

leren. durven. doen.



# SOFTWARE TESTEN

*Grondbeginselen / Test Niveaus  
4 werkuur*

Hoofdtitel: Software Testen

Subtitel: Grondbeginselen / Basis principes

Auteur / docent: Vincent Van De Walle

Versie: Gebruikt tijdens opleiding

© Syntra West

*De docent/auteur en Syntra West behouden zich het recht voor de inhoud, het beeldmateriaal en alle verdere informatie in deze PowerPoint zonder voorafgaand bericht aan te passen of te verbeteren.*

*Deze PowerPoint is met zorg samengesteld, waarbij alles werd in het werk gesteld om de juistheid van de inhoud te verzekeren, niettemin wordt deze PowerPoint ter beschikking gesteld zonder enige garantie.*

*De docent/auteur en Syntra West kunnen niet verantwoordelijk noch aansprakelijk gesteld worden voor eventuele materiële schade of andere vergissingen die voortvloeien uit het gebruik van de informatie uit deze PowerPoint of voor eventuele fouten of vergissingen in de PowerPoint. Dit cursusmateriaal is auteursrechtelijk beschermd.*

**Alles** uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, elektronisch of op welke andere wijze ook, of in een retrieval systeem worden opgeslagen, zonder voorafgaandelijke schriftelijke toestemming van de docent/auteur. Een verwijzing naar de bron en auteur wordt gewaardeerd.

**Niets** uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, elektronisch of op welke andere wijze ook, of in een retrieval systeem worden opgeslagen, zonder voorafgaandelijke schriftelijke toestemming van de Syntra West.

leren. durven. doen.



# VINCENT VAN DE WALLE

*Vincent@CopyPaste.be*

*VincentVanDeWalle@digi4u.be*

+32 (0) 475 / 97.63.80

# De wielerwedstrijd

- **JE DOET MEE AAN EEN WIELERWEDSTRIJD**

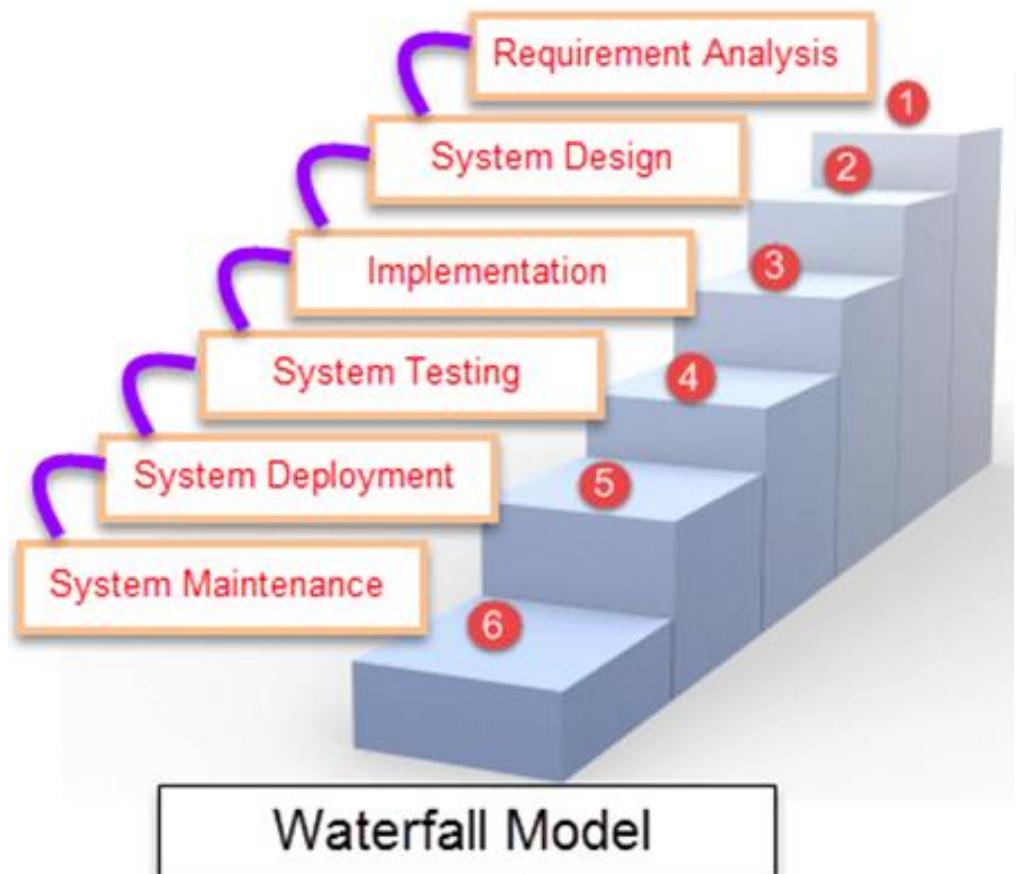
- Je rijdt rond, mee met de bende.
- Naarmate de wedstrijd het einde nadert, wordt het interessant.

# De wielerwedstrijd

- **JE DOET MEE AAN EEN WIELERWEDSTRIJD**
  - Je rijdt rond, mee met de bende
  - Naarmate de wedstrijd het einde nadert, wordt het interessant.
- **IJ, ALS WIELRENNER OF WIELRENSTER, HAALT DE TWEEDE IN**
  - Op de hoeveelste plaats rij je dan?
- **IJ, ALS WIELRENNER OF WIELRENSTER, HAALT DE LAATSTE IN**
  - Op de hoeveelste plaats rij je dan?

# Terminologie en hun betekenis

- **SOFTWARE DEVELOPMENT LIFE CYCLE**
  - SDLC
  - Stappen in het maken van de software
- **EEN AANTAL METHODOLOGIEËN**
  1. **WATERVAL MODEL**
  2. V-MODEL
  3. SPIRAAL MODEL  
Incrementeel
  4. ITERATIEF MODEL  
Iteratief
  5. AGILE / SCRUM  
Incrementeel iteratief
  6. DEVOPS  
Het realiseren van incrementeel iteratief



# Terminologie en hun betekenis

- **SOFTWARE DEVELOPMENT LIFE CYCLE**
  - SDLC
  - Stappen in het maken van de software
- **EEN AANTAL METHODOLOGIEËN**

