

DevOps 2TIN

Chapter 2

Agile, DevOps & the 3 ways



**DE HOGESCHOOL
MET HET NETWERK**

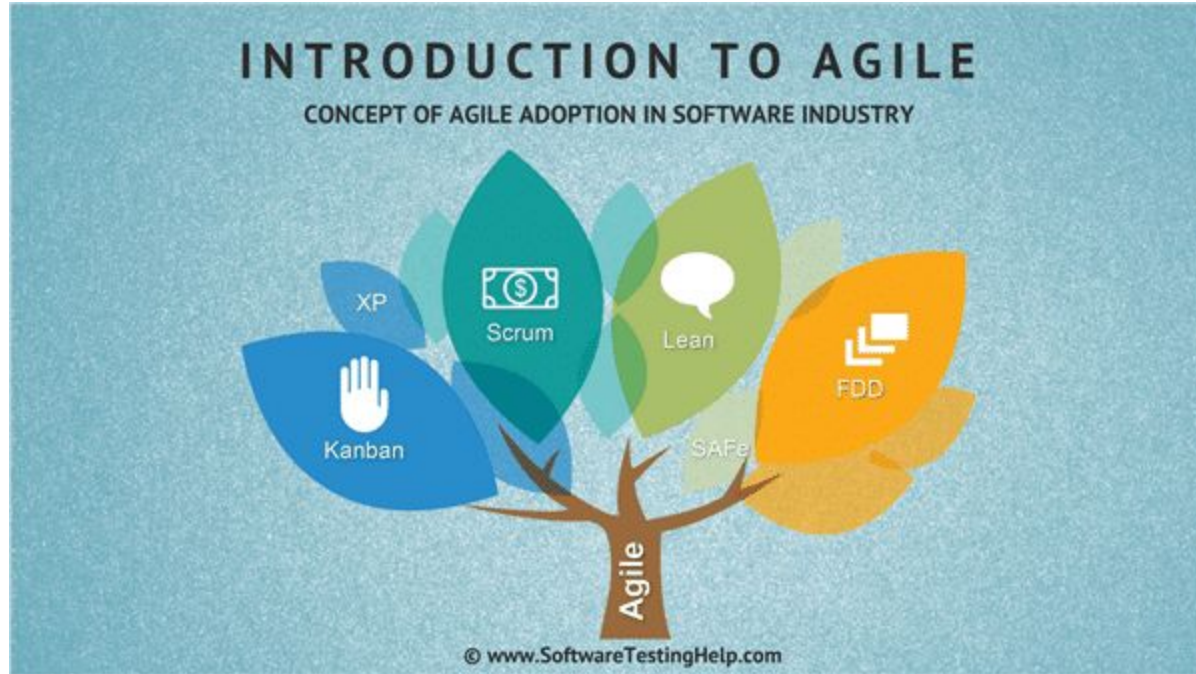
Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



