

bartosz



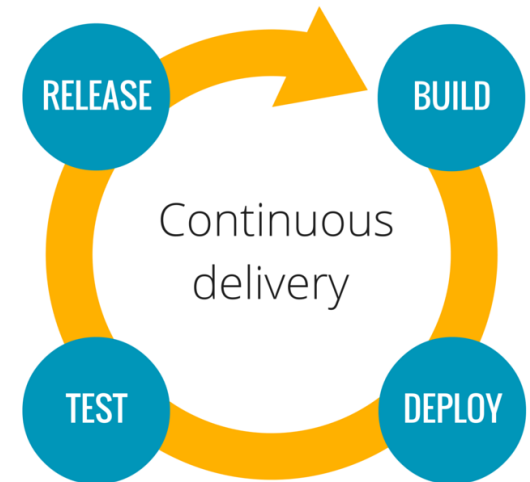
Testautomatisering werkt niet bij Continuous Delivery en DevOps

Testnet Najaarsevent 2016

Joost Voskuil

Setting the stage

- Continuous Delivery & DevOps
 - Gaat niet over hoe vaak je software naar productie brengt
 - Gaat wel over dat je altijd software naar productie **kan** brengen



- Altijd naar productie kunnen?
 - User story done = potential shippable
 - Software ontwerp, bouw, deploy, test en release vinden continu plaats



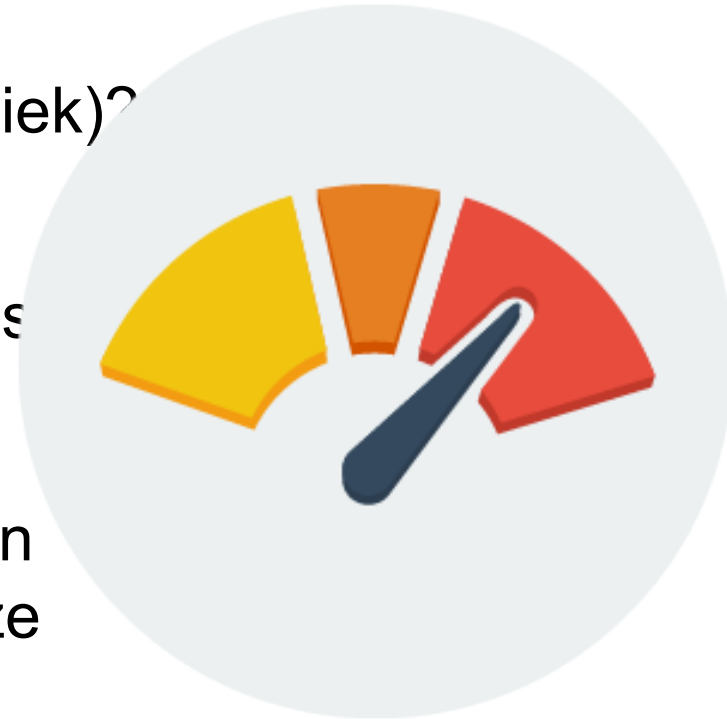
Testen

- Testen vind volledig plaats tijdens de realisatie van de user story;
 - Testen
 - Testautomatisering
 - Defect Fixing
 - Acceptatie testen
 - Non-functionals testen (security, performance)
- Geen V (of W) -model meer, geen aparte fases (ook geen regressietest fase meer)

Feedback loops

- Bouwen we het juiste product (business)?
 - Shift left (BDD / SbE)
 - Exploratory Testing
- Bouwen we het product juist (techniek)?
 - TDD
 - Performance & Security testing
- Zorgen wijzigingen niet voor regressie
 - Continuous Integration & Delivery

Hoe sneller we hier antwoord op kunnen krijgen op deze vragen, hoe sneller onze throughput!





‘Andere’ kijk op testen

Als we die snelheid willen moeten we:

- Als team nadenken over kwaliteit en allemaal testen
- Hoge mate van test-automatisering
- Testen voor nieuwe inzichten
 - Experimenteren
 - Leren

Testen:

- Niet meer controlerend maar ondersteunend:
- Van Quality Assurance naar Quality Assistance!

