

leren. durven. doen.



SOFTWARE TESTEN

*Grondbeginselen / Basisprincipes
8 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

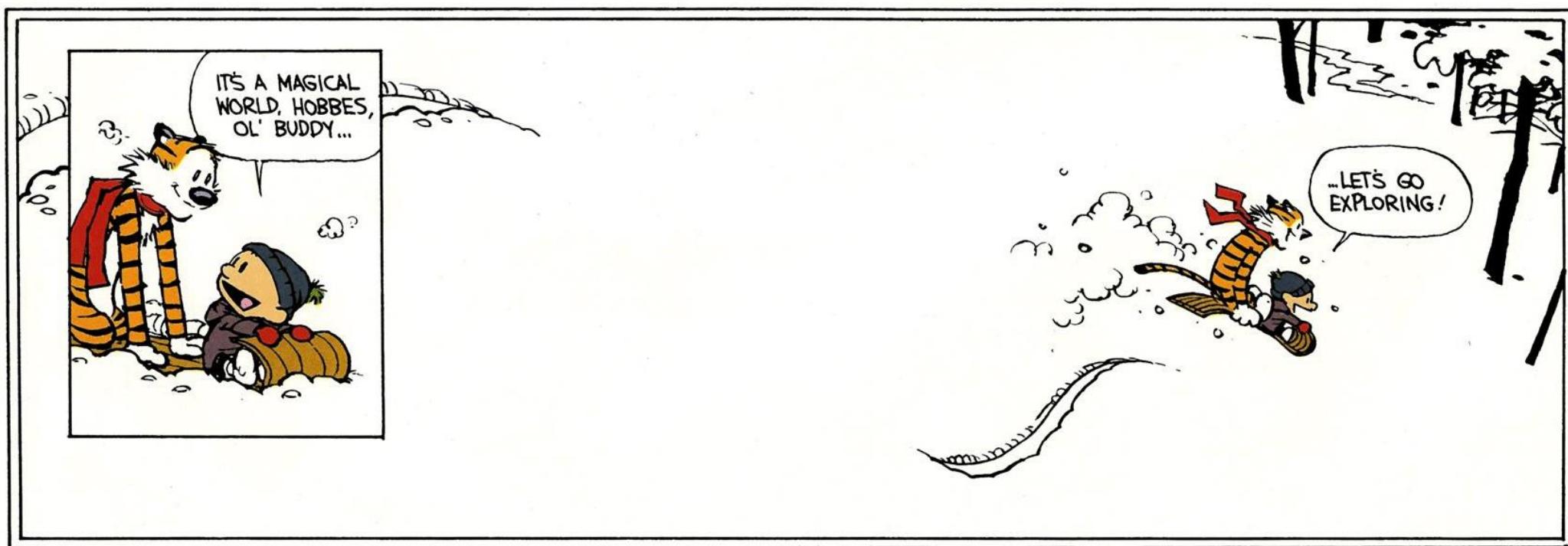
Vincent@CopyPaste.be

VincentVanDeWalle@digi4u.be

+32 (0) 475 / 97.63.80



Knowing me, knowing you...



Voorwaarden – Vereisten (Uk: Requirements)

- **STIJLOEFENING (+/- 15 MIN)**

- De verkoop van een 2 euro stuk.
 - Opdracht wordt niet verteld, maar getikt.

Wat is testen?

- WAT DENK JE DAT DIT IS?



Wat is het doel van testen?

- WAT DENK JE DAT DIT IS?



