

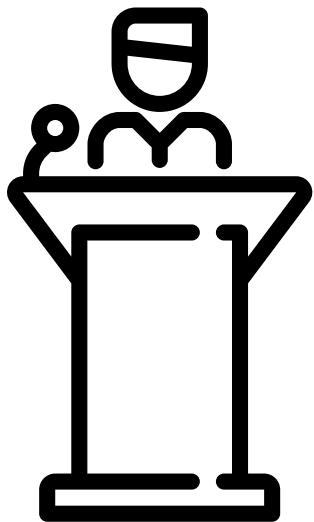
**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP

BERLIN, 26-28 SEPTEMBER 2016

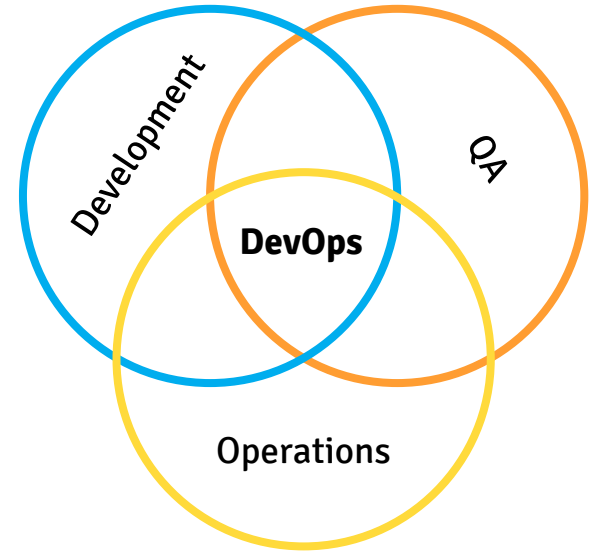
**Get the Flow!**

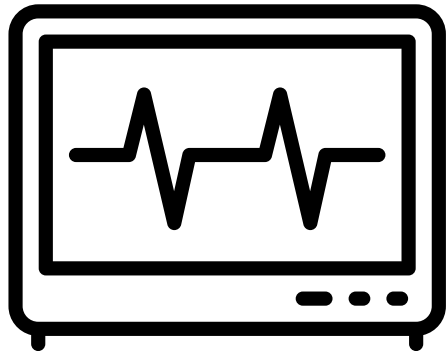
**Dominik Süß, Adobe, Conrad Wöltge, Netcentric**



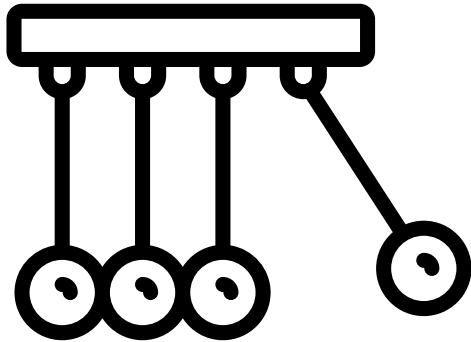
- DevOps & Co.
- The Flow
- ***Demo***
- Challenges

**DevOps** is a [...] practice that emphasizes the **collaboration and communication** of both software developers and [...] (IT) professionals while **automating the process** of software delivery and infrastructure change. (<https://en.wikipedia.org/wiki/DevOps>)





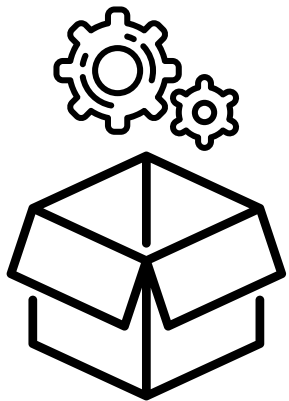
**Continuous Integration (CI)** is a practice to **continuously merge the working state** of all developers together and **test the sanity of the integrated system.**



*Releases are expensive*  
*Release more often!*

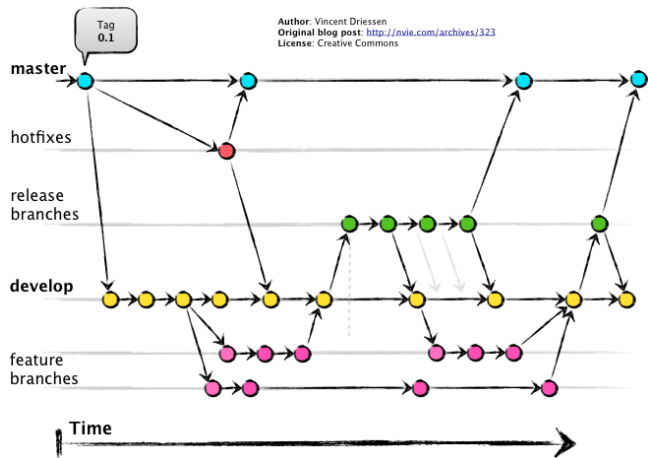
**Continuous Delivery(CD)** is the practice to **continuously** and automatically **validate** **each integrated state** for all **functional and non-functional aspects**