## 1. WATERVAL MODEL

## 2. V-MODEL

## 3. SPIRAAL MODEL

Incrementeel

## 4. ITERATIEF MODEL

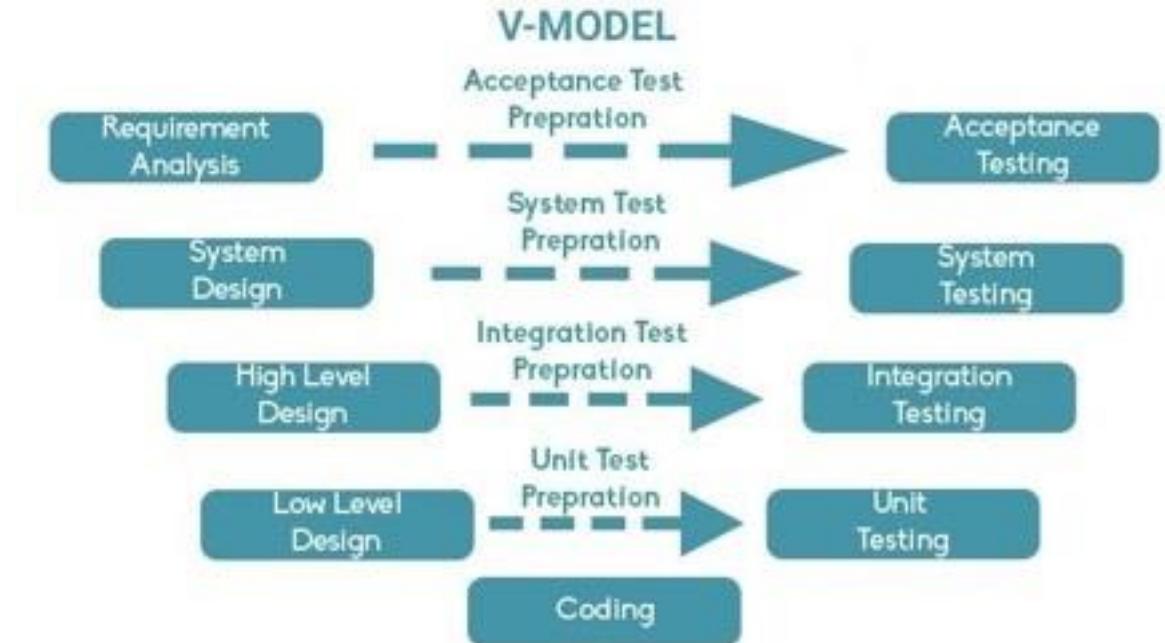
Iteratief

## 5. AGILE / SCRUM

Incrementeel iteratief

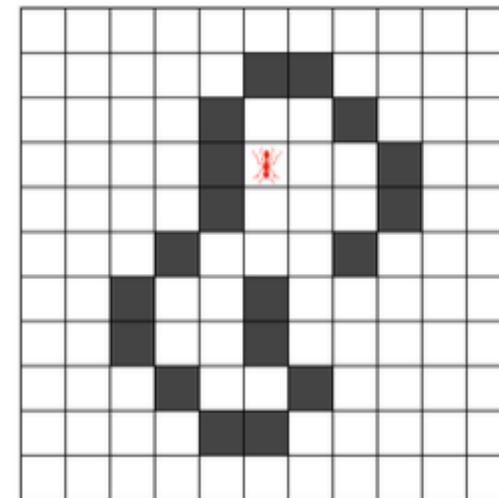
## 6. DEVOPS

Het realiseren van incrementeel iteratief



# De requirements in de verschillende methodes

- DE MIER VAN LANGTON
  - SOFTWARE DEVELOPMENT LIFE CYCLE
    - SDLC
    - Stappen in het maken van de software
  - EEN AANTAL METHODOLOGIEËN
1. WATERVAL MODEL
  2. V-MODEL
  3. SPIRAAL MODEL  
Incrementeel
  4. ITERATIEF MODEL  
Iteratief
  5. AGILE / SCRUM  
Incrementeel iteratief
  6. DEVOPS  
Het realiseren van incrementeel iteratief



# Wat is een computer programma?

- EEN SET VAN INSTRUCTIES VOOR EEN COMPUTER.



© Simon Allardice  
*What is programming?*  
Pluralsight (betaling)  
Tekst op Moodle

A blurred screenshot of a computer monitor showing a block of Java code. The code appears to be a JPA query builder example, specifically using TypedQuery<Person>. The code includes several if statements for filtering by name and age, and a call to em.createQuery().

```
String hql = "SELECT p FROM Person p";  
9  if(parameters.contains("name")){  
10     hql += " WHERE p.name = :name";  
11 }  
12 if(parameters.contains("age")){  
13     hql += " WHERE p.age = :age";  
14 }  
15 TypedQuery<Person> query = em.createQuery(hql, Person.class);  
16 if(parameters.contains("name")){  
17     query.setParameter("name", values[0].toString());  
18 } else if(parameters.contains("age")){  
19     query.setParameter("age", Integer.valueOf(values[1].toString()));  
20 }  
21 return query.list();
```

# Wat is een computer test?

- EEN SET VAN INSTRUCTIES VOOR EEN PERSOON OF EEN SYSTEEM.

```
String hql = "from Person p where ";
01 02 03 04 05 06 07
07 08 09 10 11 12 13 14 15 16
if(parameters.contains("name")){
    hql += " and p.name = :name";
}
if(parameters.contains("age")){
    hql += " and p.age = :age";
}
TypedQuery<Person> query = em.createQuery(hql, Person.class);
if(parameters.contains("name")){
    query.setParameter("name", values[0].toString());
}
if(parameters.contains("age")){
    query.setParameter("age", Integer.valueOf(values[1].toString()));
}
List<Person> result = query.list();
```

# Een test in zijn eenvoudigste vorm

- **JE GAAT EEN ACTIE UITVOEREN**
  - Doe dit, doe dat, doe iets
- **JE CONTROLEERT OF HET GEDRAG JUIST IS**
  - Hoe reageerde de software, het systeem, het proces?

# Windows verkenner wordt 2 maal gestart

- **JE GAAT EEN ACTIE UITVOEREN (EEN STAP)**
  - Je zult een bestand van het ene venster naar het ander verplaatsen, door te slepen.
- **JE CONTROLEERT OF HET GEDRAG JUIST IS**
  - Hoe reageerde de software, het systeem, het proces?

# Een test, iets uitgebreider

- **JE GAAT EEN ACTIE UITVOEREN (EEN STAP)**
  - Doe dit, doe dat, doe iets
- **JE CONTROLEERT OF HET GEDRAG JUIST IS**
  - Hoe reageerde de software, het systeem, het proces?
- **ER ZIJN VOORWAARDEN AAN DE ACTIE EN/OF HET RESULTAAT**
  - Wanneer wordt de actie uitgevoerd?
  - Wanneer is het resultaat juist of fout?
- **ER ZIJN VOORWAARDEN AAN DE TEST**
  - Wanneer kan de test worden uitgevoerd?

# De “en”, de “of”, de “niet”

- **DE 16 TEKENINGEN**

- Hoeveel tekeningenetjes zijn er ...?