Agenda (Grondbeginselen dit deel)

- KRITISCH DENKEN
- VOORWAARDEN – VEREISTEN (UK: REQUIREMENTS)
- WAT IS TESTEN?
- KWALITEIT VAN SOFTWARE (UK: SOFTWARE QUALITY)
- EEN HOOP DLW
 - Drie Letter Woorden

Agenda (Grondbeginselen dit deel)

- VERKENNEND TESTEN (UK: EXPLORATORY TESTING)
- DE STRUCTUUR VAN VERKENNEND TESTEN
- HOE FOUTEN ONTDEKKEN?
- WETEN WAT JE MOET TESTEN?
- UITGESCHREVEN TESTEN (UK: SCRIPTED TESTING)
- ALGEMENE TEST TECHNIEKEN
 - Niveaus – Methodes - Types

Agenda (Grondbeginselen andere delen)

- **TEST DOCUMENTATIE**
- **MANAGEN VAN UITGESCHREVEN TESTEN**
 - SBTM (Sessie gebaseerd) – TBTM (Actie gebaseerd) – xBTM (beide)
- **MANAGEN VAN VERKENNEND TESTEN**
 - SBTM (Sessie gebaseerd) – TBTM (Actie gebaseerd) – xBTM (beide)
- **HEURISTISCH TESTSTRATEGIEMODEL (Uk: HEURISTIC TEST STRATEGY MODEL)**
- **PROJECT OMGEVING EN CONTEXT (Uk: PROJECT ENVIRONMENT AND CONTEXT)**
- **MODELLEREN VAN HET PRODUCT (Uk: MODELLING THE PRODUCT)**

Voorwaarden – Vereisten (Uk: Requirements)

- **VIDEO (+/- 10 MIN)**
 - Het smeren van een boterham met pindakaas en jam

Voorwaarden – Vereisten (Uk: Requirements)

- **TEKSTVERWERKING (+/- 15 MIN)**
 - Hoe zet je iets in het vet?

Voorwaarden – Vereisten (Uk: Requirements)

- **REKENBLAD (+/- 15 MIN)**
 - Hoe tel je, bij een waarde van een cel, 1 erbij op?

Wat is testen?

Wat is het doel van testen?

- **JE GSM**
 - Je hebt je GSM laten vallen
 - Misschien is je toestel kapot
- **WAT GA JE DOEN OM TE TESTEN OF JE GSM NOG WERKT?**

leren. durven. doen.



CULTURE EATS STRATEGY FOR BREAKFAST

Peter Drucker

leren. durven. doen.



CONTEXT EATS STRATEGY FOR BREAKFAST

Vincent Van De Walle

Voorwaarden – Vereisten (Uk: Requirements)

- **HET CHINEES TOILETPROBLEEM (+/- 20 MIN)**



Vergeet dit zeker niet

"No battle plan survives contact with the enemy."

- Prussian Army Field Marshal
Helmuth von Moltke



Terminologie en hun betekenis

- **SOFTWARE KWALITEIT**
 - De mate van conformiteit met expliciete of impliciete vereisten en verwachtingen
- **SOFTWARE KWALITEIT – DIMENTIES**
 - Standpunt van de programmeur
 - Standpunt van de gebruiker
 - Termen in alfabetische volgorde
 - Accuraat zijn (Accuracy)
 - Niet hetzelfde als precisie
 - Capaciteit van de gegevens
 - Communicatie
 - Conformiteit
 - Context
 - Compatibiliteit
 - Efficiëntie
 - Flexibiliteit
 - Gebruiksgemak (User Friendliness)
 - Gelijktijdigheid (Concurrency)
- **EEN HOOP VOORBEELDEN**
 - Programma onderdelen
 - Onderhoudsgemak (Maintainability)
 - Snelheid
 - Stabiliteit
 - Structuur
 - Toegankelijkheid (Accessibility)
 - Veiligheid (Security)
 - Volledigheid
 - ...

