

DevOps 2TIN Chapter 1

Introductie DevOps



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



Introductie Devops

Flow van een applicatie
In kaart brengen flow
Hands on: Dev(FI)ops
Reflectie

Levenscyclus van een applicatie



- Bouwen van applicaties is maar één onderdeel van de levenscyclus van een applicatie
- Wat als wij als developer klaar zijn met ontwikkeling?
- Wat als de applicatie draait in productie?
- Wat zit daar nog tussen? Wat komt daarvoor? Wat komt daarna?

Levenscyclus van een applicatie



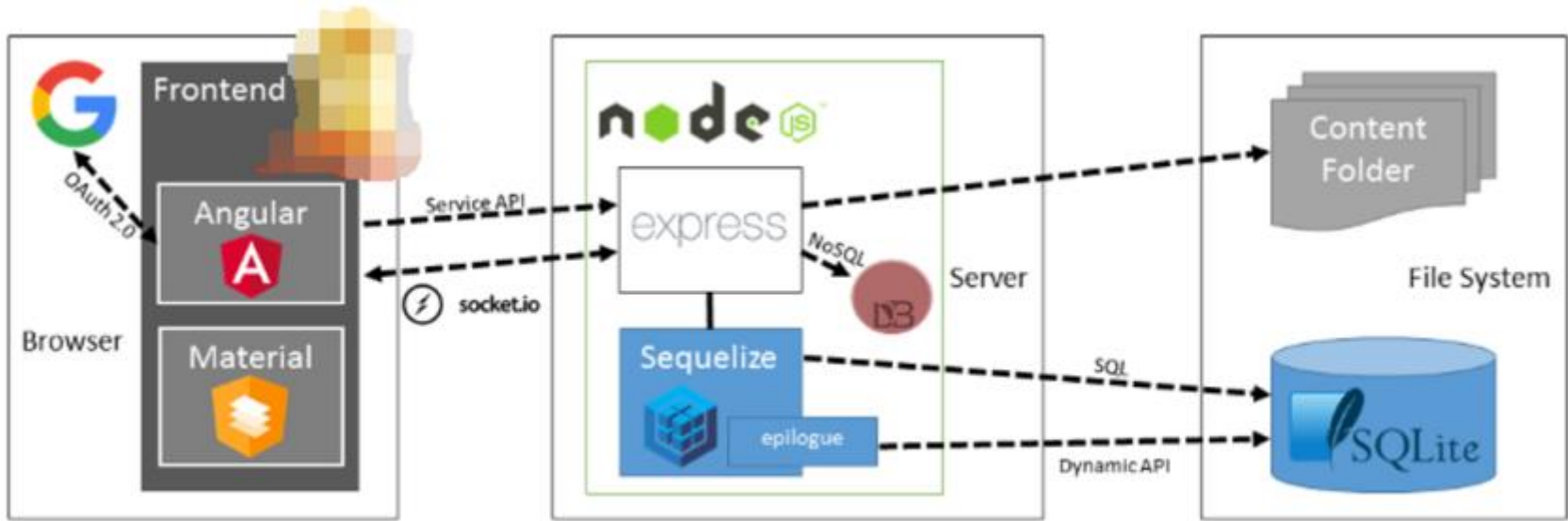
Flow van een applicatie



- Flow van een applicatie = Vanaf het plannen van het bouwen tot aan het punt waarop de applicatie live staat
 - Welke stappen zitten hier allemaal tussen?
 - Wat moet er tijdens deze stappen allemaal gebeuren?

Flow van een applicatie

- In kaart brengen van deze flow



Flow van een applicatie

A screenshot of a web browser window. The tab is titled "2TIN Devops - Calculator". The address bar shows "localhost/2TIN-DevOps-Ca...". The page title is "2TIN DevOps - Calculator". Below the title, there are two input fields labeled "Number 1" and "Number 2". Below these fields is a blue button labeled "Add".