Agile, DevOps & the 3 ways

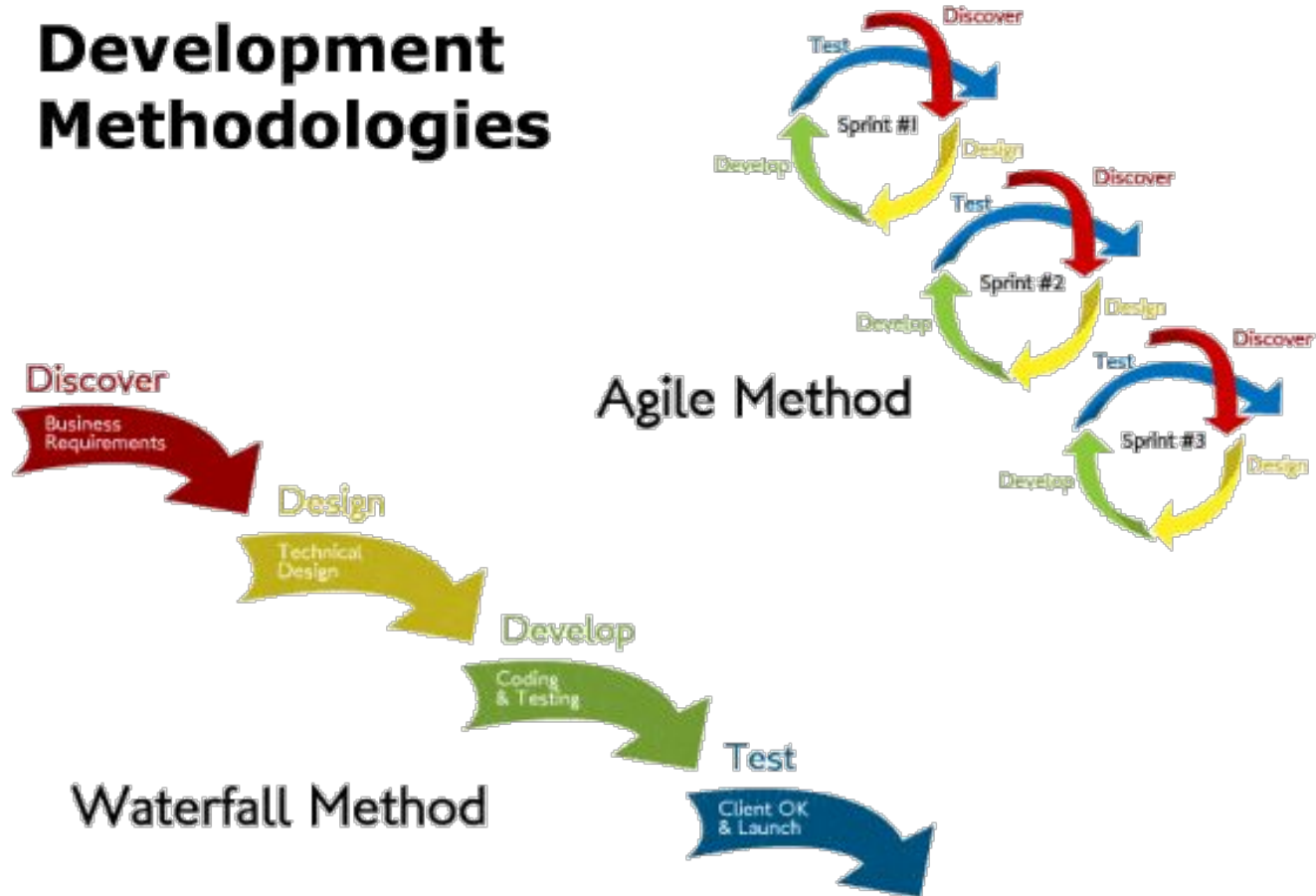
Agile vs DevOps
Historiek
The 3 ways
Reflectie

Wat is agile?