Terminologie en hun betekenis

- **SOFTWARE KWALITEIT**
 - De mate van conformiteit met expliciete of impliciete vereisten en verwachtingen
- **SOFTWARE KWALITEIT – ZEKERHEID**
 - Software Quality Assurance
 - SQA
- **EEN HOOP VOORBEELDEN**
 - Een reeks van acties en activiteiten zodanig dat je kwaliteit verzekerd in het ontwikkel proces van software / producten / ...
 - Proces gefocused
 - Voorkomen van fouten en problemen

Terminologie en hun betekenis

- **SOFTWARE KWALITEIT**
 - De mate van conformiteit met expliciete of impliciete vereisten en verwachtingen
- **SOFTWARE KWALITEIT – CONTROLE**
 - Software Quality Control
 - SQC
- **EEN HOOP VOORBEELDEN**
 - Een reeks van acties en activiteiten zodanig dat je kwaliteit verzekerd in software / producten / ...
 - Product gefocused
 - Detecteer fouten en problemen

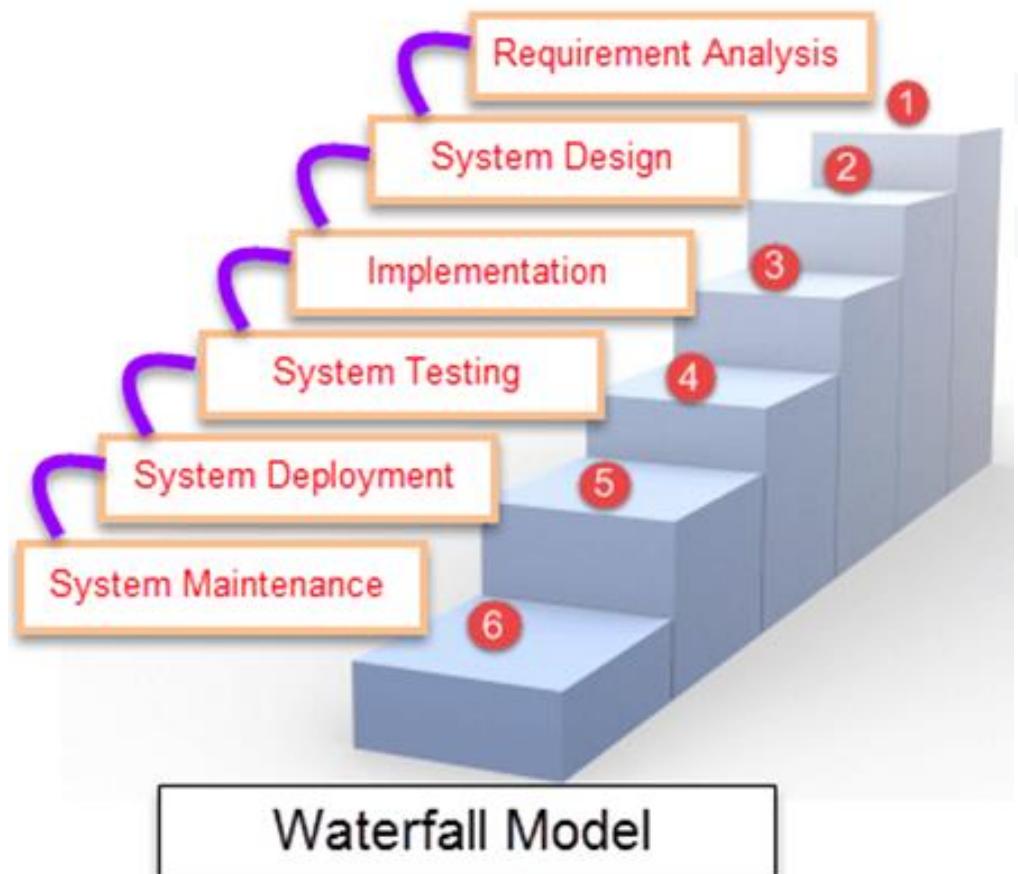
Terminologie en hun betekenis

- **SOFTWARE DEVELOPMENT LIFE CYCLE**
 - SDLC
 - Stappen in het maken van de software
 - **EEN HOOP VOORBEELDEN**
- 1. PLANNEN**
 - Wat wil je doen?
 - 2. ANALYSEREN**
 - Hoe wil je het doen?
 - 3. ONTWERPEN**
 - Hoe ga je het technisch oplossen?
 - 4. IMPLEMENTEREN**
 - Technische uitvoering
 - 5. TESTEN EN INTEGREREN**
 - Checken op correctheid en samenwerking met andere ...
 - 6. ONDERHOUD**
 - Hou het draaiende (keep it running)