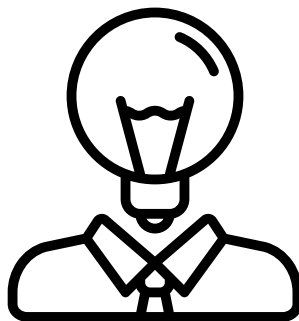


**Continuous Deployment** defines the practice to automate the flow of software **from development to production** through a continuous pipeline.

# The Flow

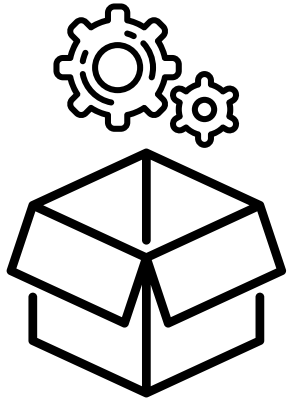


**git Flow**  
(Pull-Request based)



**Jenkins Pipeline**





## Ingredients

- Slingshot application to build
- Jenkins 2.0 with Pipeline Plugin
- Docker Slingshot and Maven
- Github

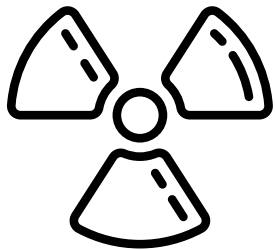
## Szenario

- Pull request: test and deploy to  
PROD 😊

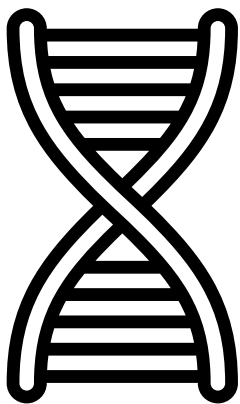
# DEMO

# Challenges: Testing Content

- When to to update slave state for various stages?



Installation state of production might change after hitting early pipeline stages

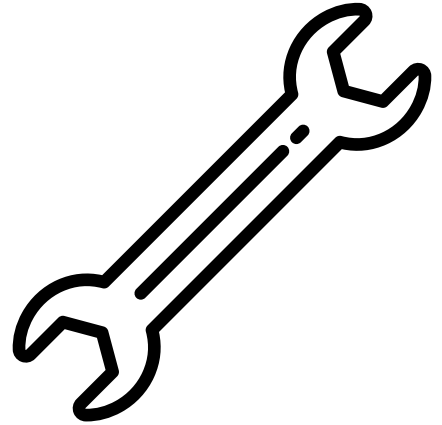


## Tests to be automated in a pipeline:

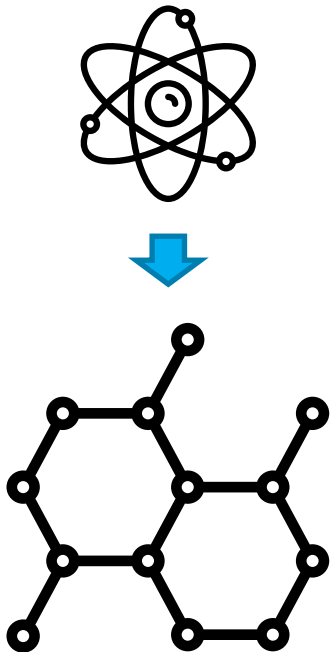
- **Test coverage** (Unit tests, integration tests,...)
- **Code analysis tests** (style, security, ... )
- **Non-functional tests** (endurance, load, stress, security, ...)

# Challenges: Configuration Management

- **Split** Configuration from Code
- Configuration needs to be **versioned**
- Keep configuration and **infrastructure in synch**
- Plan **access control** across stages for Continuous Deployment pipelines.



# Challenges: Sling Specifics



- **Continuous Delivery**  
Coming from file based technologies on monolith repositories
- **Challenges for Sling based Projects**
  - Maven and Java versioning
  - OSGi load specifics
  - Everything is content, data is key
  - Content and configuration states
  - Modularity of application

# Continue on Unattended Installations