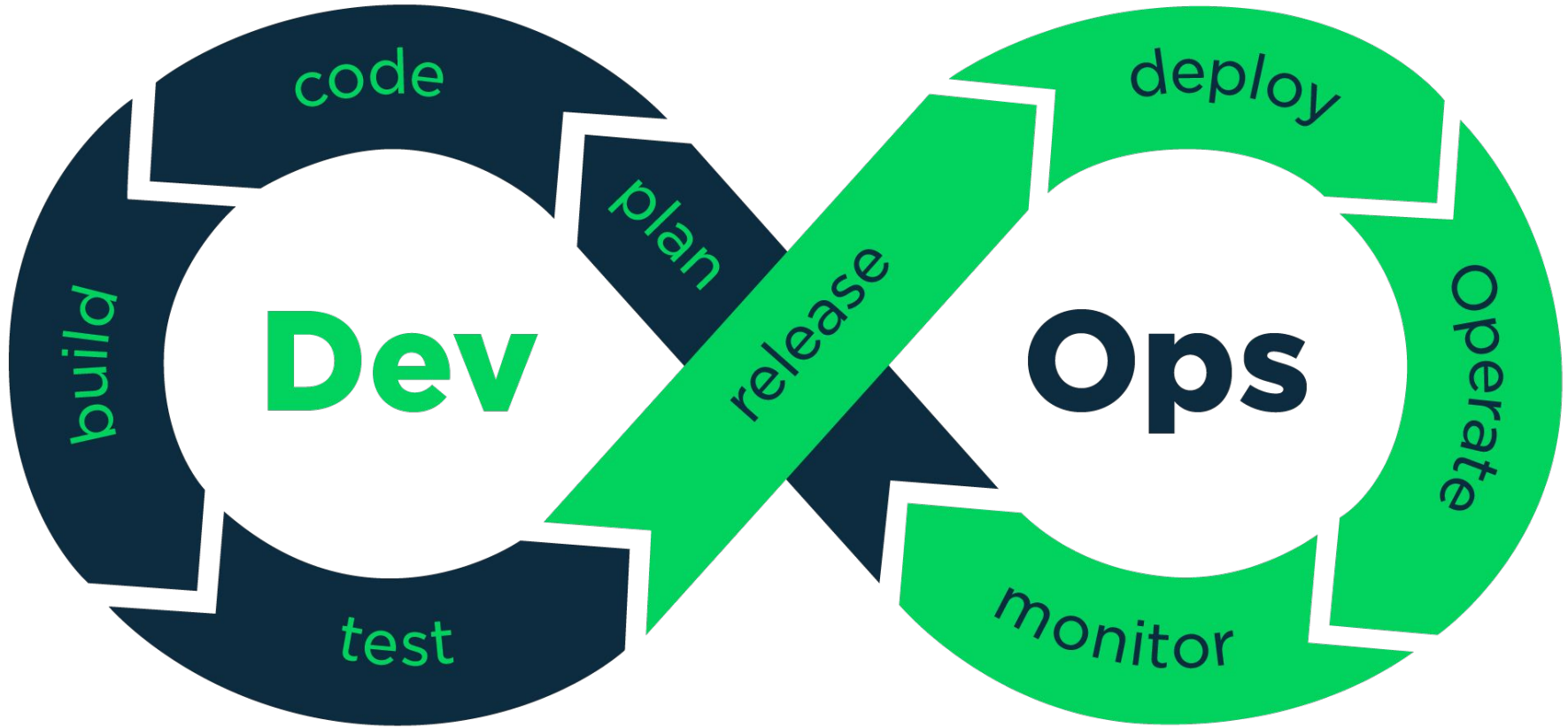


Kleine geschiedenis les

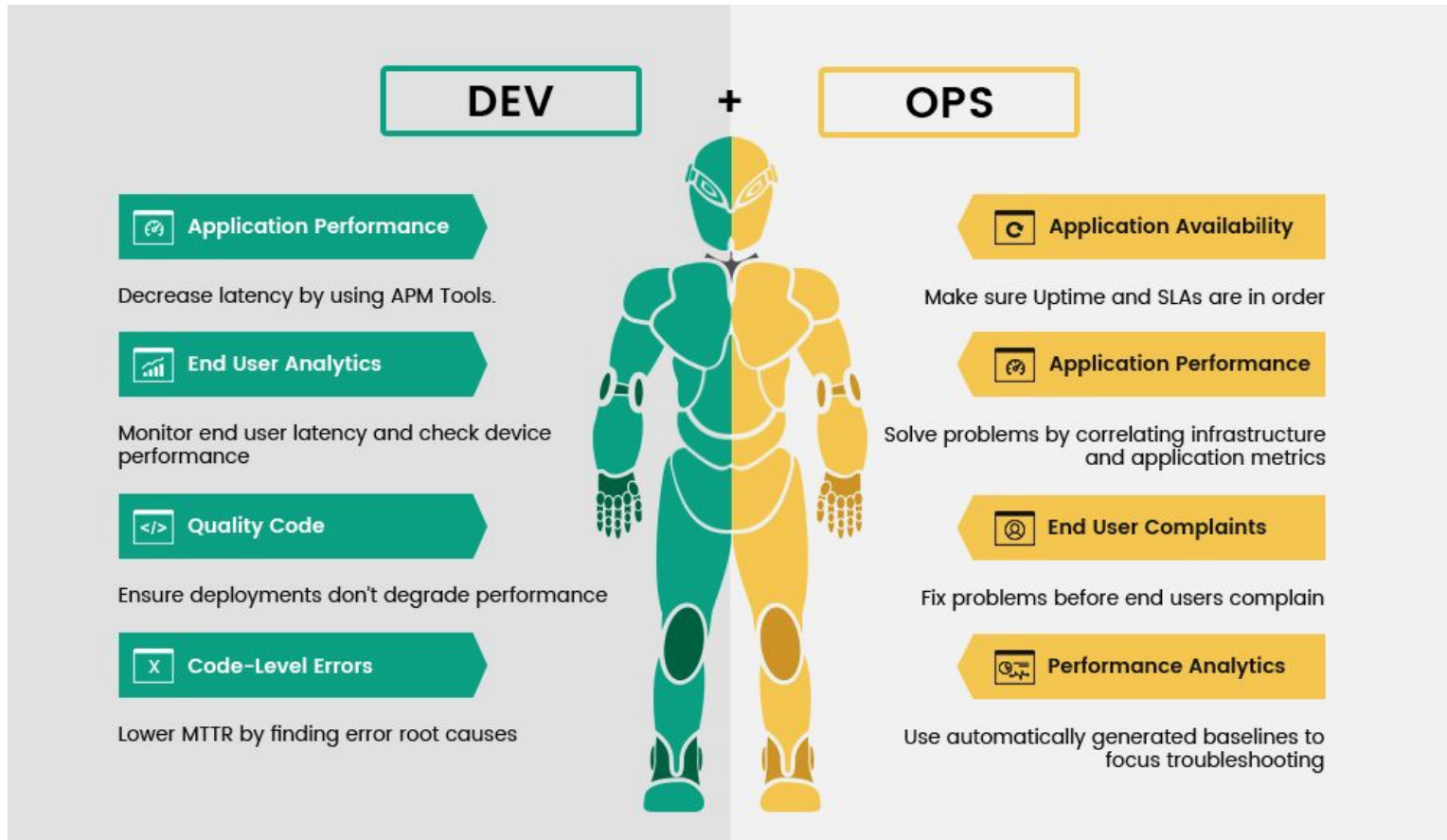
Development Methodologies



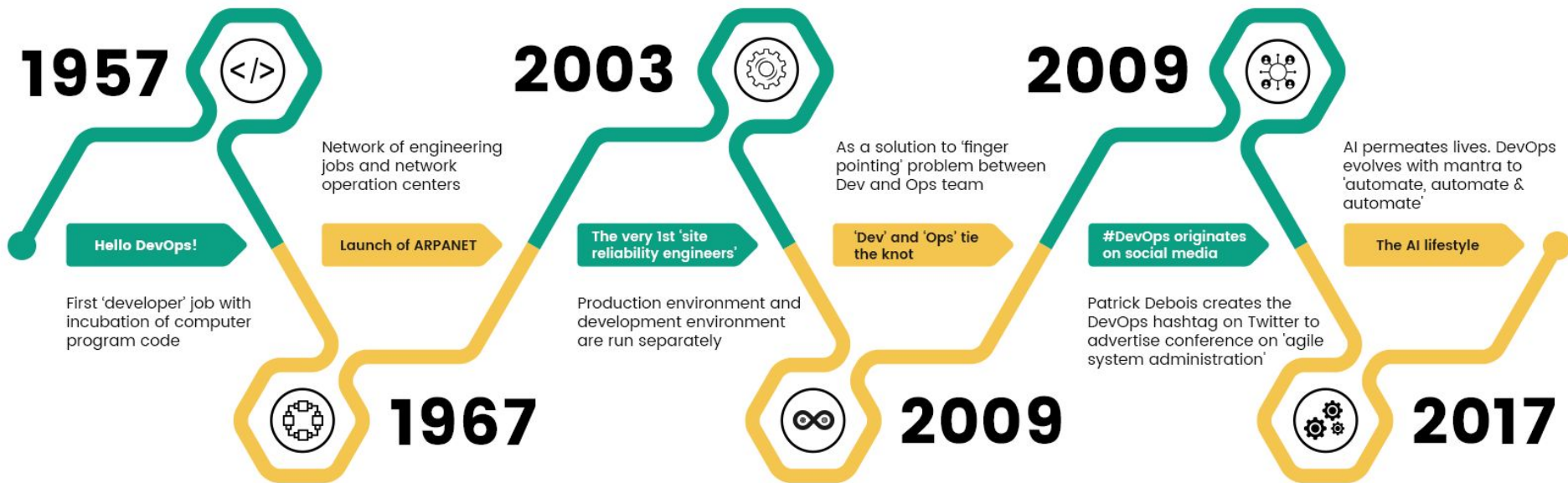
En dan nu DevOps?



Dus DevOps is een praktische aanpak?



Ohja....DevOps is niet nieuw



Veel dingen doen, maar waarom?

De Filosofie achter DevOps:

The 3 Ways

(of, waarom we al die dingen doen)

The first way

The principles of Flow

- Waterfall (van boven naar beneden)
- Agile (van boven naar beneden, in loopjes)
- DevOps (van links naar rechts naar links, altijd in beweging)

Maar, Hoe?

Hoe verhogen we de “flow”

Kleinere stapjes, maar meer releases betekent:

- Minder WiP
- Makkelijker om fouten te vinden en op te lossen
- Maakt werk zichtbaarder



The first way

Identificeer de onderdelen die zorgen dat het business-idee omgezet wordt in een realiteit

- Code, Infrastructuur, Data, Testen, Nieuwe features = Flow

(Business)

(Customer)

Dev



Ops

En maak er een pipeline van ———> Eén systeem ivp verschillende silos

Wat is een Pipeline?

Definitie “deployment pipeline”:

(first defined by Jez Humble and David Farley in their book *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*)