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 HOOP VOORBEELDEN**
- **EEN AANTAL METHODOLOGIEËN**

1. WATERVAL MODEL

2. V-MODEL

3. SPIRAAL MODEL

Incrementeel

4. ITERATIEF MODEL

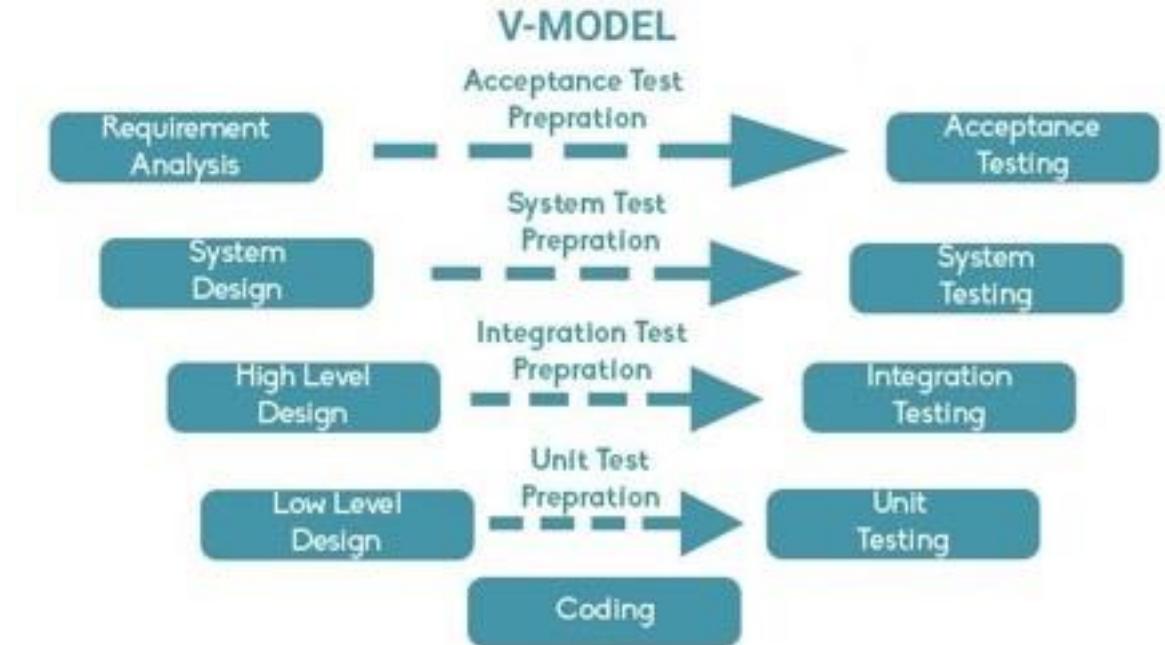
Iteratief

5. AGILE / SCRUM

Incrementeel iteratief

6. DEVOPS

Het realiseren van incrementeel iteratief



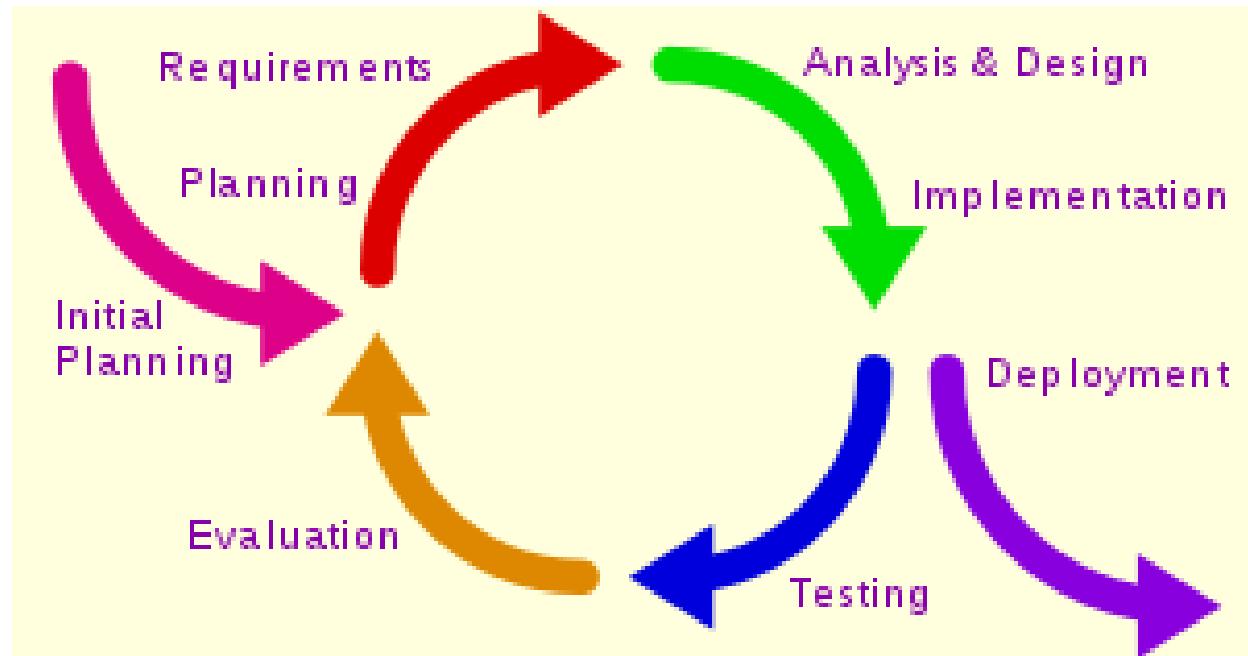
Terminologie en hun betekenis

- **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
- **EEN HOOP VOORBEELDEN**