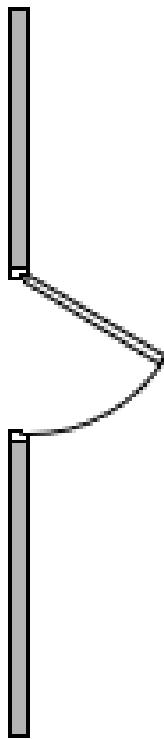
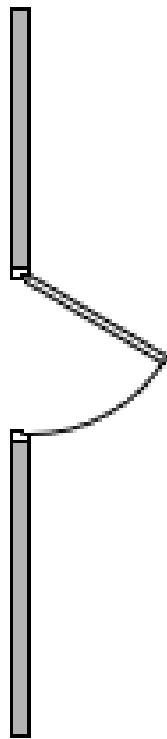
- **VISUALISEER DE “EN”**

- **VISUALISEER DE “OF”**

- **DOE DE OEFENING MET EEN “NIET”**

# De “En”

$$\square \quad \circ + \circ$$



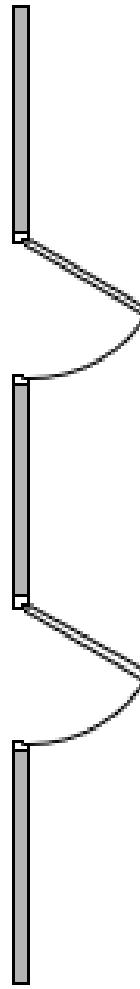
# De “Of”

□ ○ + :

□ ○ + :

□ ○ + :

□ ○ + :

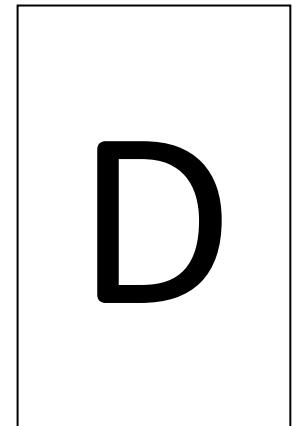
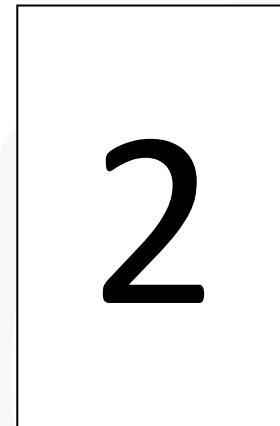
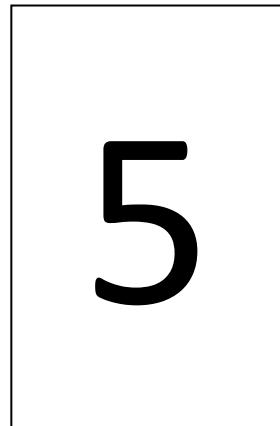
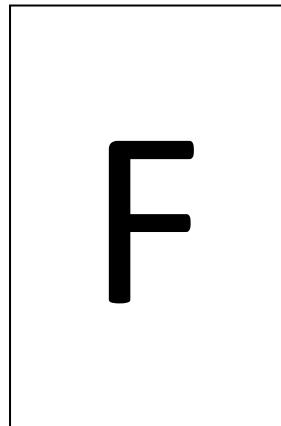


# Voorwaarden: Het bevestigen of niet van iets

- **STEL JE HEBT KAARTJES**

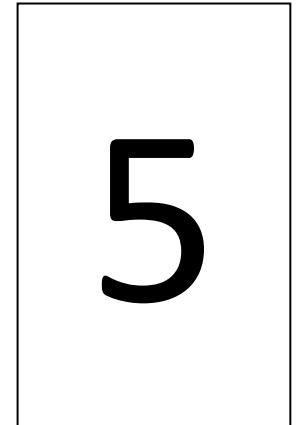
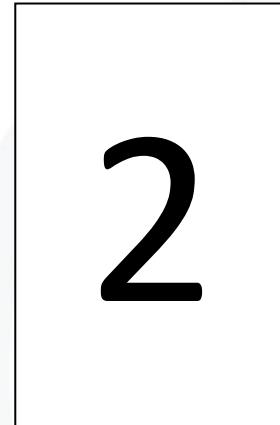
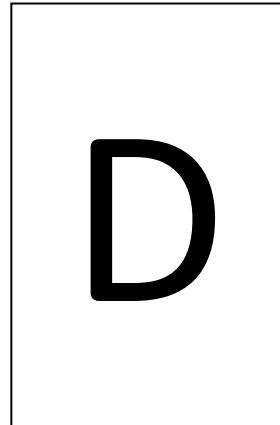
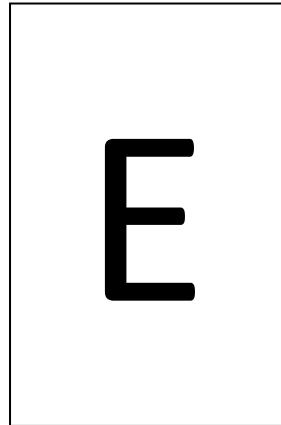
- Op de ene kant van het kaartje staat een letter.
- Op de andere kant van het kaartje staat een getal.
- Er liggen 4 kaartjes voor je.

- **EEN VOORBEELD**



# Voorwaarden: Het bevestigen of niet van iets

- **TEST DEZE REGEL**
  - Als een kaartje een klinker heeft op één kant, dan heeft het een even getal aan de andere kant.
- **WELKE KAARTJES MOETEN OMGEDRAAID WORDEN OM DEZE REGEL TE TESTEN?**



# Voorwaarden: Het bevestigen of niet van iets

- **TEST DEZE REGEL**
  - Als een kaartje een klinker heeft op één kant, dan heeft het een even getal aan de andere kant.
- **WELKE KAARTJES MOETEN OMGEDRAAID WORDEN OM DEZE REGEL TE TESTEN?**