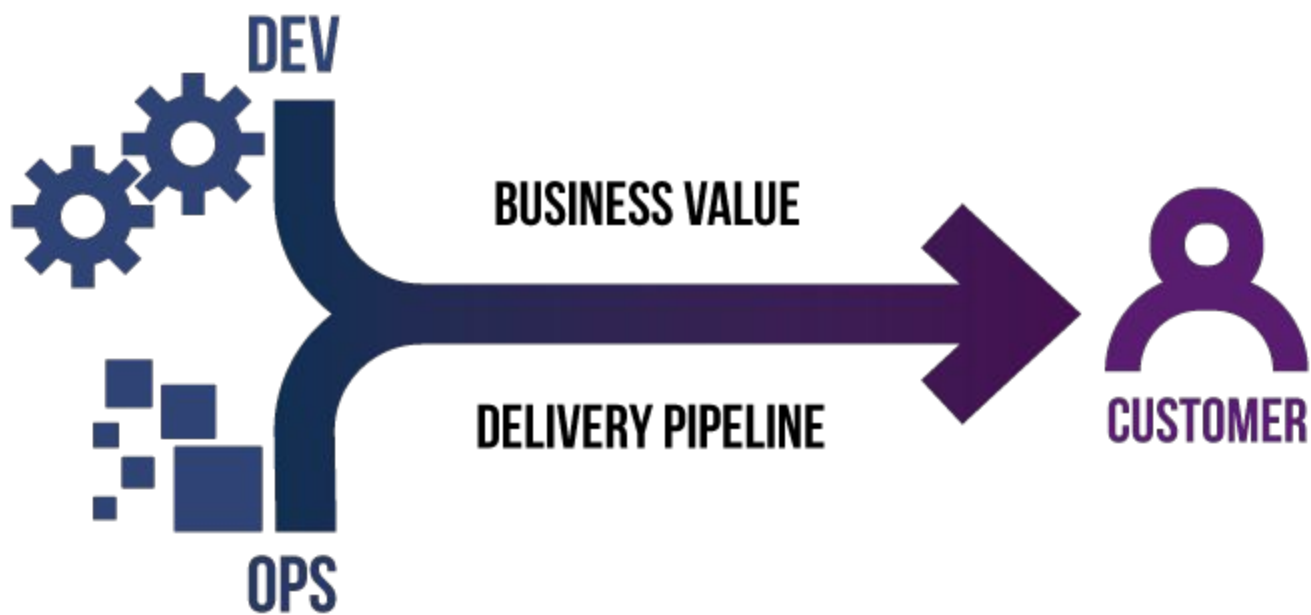
It ensures that all code checked in to version control is automatically built and tested in a production-like environment.

Sleutelwoorden:

- Alle Code
- Versiebeheer
- Automatisch gebouwd
- Automatisch getest
- Productie-waardige omgeving

THE FIRST WAY

CREATE A DELIVERY SYSTEM



The Second Way

The principles of feedback

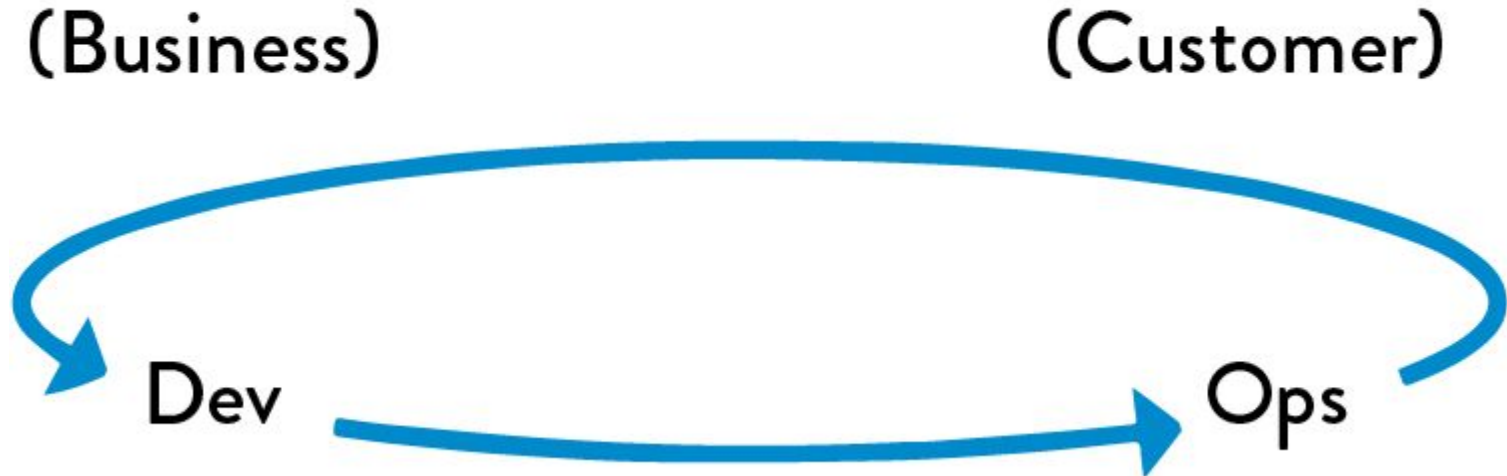
- Waterfall (enkel op het einde)
- Agile (enkel op het einde van de sprint)
- DevOps (continue)

Maar, Hoe?