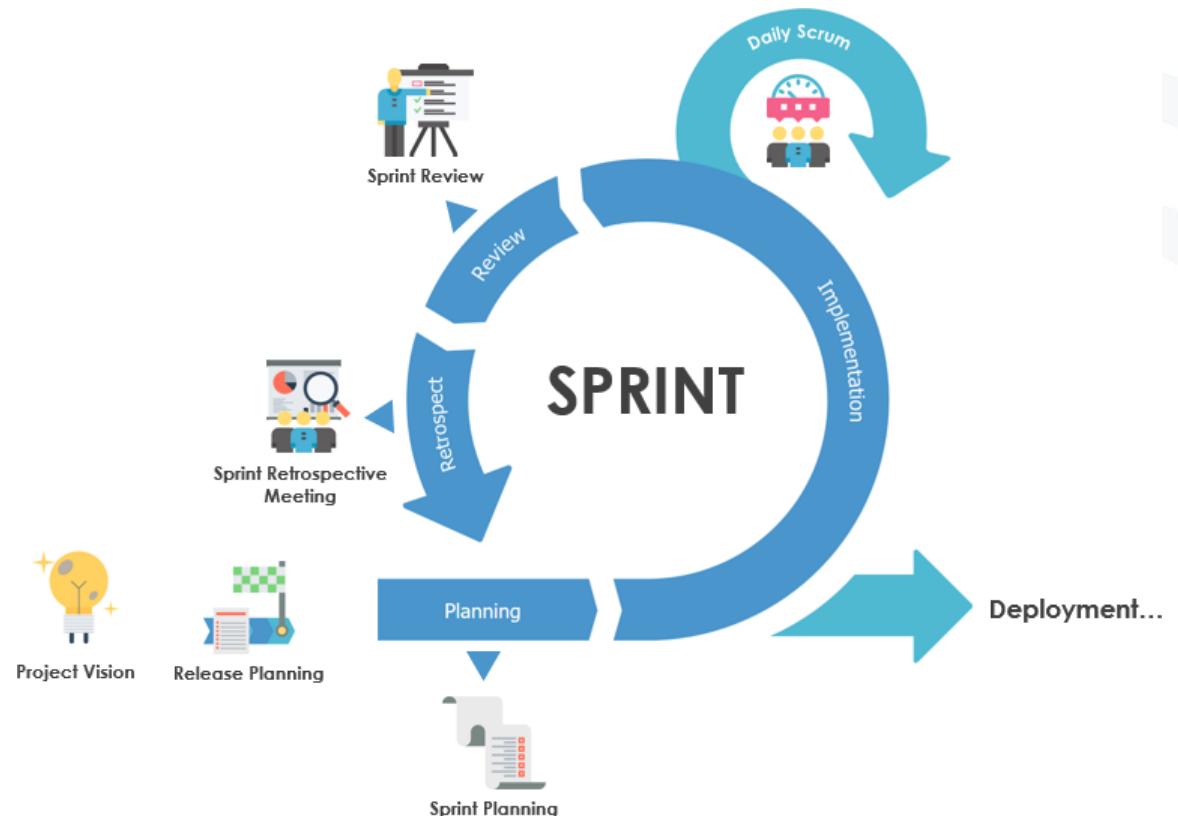
Terminologie en hun betekenis

- **SOFTWARE DEVELOPMENT LIFE CYCLE**
 - SDLC
 - Stappen in het maken van de software
- **EEN HOOP VOORBEELDEN**
- **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
- **EEN HOOP VOORBEELDEN**



Terminologie en hun betekenis

- **SOFTWARE DEVELOPMENT LIFE CYCLE**
 - SDLC
 - Stappen in het maken van de software
- **EEN HOOP VOORBEELDEN**
- **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
- **EEN HOOP VOORBEELDEN**