Checken vs Testen

Bach/Bolton:

*“**Testing** is the process of evaluating a product by learning about it through exploration and experimentation, which includes to some degree: questioning, study, modeling, observation, inference, etc.”*

*“**Checking** is the process of making evaluations by applying algorithmic decision rules to specific observations of a product.”*

Traditioneel testautomatisering

- Onderscheid tussen 'black-box' en 'white-box' testing
 - Unit testing (als je geluk hebt want gebeurd vaak niet)
 - GUI testing
 - Selenium
 - UFT
 - Ranorex
- Verantwoordelijkheid van testers & testautomatiseerders;
- Bennefits alleen voor testers & testautomatiseerders;
- Wordt gerealiseerd na de implementatie van de software

Problemen testautomatisering

- Testautomatisering is hoofdzakelijk black-box testing;
 - Trage test-uitvoer is een trage feedback loop;
 - Complete infrastructuur is nodig incl. testdata;
 - Onderhoudsgevoelig en vallen snel om (brittle / flacky)
 - Onvoldoende waarde voor ontwikkelaars;
 - Er is een testautomatiseerder en daardoor hand-overs
 - De testen raken alle applicatie lagen maar onvoldoende de paden in een applicatie laag

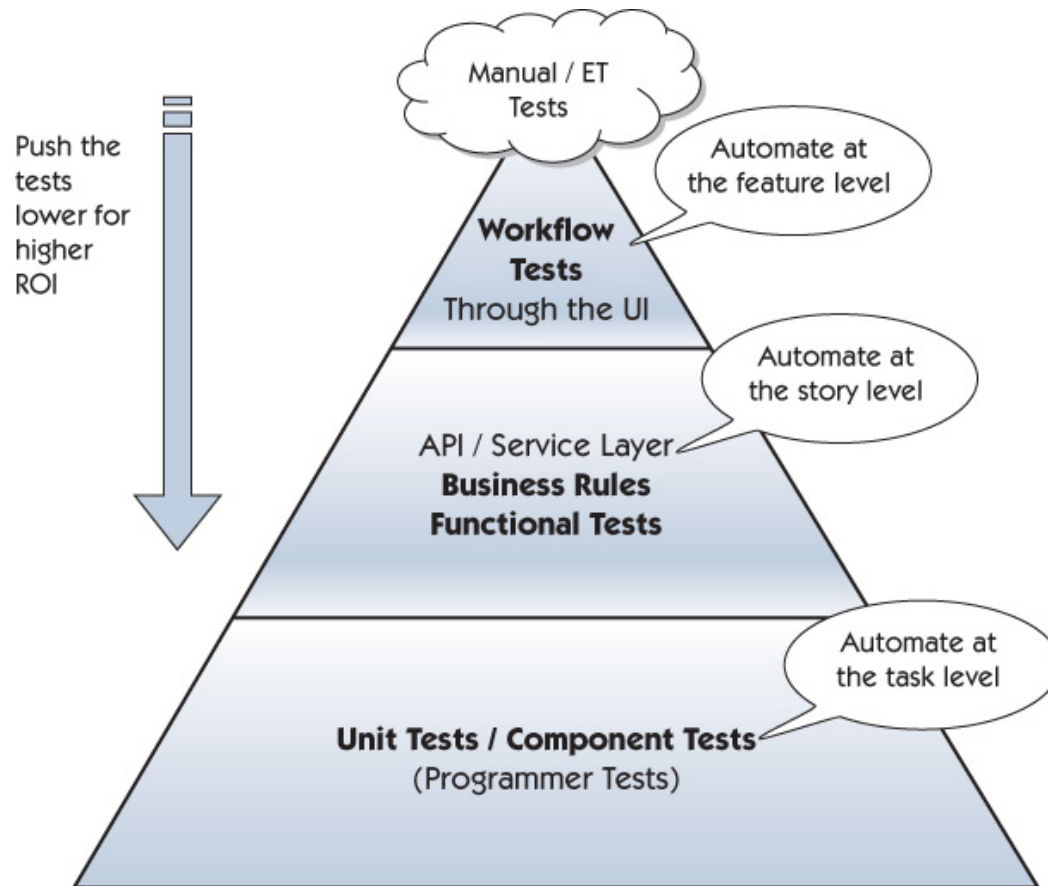


CD / DevOps Testautomatisering

- Geen onderscheid meer tussen 'black-box' en 'white-box'
- Verantwoordelijkheid van het hele team
- Bennefits voor het hele team
- Volledig vangnet
 - Snel
 - Betrouwbaar en stabiel
 - Nauwkeurig
 - Makkelijk te onderhouden
- Wordt gerealiseerd tijdens of zelfs voor de implementatie van de software (zoals bij BDD / SbE)