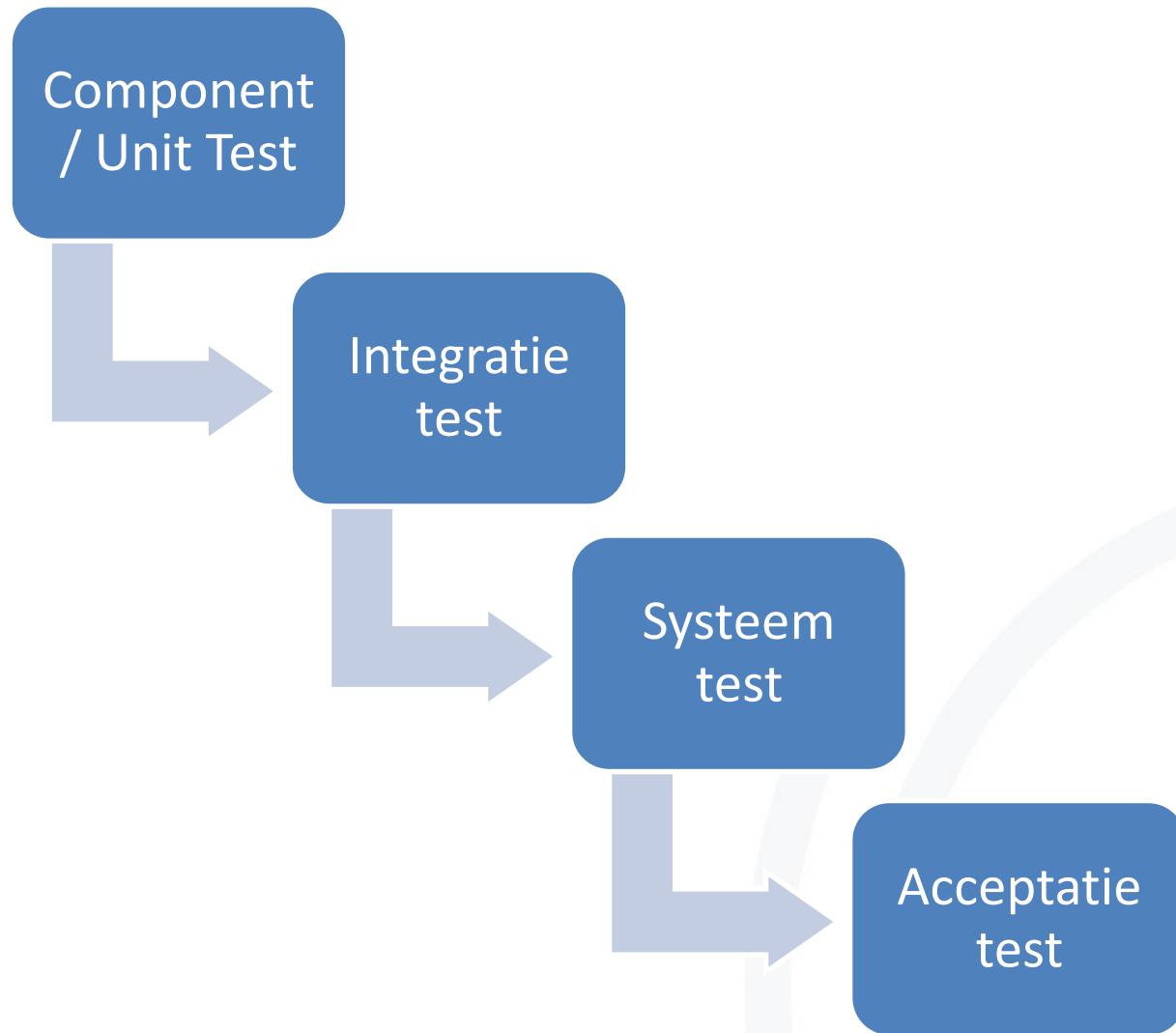
21

Y

0

D

# Test niveaus



# Test niveaus: Component Test / Unit test

- **CHECK WAAR HET STAAT IN HET V-MODEL**
- **WAT IS HET?**
  - Een niveau in het test verhaal, waar individuele componenten of units worden getest, alsof ze het eindproduct waren.
  - We proberen dit component zo klein mogelijk te houden.
  - Het doel is na te gaan of de unit / de component / het onderdeel correct werkt, zoals het ontworpen is.

# Game: Ontdek het patroon

- **JE OPDRACHT IS HET PATROON TE VINDEN WANNEER IETS CORRECT OF FOUT IS**
  - Er is een component of een unit die een “Ja” / “Nee” antwoord terug geeft.
  - De input zijn 3 getallen
    - Ofwel krijg je een “Ja”, de drie getallen voldoen aan de regeltjes binnen in het component / unit
    - Ofwel krijg je een “Nee”, de drie getallen voldoen niet aan de regeltjes binnen in het component / unit
  - Iedere input wordt bijgehouden.
  - Je krijgt 1 mogelijkheid om te zeggen wat het patroon is.
    - Doe dit dus enkel en alleen wanneer je zeker bent dat het juist is.

# Game: Ontdek het patroon

- JE OPDRACHT IS HET PATROON TE VINDEN WANNEER IETS CORRECT OF FOUT IS
- INPUT 1, 2, 4 IS CORRECT.
- [HTTPS://EXPLORABLE.COM/CONFIRMATION-BIAS](https://explorable.com/confirmation-bias)

# Een test, die wordt gemaakt

- **GEEF DE TEST EEN DUIDELIJKE NAAM**
- **ZORG DAT JE DE TEST TERUG KUNT VINDEN, TUSSEN  
DUIZENDEN ANDERE TESTEN**

# Een test, die wordt uitgevoerd

- **JE VOERT EEN ACTIE UIT**
  - Doe dit, doe dat, doe iets
- **JE CONTROLEERT OF HET GEDRAG JUIST IS**
  - Hoe reageerde de software, het systeem, het proces
- **ER IS EEN VOORWAARDE AAN DE ACTIE EN/OF HET RESULTAAT**
  - Wanneer wordt de actie uitgevoerd?
- **ER IS EEN VOORWAARDE AAN DE TEST**
  - Wanneer wordt de test uitgevoerd?
- **JE REGISTREERT HET RESULTAAT**

# Testen uitschrijven of niet

- **DE VRAGEN DIE JE HIER MOET STELLEN IS:**

- Tussen welke opties kun je kiezen?
- Wat is er optimaal voor ons project?

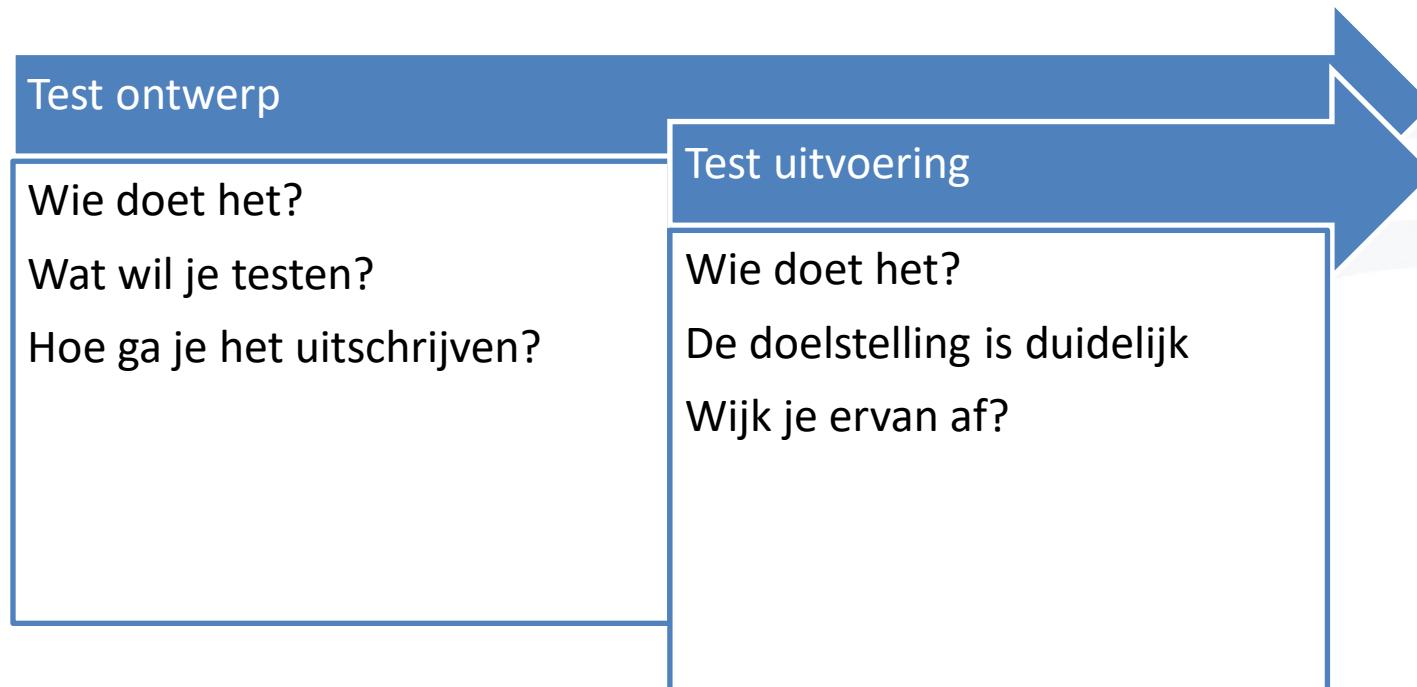
- **VERKENNEND TESTEN <- -> SCRIPT TESTEN**