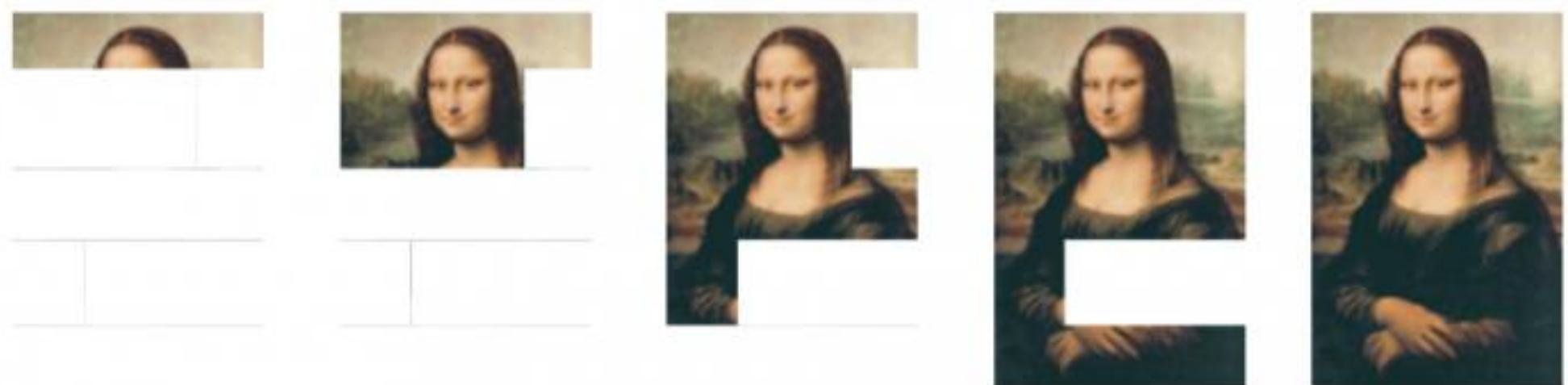
Terminologie en hun betekenis

- WATERVAL



Terminologie en hun betekenis

- INCREMENTEEL



Terminologie en hun betekenis

- INCREMENTEEL



Terminologie en hun betekenis