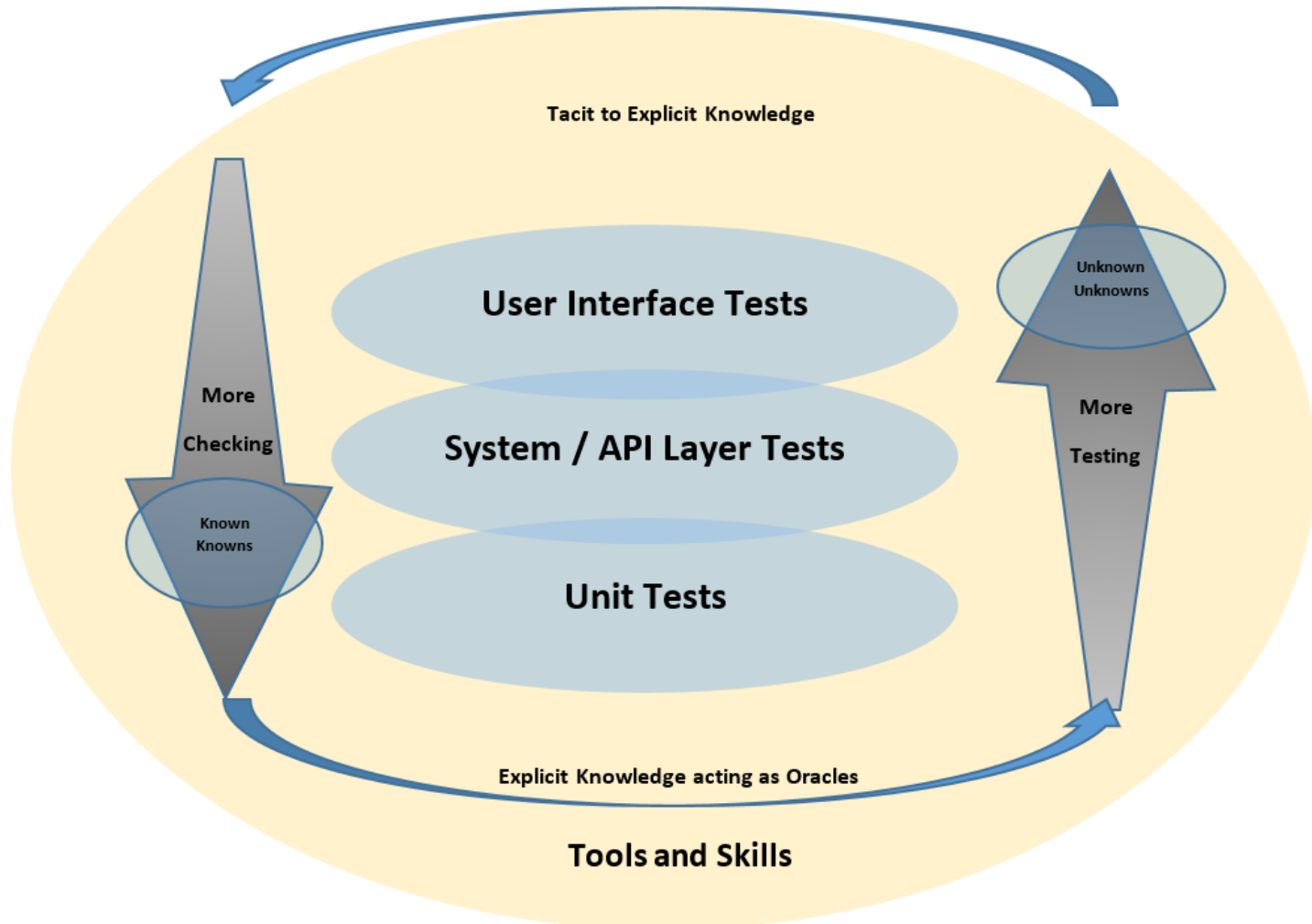
Test automation piramide



bartosz 



@steveo1967





Strategie

- Checks dienen uitgevoerd te worden op de juiste laag van architectuur. Deze leveren een horizontale (automation) dekking
 - Rekenregels
 - Business rules
 - Beslissingen
- Integratie of Component checks geven een samenhangend geheel en testen verticaal door de piramide.

