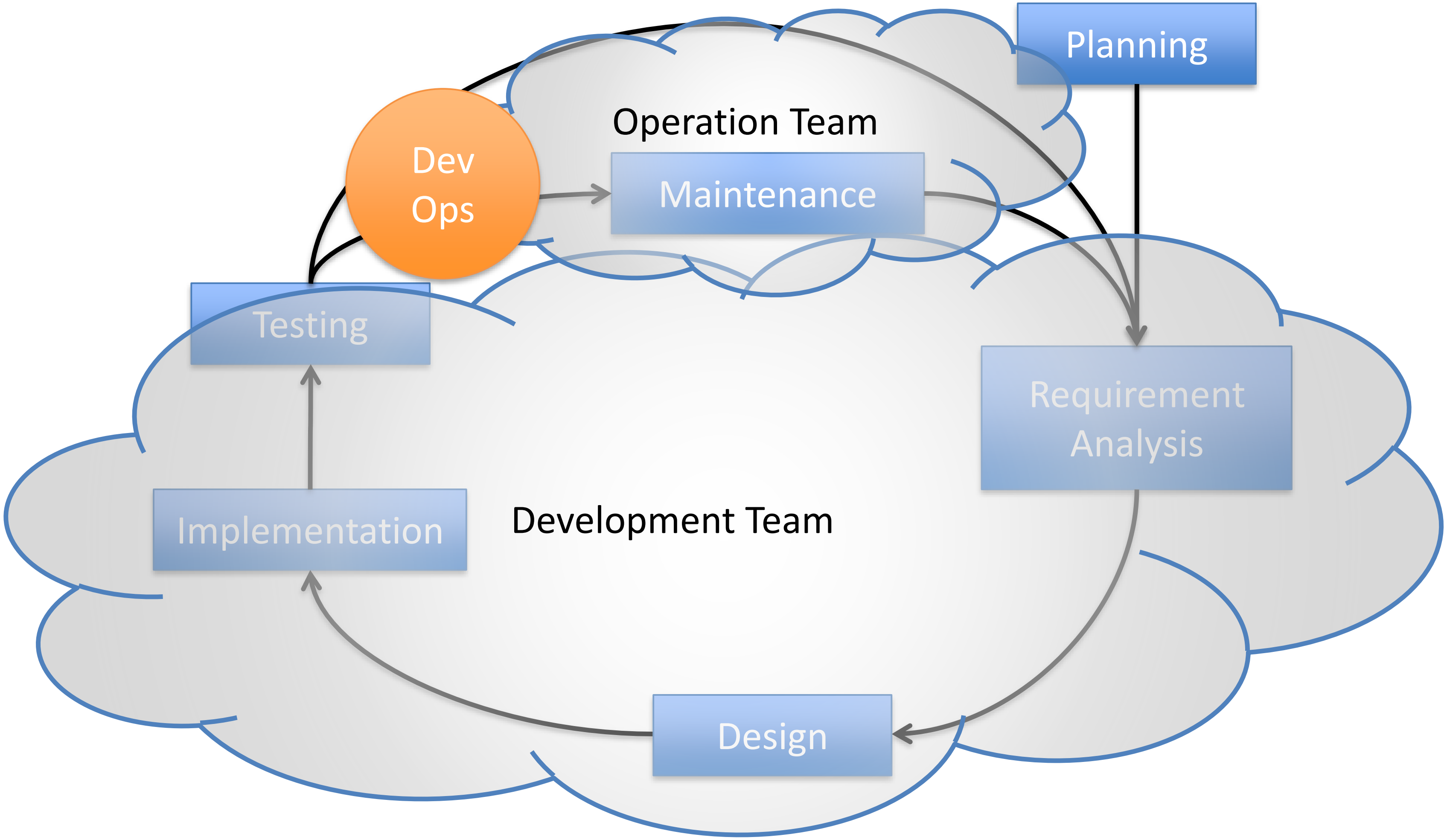
What do have I have to concern?

- Software is not just about **development** and releasing, it is also about **deployment** and maintenance (or **operations**)
- One cannot simple
- throw the product ***over the fence***
- **Dev**eloper must also collaborate with the **op**eration staff to maintain and improve the system