- ITERATIEF

1



2



3



4

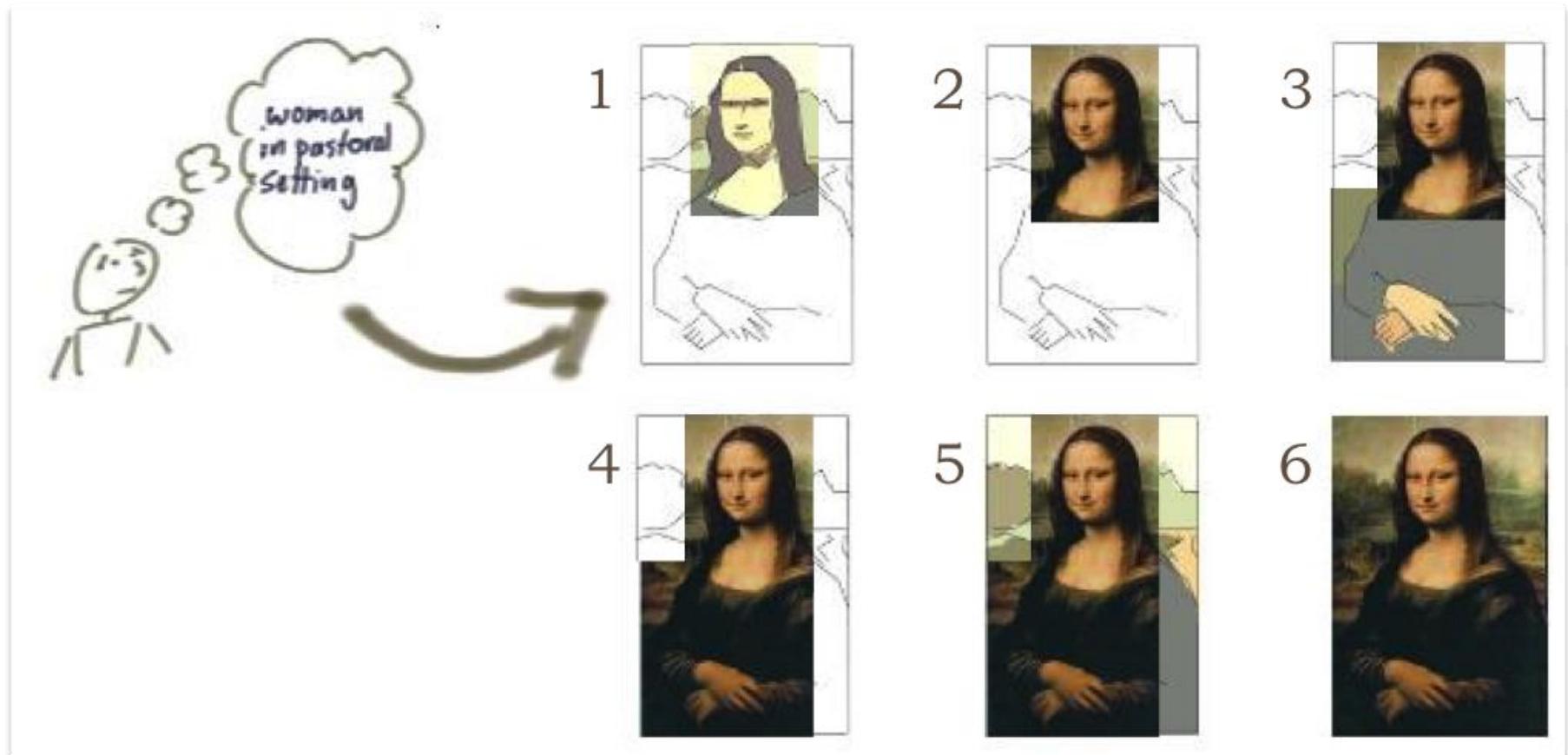


5



Terminologie en hun betekenis

- INCREMENTEEL ITERATIEF



Terminologie en hun betekenis

- NOG BETER?