- Exploratory testen (Click here, click there, click everywhere)
  - Geen planning op voorhand
  - Geen verslag van de testen nadien.
  - Gaat snel
- Scripted testing
  - Wel planning op voorhand
  - Wel verslag van de testen nadien.
  - Is tijdrovend

# Uitgeschreven testen

- **BESTAAN UIT TWEE VERSCHILLENDEN DELEN**

- De voorbereiding ervan
- Het uitvoeren ervan



# Test ontwerp

- **DIT VERGT EEN SPECIALE ‘SKILL’ EN ‘TECHNIEK’**

- Selecteren wat er moet getest worden uit een hele grote hoop mogelijkheden.
- Uitdenken hoe je dit gaat neerschrijven in een optimale manier.
- Je moet geduldig zijn.
- Je moet kunnen oordelen wat nuttig is en wat niet.
- Je moet je kunnen inleven in de persoon die de test gaat moeten uitvoeren.
- Een duidelijk en volledig “verwacht resultaat” is moeilijk.

# Verwacht resultaat

- **EEN SIMPEL PROBLEEM**
  - Je test een rekenmachine
  - En je tikt “ $2+2=$ ”
- **WAT IS HET VERWACHTE RESULTAAT?**

# Test uitvoeren

- **DIT VERGT EEN SPECIALE ‘SKILL’ EN ‘TECHNIEK’**

- Is moeilijker dan verkennend testen, tenzij je verwacht dat de test uitvoering het script exact gaat volgen.
- Je verwacht ook dat de ontwerper en de uitvoerder dezelfde kennis en skill hebben.
- Om het goed te doen, moet de uitvoerder weten wat de bedoeling van de ontwerper was.
- De uitvoerder moet het tactisch inzicht en de oorspronkelijke bedoeling van het test scenario respecteren.

# Test ontwerpen en uitvoeren

- **HET LIJKT ONLOGISCH DAT ...**

- Ontwerp en uitvoeren is stukken harder / moeilijker dan verkennend testen.
- Je één activiteit in twee stukken splitst en dat daarna samen laat werken.

# Best practices?

- **HET IS LOGISCH DAT ...**

- Mensen mij die vraag stellen
- De probleem zit niet in welke stijl van testen het beste is, maar welke stijl ons de informatie geeft die we nodig hebben, op dat moment.
- Projecten hebben meestal andere types informatie nodig in de verschillende fases.

- **DOEL VAN TESTEN IS ...**

- Exact weten wat de risico's zijn.
- En dus de hoeveelheid "Wat je niet weet" verminderen.

# Heuristisch (richtlijnen en strategie)

- **ALS HET SNEL MOET GAAN**
  - Verkennend testen.
- **ALS ER NOG VEEL ONDUIDELIJK IS**
  - Verkennend testen.
- **ALS JE TRACK EN TRACE NODIG HEBT**
  - Testen uitschrijven en verkennend testen
- **ALS JE LATER DE TESTEN WILT AUTOMATISEREN**
  - Testen uitschrijven
- **ALS JE FUN WILT HEBBEN**
  - Combineer en probeer

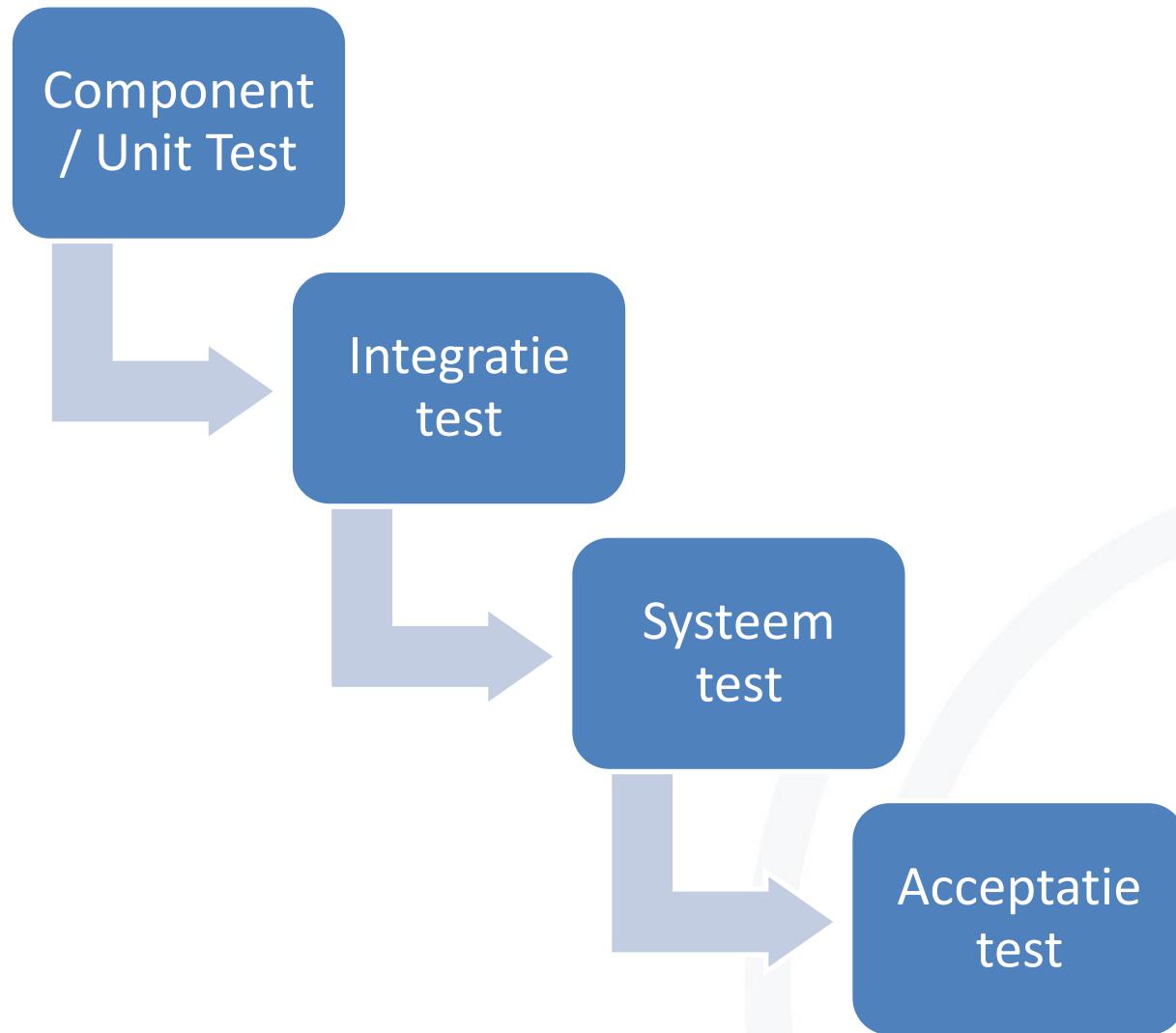
# Heuristisch (richtlijnen en strategie)