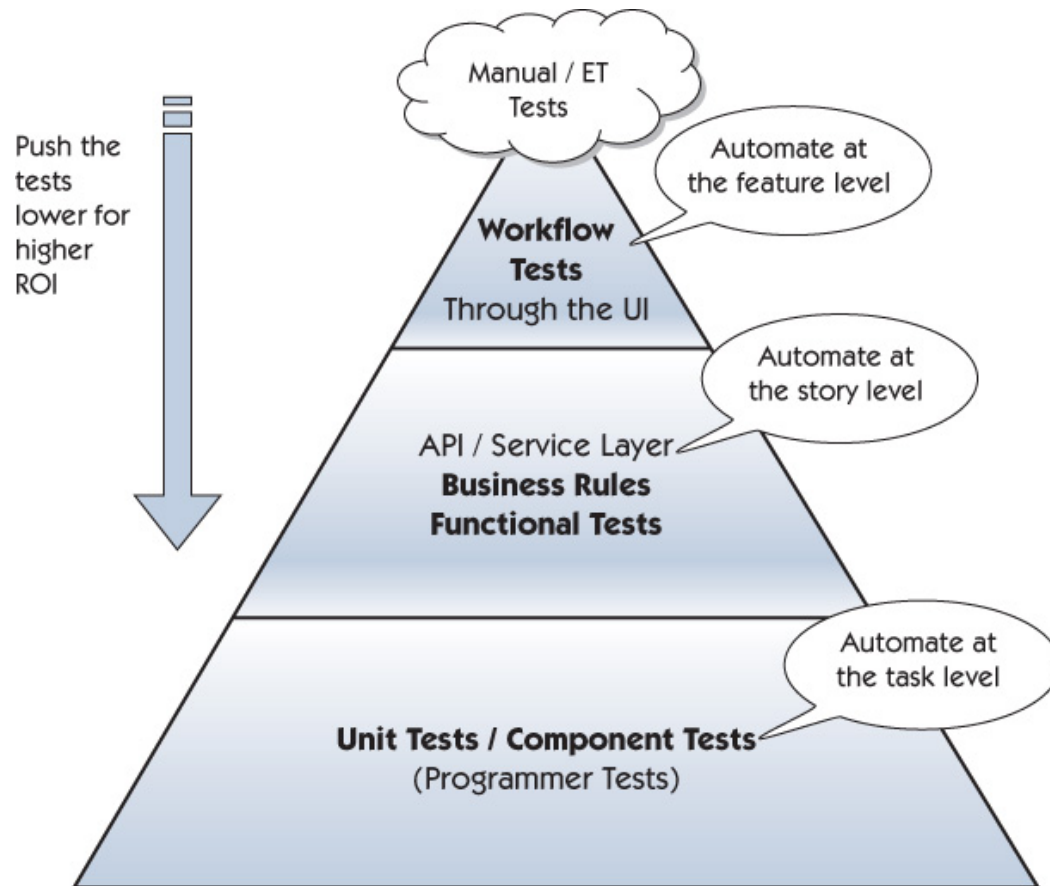


Technischer

- Kwaliteit van testautomatiseringscode hetzelfde als productiecode
- SOLID
 - SRP Singe Responsibility principle
 - OCP Open/Closed principle
 - LSP Liskov substitution principle
 - ISP Interface Segregation principle
 - DIP Dependency Inversion principle
- Mocking frameworks
- Testdata Builders



Test automation piramide





Take aways – stop met:

- Last line of defence zijn
 - Het controleren van je collega heeft onvoldoende toegevoegde waarde.
- Testautomatisering via de UI
 - Leveren trage en onbetrouwbare feedbackloops op.
- Testautomatiseerder zijn
 - Je hebt onvoldoende technische kennis.
 - Bij BDD/SbE/TDD is er outside in development.
- Gherking en Cucumber gebruiken als je geen BDD / SbE doet
 - Waarom zou je, verkeerde tooling

Take aways – start met:

- **Quality Assistance**
 - Infecteer je team met de kwaliteitsmindset
- **Maak een keuze:**
 - Leer echt ontwikkelen (OO, SOLID)
 - Of richt je op de Specificaties
 - Of richt je op Exploratory Testing
- **BDD en SbE**
 - Op gedrag & beslissingen (maar niet via de GUI)
 - Outside in development (dus ontwikkelaar)