The Second Way

Door continue te monitoren & testen en ervoor te zorgen dat we problemen opmerken als ze gebeuren, ze aan te pakken en op te lossen alvorens verder te gaan



Test **ALLES** vanaf het moment dat het kan!

Test **automatisch** om Continuous Integration te bekomen

Definitie *continuous integration*:

- **A comprehensive and reliable set of automated tests that validate we are in a deployable state.**
- **A culture that “stops the entire production line” when our validation tests fail. (=digital andon cord)**

Ideal vs. Non-Ideal Testing Pyramids

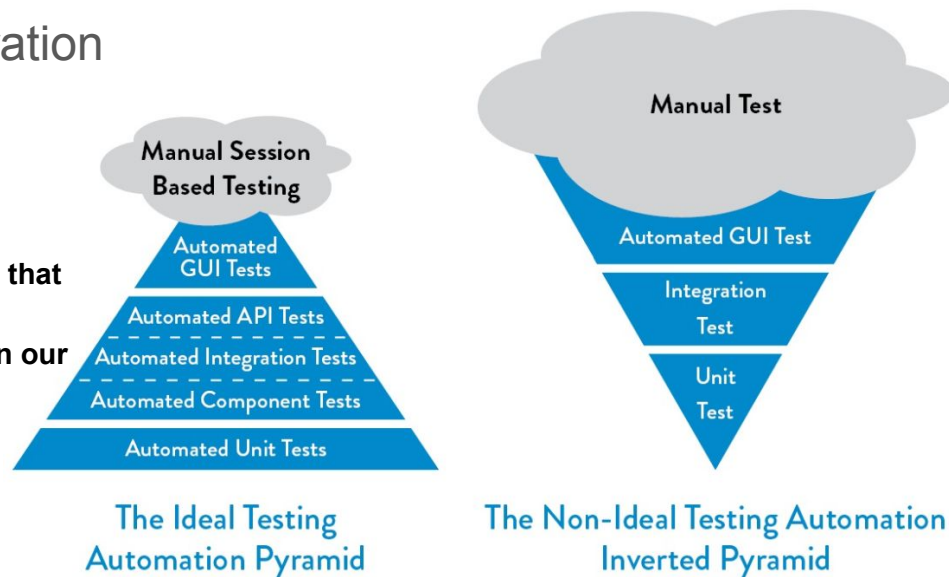


Figure 14: The ideal and non-ideal automated testing pyramids (Source: Martin Fowler, “TestPyramid.”)

Stop de pipeline, gooi het probleem in de groep en los het op
Swarming zorgt ervoor dat dit snel en efficiënt kan



Source: <https://medium.com/devops-cloud-it-career/the-three-ways-of-devops-part-2-of-3-8d473f50d99d>

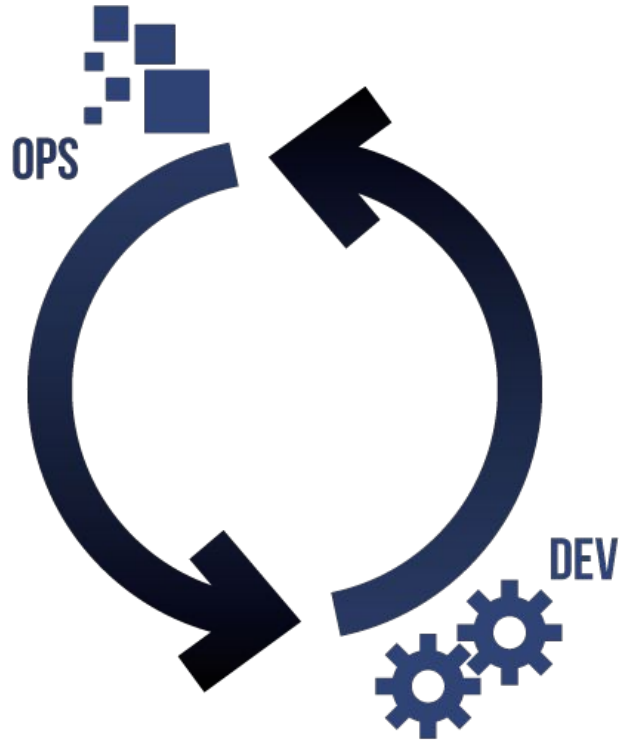
Problemen NU oplossen voorkomt issues later!