- **EEN HEURISTIEK IS EEN FEILBARE METHODE OM EEN PROBLEEM OP TE LOSSEN OF EEN BESLISSING TE NEMEN.**
  - Als bijvoeglijk naamwoord suggereert 'heuristiek' tegelijkertijd 'feilbaar' en 'bevorderlijk voor leren'.
  - Heuristische redeneringen worden niet als definitief en strikt beschouwd, maar alleen als voorlopig en plausibel, met als doel de oplossing voor het huidige probleem te ontdekken.

# Heuristisch (richtlijnen en strategie)

- **HEURISTIEKEN KUNNEN DE VORM AANNEMEN VAN RICHTWOORDEN, AFORISMEN, MODELLEN OF PROCEDURES**
  - Een heuristisch model is een weergave van een idee, object of systeem door een ander - vermoedelijk eenvoudiger - idee, object of systeem, zodat het begrijpen van iets over het eenvoudigere systeem u kan helpen het meer complexe te begrijpen.
  - Een heuristische procedure is een plan van aanpak dat kan helpen bij het oplossen van een categorie problemen.
  - Heuristieken worden niet gevolgd; ze worden toegepast als middel om problemen op te lossen in onzekere omstandigheden met onzekere, onvolmaakte of onvolledige informatie.
- **ZE KUNNEN VOLLEDIGE EN RIGOUREUZE ANALYSE VERVANGEN WANNEER DAT SOORT ANALYSE ERG DUUR OF ONMOGELIJK ZOU ZIJN.**

# Test niveaus

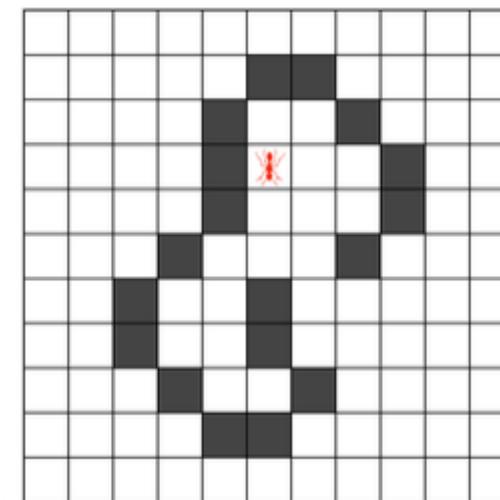


# Test niveaus: Integratie test

- **CHECK WAAR HET STAAT IN HET V-MODEL**
- **WAT IS HET?**
  - Een niveau in het test verhaal, waar individuele componenten of units worden gecombineerd, alsof ze het eindproduct waren.
  - We proberen componenten te testen als een groep. Hoe werken ze samen?
  - Het doel is na te gaan de units / de componenten / de onderdelen correct samenwerken, zoals het ontworpen is.

# De requirements in de verschillende methodes

- DE MIER VAN LANGTON
  - SOFTWARE DEVELOPMENT LIFE CYCLE
    - SDLC
    - Stappen in het maken van de software
  - EEN AANTAL METHODOLOGIEËN
1. WATERVAL MODEL
  2. V-MODEL
  3. SPIRAAL MODEL  
Incrementeel
  4. **ITERATIEF MODEL**  
Iteratief
  5. AGILE / SCRUM  
Incrementeel iteratief
  6. DEVOPS  
Het realiseren van incrementeel iteratief