2TIN Devops - Calculator

← → ↻ ⓘ localhost/2TIN-DevOps-Ca... ☆ ABP ⚙️ D ⋮

2TIN DevOps - Calculator

Number 1

Number 2

Add

Flow van een applicatie



- Business requirements
- Maken van userstories
- Ontwikkeling van wireframes/prototypes
- Development - ontwikkelen van nieuwe feature
- Development - unit testen nieuwe code schrijven
- Development - installeren dependencies
- Ops - runnen in testomgeving

Flow van een applicatie

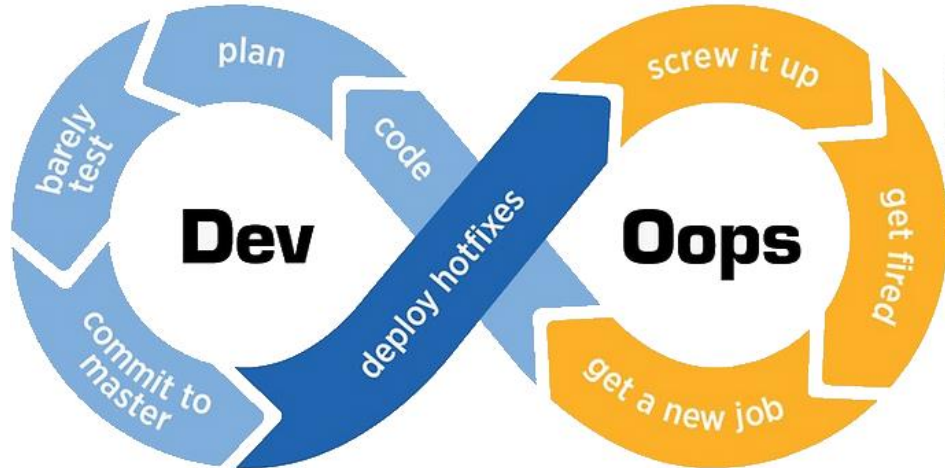


- QA - Functionele testen & Integratietesten, end to end testen,...
- Ops - Deployment naar (productie)omgeving
- Feedback - bugs, issues, logs, nieuwe features?
 - Feedback meenemen naar nieuwe cycle

Dev(fl)Ops



- Lets get our App into production!
- Linux VM with
 - NodeJS
 - Make it work when you go to <http://localhost>



- We did it!
- Wat als we dit proces elke dag opnieuw moeten doen?
 - Meerdere keren per dag
 - Met meerdere developers, QA engineers & operation engineers
 - Zonder fouten

- Geen communicatie tussen Dev (docent) & Ops (student)
 - Hoe weet Ops wat Dev nodig heeft en omgekeerd?
 - Hoe weet Dev hoe de productieomgeving eruit ziet?
 - Hoe weet Dev, Ops & QA wanneer er een nieuwe versie van onze app is?

- Geen automatisatie van onze applicatie flow (op zowel AON, SWM als SNB vlak)
 - Unittesten kunnen genegeerd worden
 - Unittesten moeten handmatig uitgevoerd worden
 - e2e, integratietesten, ... moeten handmatig gebeuren

- Geen single point of truth (centraal beheer van code/configuratie/...)
 - Code op USB stick, verschillende versies
 - Foute code wordt gewoon naar productie gedeployed
 - Geen duidelijk versiebeheer
 - Geen overzicht van configuraties (zowel software als infra)

Reflectie

Wall of
Confusion



LOCAL OBJECTIVES :
Deliver new
functionalities
(ideally of quality)

**Product culture
(software
development)**

WANTS TO DELIVER



LOCAL OBJECTIVES :
Guarantee the “run”
of applications
(stability)

**Service culture
(Archiving,
supervision, support)**

WANTS TO RATIONALIZE



Functional requirements

- Specification of features
- Priorization of features



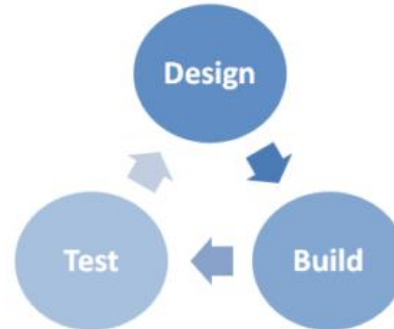
Functional tests

- automated functional tests
- Integrated in build process



Business U

- Issues and bugs
- Evolutions



Non-functional requirements

- Useful troubleshooting logs / Key metrics
- Deployment / supervision concerns
- Degraded modes / Undo-Redo operations



Non-Functional tests

- Automated technical tests
- Integrated in build process
- Continuous Deployment



Technical Operation

- Performances and technical problems



Reflectie



DevOps