Problemen laten liggen “omdat er nu geen tijd is” zorgt voor **Technical Debt**



THE SECOND WAY

AMPLIFY FEEDBACK LOOPS



The Third Way

the creation of a generative, high-trust culture that supports a dynamic, disciplined, and scientific approach to experimentation and risk-taking, facilitating the creation of organizational learning, both from our successes and failures

Enable safety

Improvement of daily work

Local discoveries to global improvements

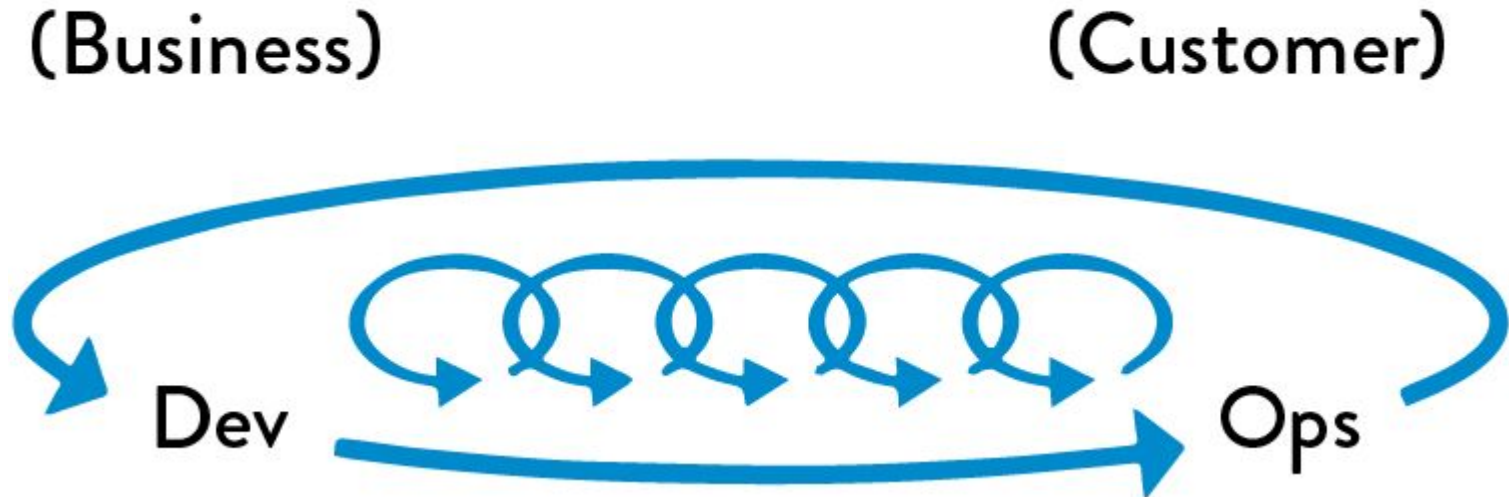
Inject Resilience



"If you are not failing, you are not innovating enough" (Elon Musk)