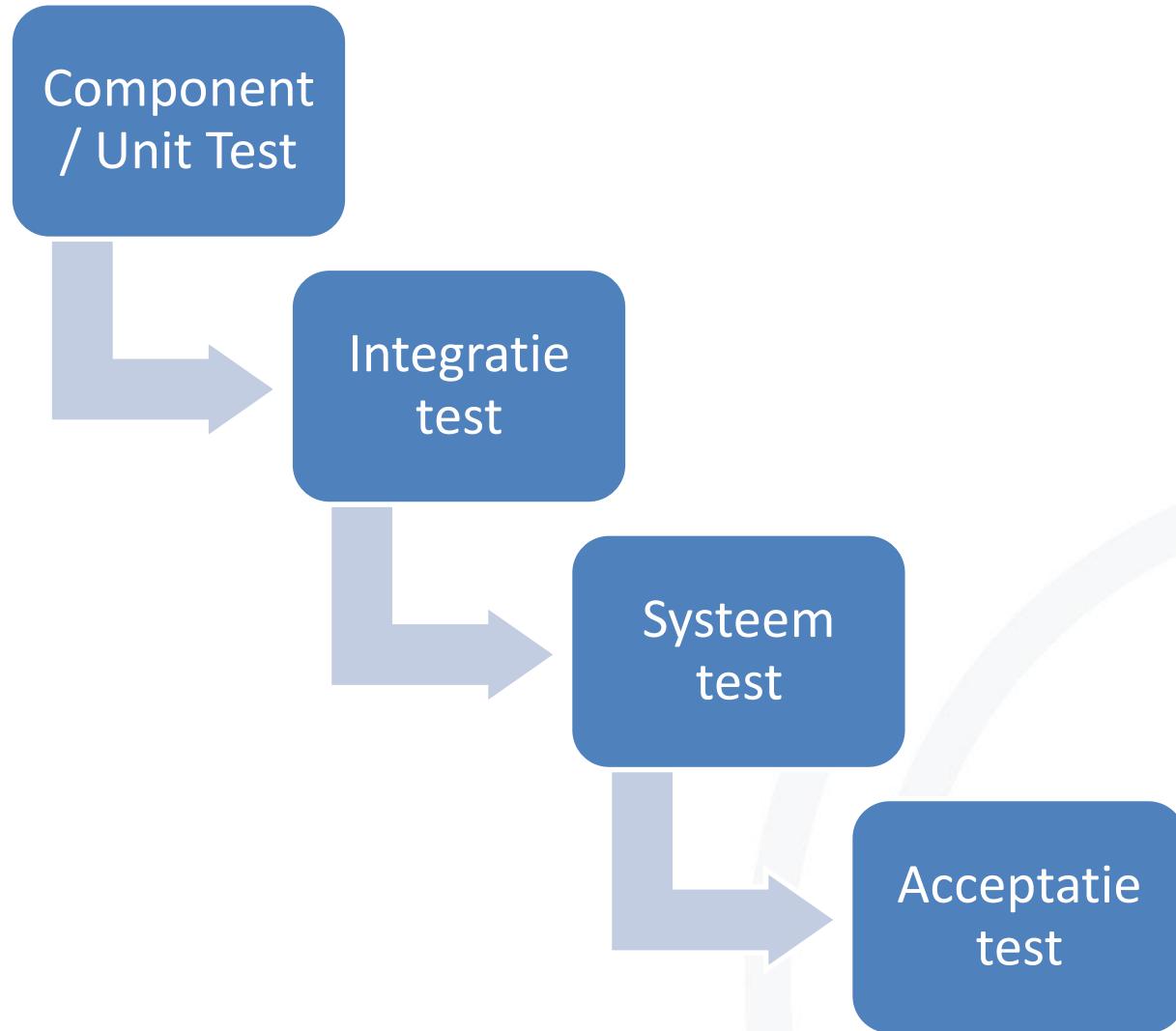
# Een test, iets uitgebreider

- **JE GAAT EEN ACTIE UITVOEREN (EEN STAP)**
  - Doe dit, doe dat, doe iets
- **JE CONTROLEERT OF HET GEDRAG JUIST IS**
  - Hoe reageerde de software, het systeem, het proces?
- **ER IS EEN VOORWAARDE AAN DE ACTIE EN/OF HET RESULTAAT**
  - Wanneer wordt de actie uitgevoerd?
  - Wanneer is het resultaat juist of fout?
- **ER IS EEN VOORWAARDE AAN DE TEST**
  - Wanneer wordt de test uitgevoerd?

# Een test, kan een andere test oproepen

- **JE GAAT EEN ACTIE UITVOEREN (EEN STAP)**
  - Doe dit, doe dat, doe iets
    - Deze stap kan een actie zijn
    - Deze stap kan een andere test zijn
- **DEZE STAP IS EEN ANDERE TEST**
  - Die uit meerdere andere stappen kan bestaan.
  - Of uit andere testen
  - Of een combinatie van de twee.

# Test niveaus

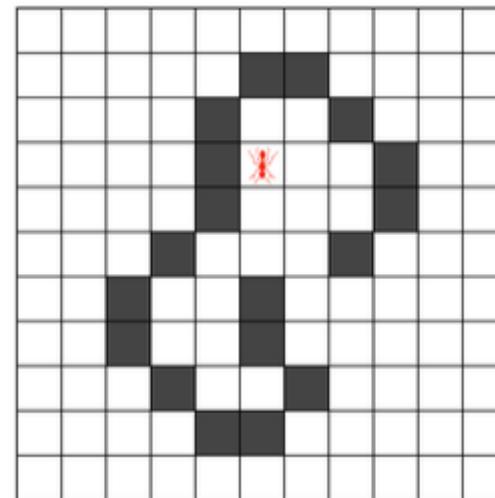


# Test niveaus: Systeem test

- **CHECK WAAR HET STAAT IN HET V-MODEL**
- **WAT IS HET?**
  - Een niveau in het test verhaal, waar het geheel wordt getest, alsof een het eindproduct was.
  - Werkt het programma?
  - Voldoet het programma aan de opgesomde requirements (vereisten van de software)?

# De requirements in de verschillende methodes

- **DE MIER VAN LANGTON**
- **SOFTWARE DEVELOPMENT LIFE CYCLE**
  - SDLC
  - Stappen in het maken van de software
- **EEN AANTAL METHODOLOGIEËN**
  1. **WATERVAL MODEL**
  2. **V-MODEL**
  3. **SPIRAAL MODEL**  
Incrementeel
  4. **ITERATIEF MODEL**  
Iteratief
  5. **AGILE / SCRUM**  
Incrementeel iteratief
  6. **DEVOPS**  
Het realiseren van incrementeel iteratief
- **EEN MOGELIJKE SITUATIE**