DevOps

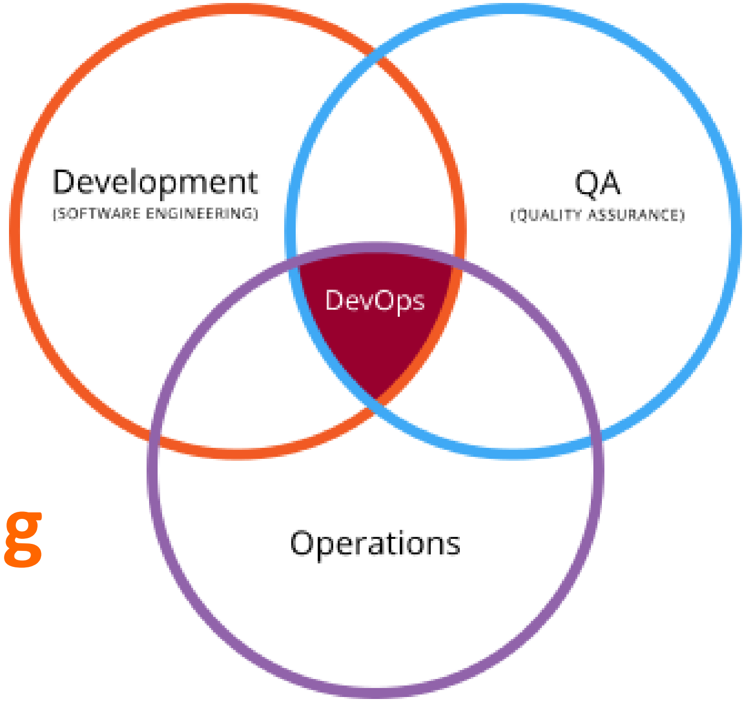
- **DevOps**: The practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support. -- Ernest Mueller
- **Dev**: DEVelopment (and or Developers and everyone that involves in product development)
- **Ops**: OPerationS (and or Operation staff)
  - Admins, release engineers, DBAs, network engineers
  - Everyone that helps running the system

Software Development Phases



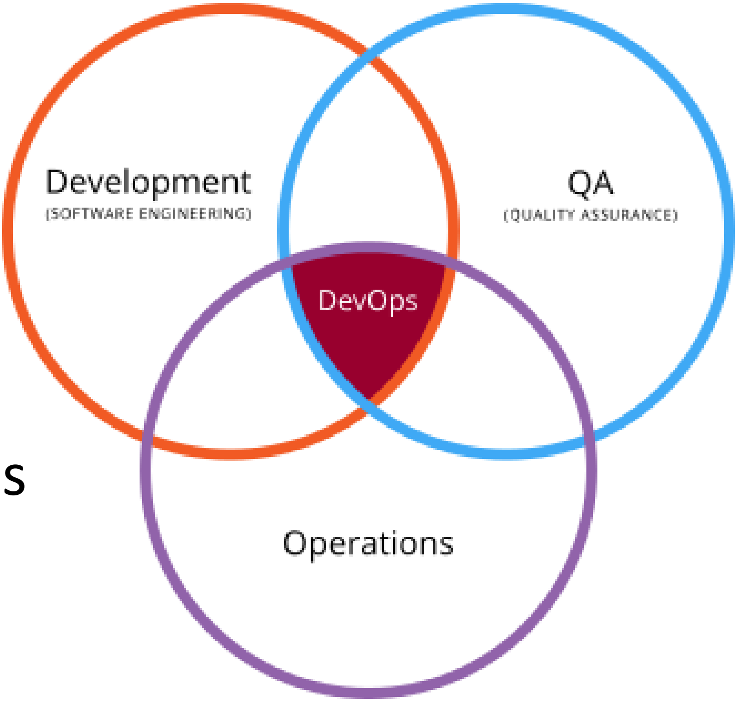
Product, not Project

- DevOps is a concept that integrate **product development, quality assurance** and **operation** together
- The developed software should be thought as **living products**
  - Not just a project that will be delivered at a deadline



Product, not Project (2)

- Your product might be evolving all the time
  - One release a day
- The product must be of certain quality
  - Perform fast
  - Pass UAT
  - Integrate well with existing parts
- It has to be deployed immediately
- Everything should be automated



Blending Dev and Ops

- Development must also care about uptime and user feedback
  - Continuous integration/testing
  - Configuration management
  - Incident command system
- Ops must also care about changes
  - Continuous integration/testing: Could be the same instance as the dev team
  - Infrastructure as Code
  - Use development tools in operation

Infrastructure as Code

- **Infrastructure as Code** (**IaC**): Manage IT infrastructure or data center using code or script rather than physical configuration
  - Aka Programmable Infrastructure
  - Write code to manage your servers
- Since infrastructure-ing is scripted
  - It can be version controlled
  - It can be automated tested
  - It can be configured (multiple times)
  - Knowledge is now in the code, not the admin's head
- Development tools can be used here

DevOps Toolchain

- In software development and deployment, we use several tools e.g.
  – **Jenkins, Travis or Bamboo**: Release management, continuous integration
  – **Vagrant or Docker**: Virtualization and containerization
  – **Git or SVN**: Version control, Source Code Management
- These tools can be used in operation side as well
  – Infrastructure management
  – Hotfixes or bugfixes

Agile and DevOps

- Agile addresses the gap between the customer (and his requirements) and the development
  - Iterative and incremental
  - Cross-functional team (design/dev/test)
  - Focuses on changes (e.g. features)
- DevOps addresses the gap between the development (and testing) and operations
  - Automated release management
  - Focuses on changes and operational readiness

When to adopt DevOps?

- DevOps respond to quick changes and evolution
- Some systems/applications are suitable for it
  - Web-based system (Amazon, Facebook, Strava)
  - Cloud-based system (Paas, Iaas)
  - New features could/should be added quickly
- Some systems do not change much
  - Banks, Factory-controlling software,
  - ATM machines
  - Smartphone OSes, Computer OSes