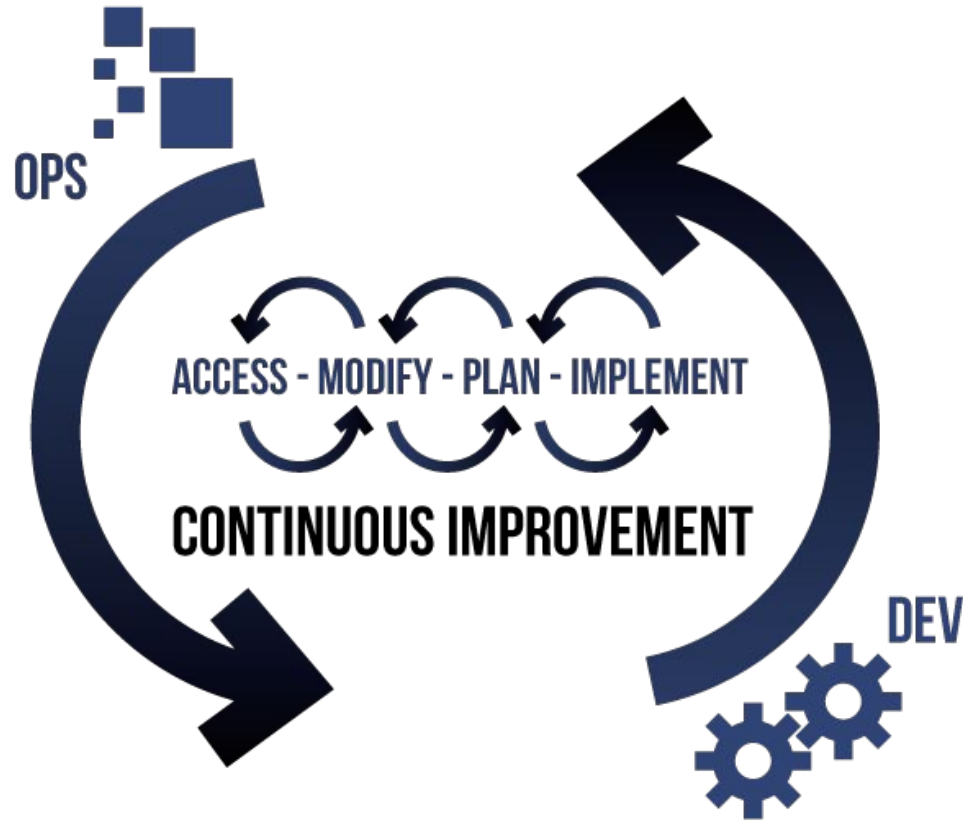
The Third Way

Dit is de moeilijkste, en heeft vaak ook te maken met een cultuur shift binnen bedrijven.



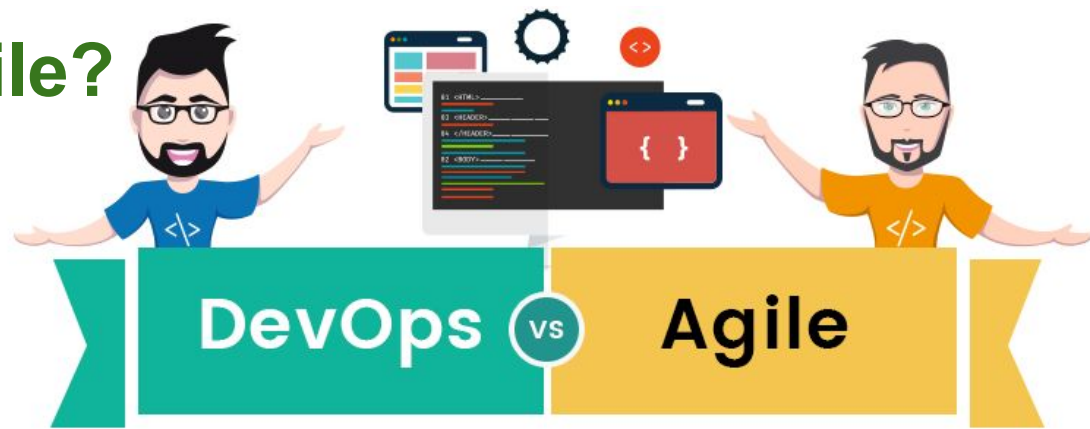
THE THIRD WAY

LEARN & EXPERIMENT CONTINUOUSLY

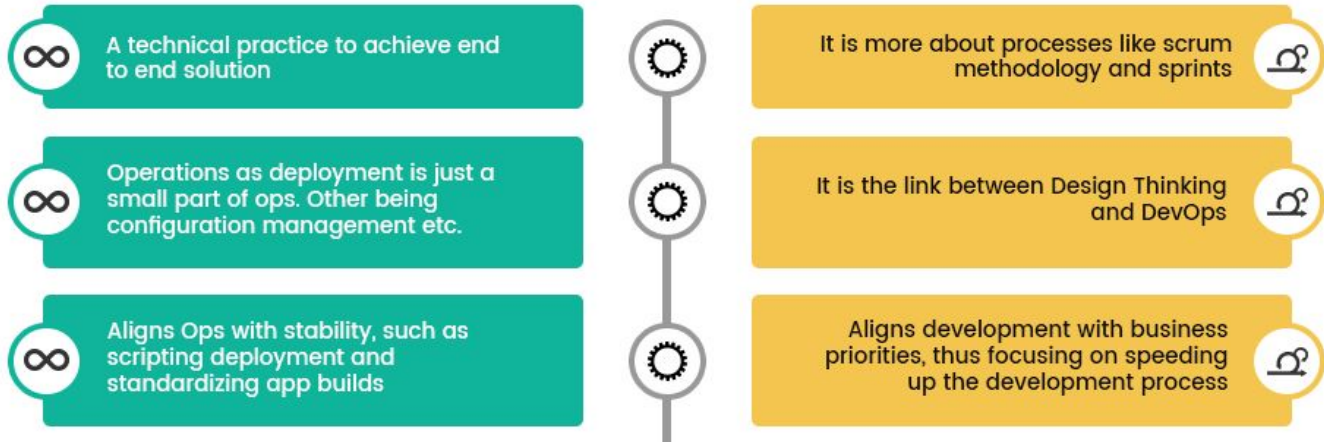


High level overview, hoe we dit in de praktijk gaan doen zien we in de volgende hoofdstukken vanuit de praktijk.

Is DevOps Agile?



What's the difference?



Recap - Is DevOps Agile?