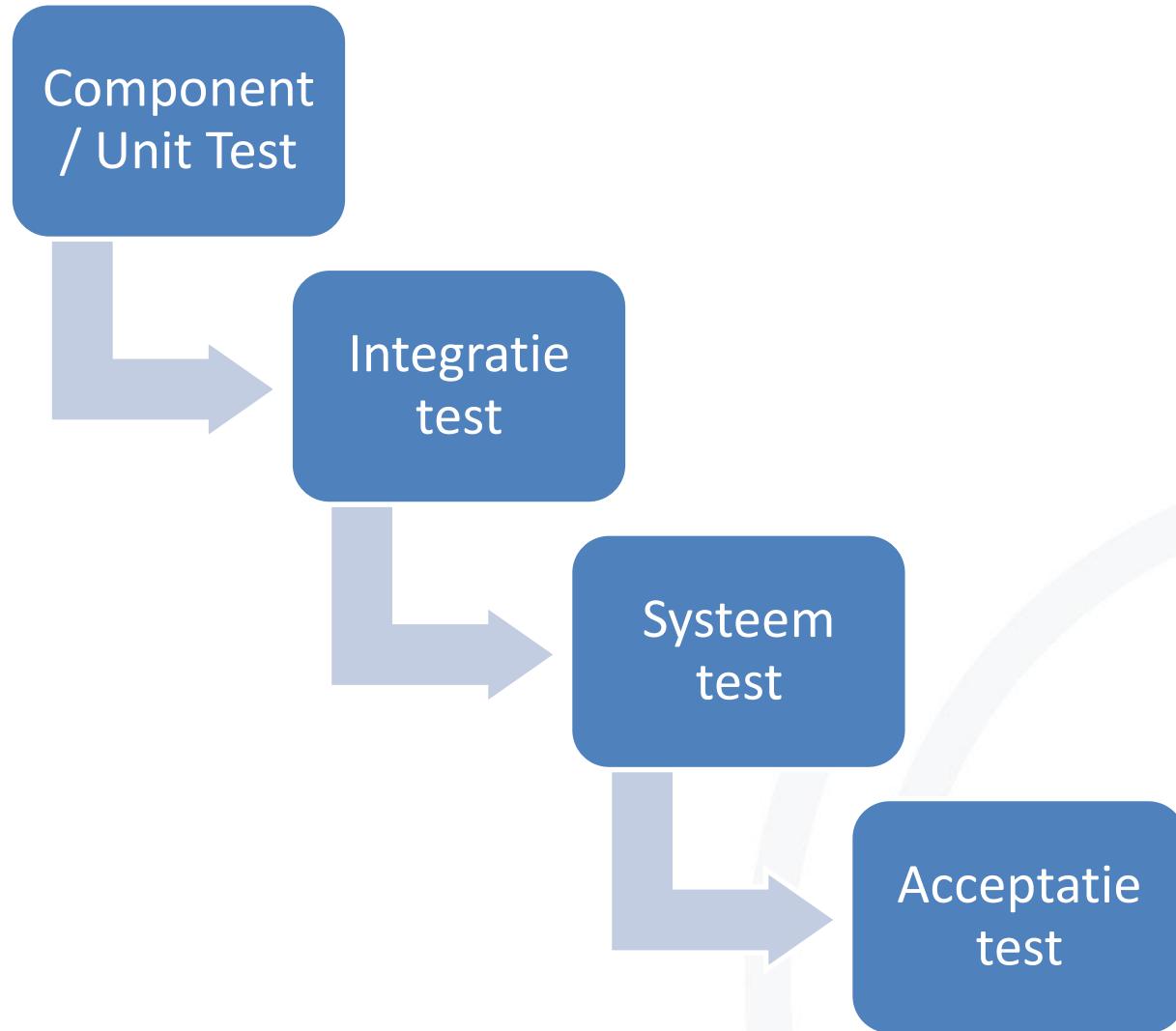


# De code is een misdaad onderzoek

- VERGELIJK DE CODE MET EEN STAD



# Test niveaus

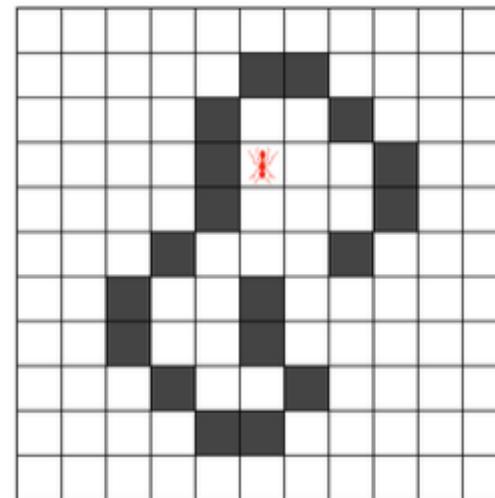


# Test niveaus: Acceptatie test

- **CHECK WAAR HET STAAT IN HET V-MODEL**
- **WAT IS HET?**
  - Een niveau in het test verhaal, waar het business proces wordt getest.
  - Is het programma bruikbaar?
  - Helpt de gemaakte software om het business proces uit te voeren?
- **BUSINESS PROCES?**
  - Documentatie in BPMN

# De requirements in de verschillende methodes

- **DE MIER VAN LANGTON**
- **SOFTWARE DEVELOPMENT LIFE CYCLE**
  - SDLC
  - Stappen in het maken van de software
- **EEN AANTAL METHODOLOGIEËN**
  1. **WATERVAL MODEL**
  2. **V-MODEL**
  3. **SPIRAAL MODEL**  
Incrementeel
  4. **ITERATIEF MODEL**  
Iteratief
  5. **AGILE / SCRUM**  
Incrementeel iteratief
  6. **DEVOPS**  
Het realiseren van incrementeel iteratief
- **EEN MOGELIJKE SITUATIE**



# Een test, kan parameters hebben

- **JE GAAT EEN ACTIE UITVOEREN (EEN STAP)**
  - Doe dit, doe dat, doe iets
  - Binnen in de stap, kan er een parameter gedefinieerd zijn.
    - Voorwaarde
    - Actie
    - Resultaat
- **WANNEER EEN PARAMETER GEEN WAARDE HEEFT, DAN MOET DEZE WAARDE GEVRAAGD WORDEN, NET VOOR HET UITVOEREN VAN DE TEST.**

# En om het af te werken

- **EEN TEST KAN EEN ANDERE TEST OPROEPEN, DIE PARAMETERS HEEFT**
  - Doe dit, doe dat, doe iets
  - Binnen in de stap, kan er een parameter gedefinieerd zijn.
    - Voorwaarde
    - Actie
    - Resultaat
  - Deze parameters zijn al dan niet ingevuld met een waarde.
- **WANNEER EEN PARAMETER GEEN WAARDE HEEFT, DAN MOET DEZE WAARDE GEVRAAGD WORDEN, NET VOOR HET UITVOEREN VAN DE TEST.**

# Dit in actie

- **IK KAN HIER GELIJK WELKE TESTTOOL VOOR GEBRUIKEN**
- **MAAR NIET ALLE TESTTOOLS KUNNEN ALLE SCENARIOS AAN.**
- **WE STARTEN MET EEN PROOF OF CONCEPT**
  - Iets dat het systeem aan jullie duidelijk kan maken.
- **EEN STOMME EXCEL, DAT ALS VOORBEELD DIENT.**

# Schrijven van test scenario's

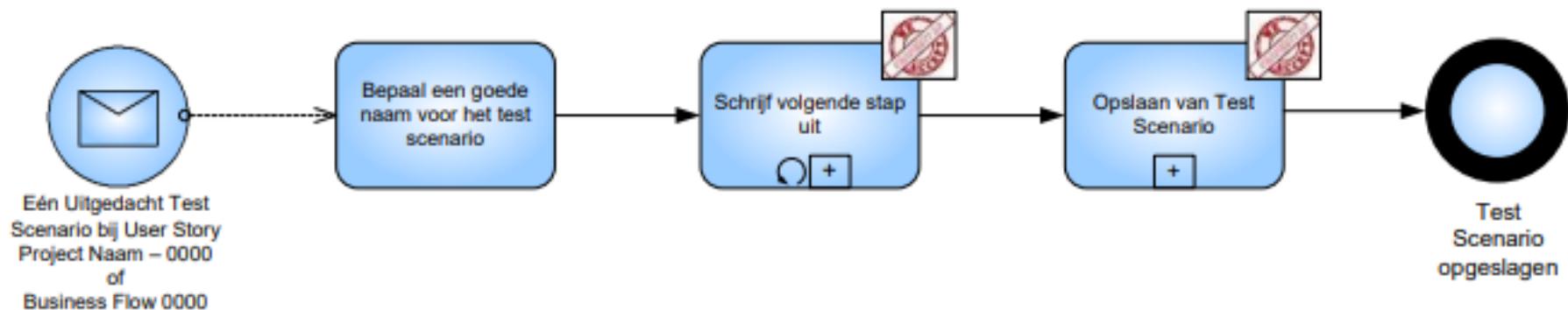
- **ER ZIJN 3 REGELS**
- **REGEL 1: ALLE GEGEVENS 1 KEER**
- **REGEL 2: LEER REGEL NUMMER 1**
- **REGEL 3: ALS JE REGEL 1 GOED KENT, EN KUNT TOEPASSEN,  
MAG JE ERVAN AFWIJKEN.**

# Het maken van testscenario's in BPMN

Regel 1: Alle gegevens één keer.

Regel 2: Leer Regel 1.

Regel 3: Als je Regel 1 goed kent, en kunt toepassen, mag je ervan afwijken.



# De opdracht (voor 40% van de punten)