Terminologie en hun betekenis

- IN DE PRAKTIJK

Agile as usually practiced

Release 1



Release 2



Release 3



Agile UX

Release 1



Release 2

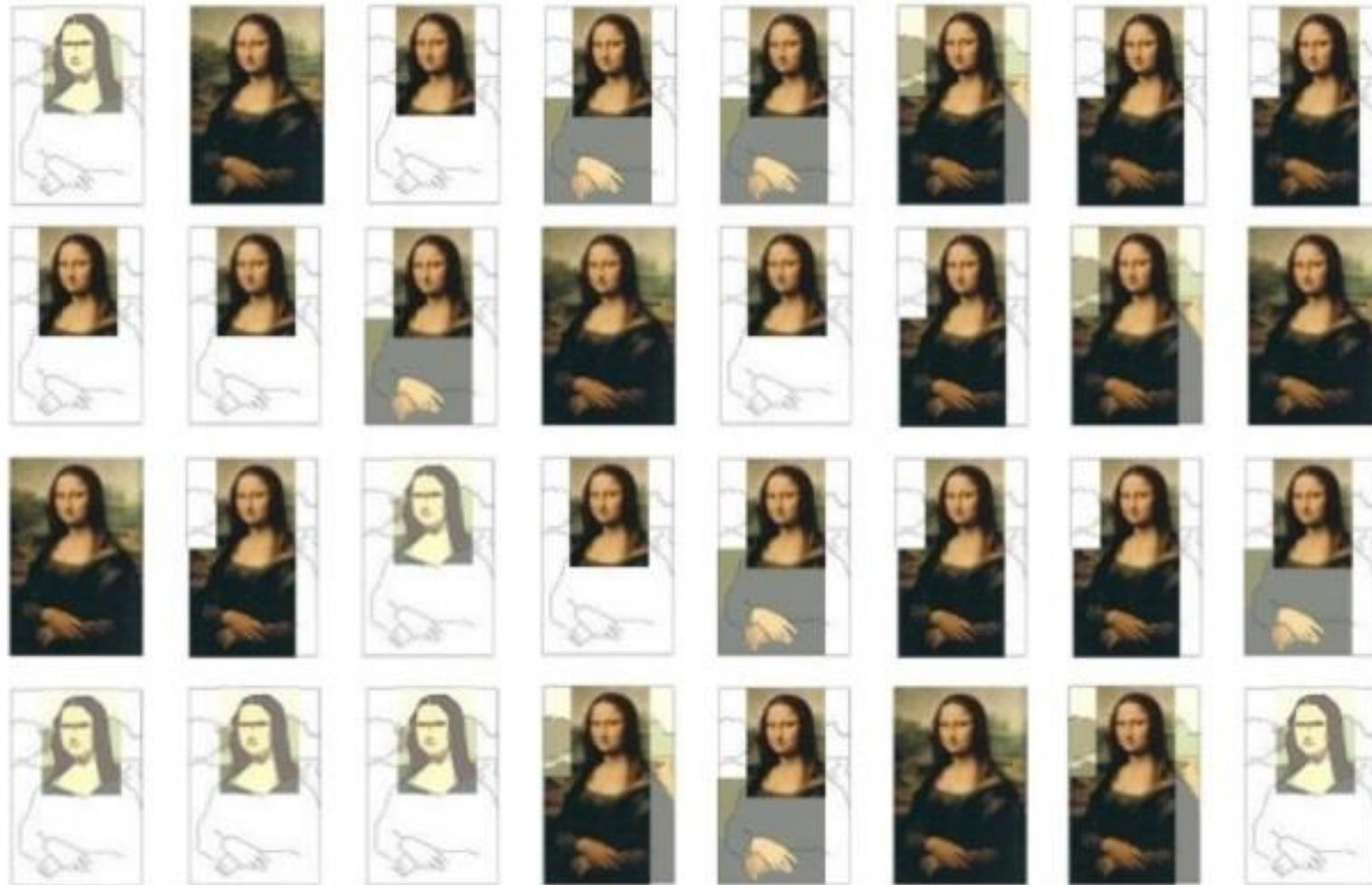


Release 3



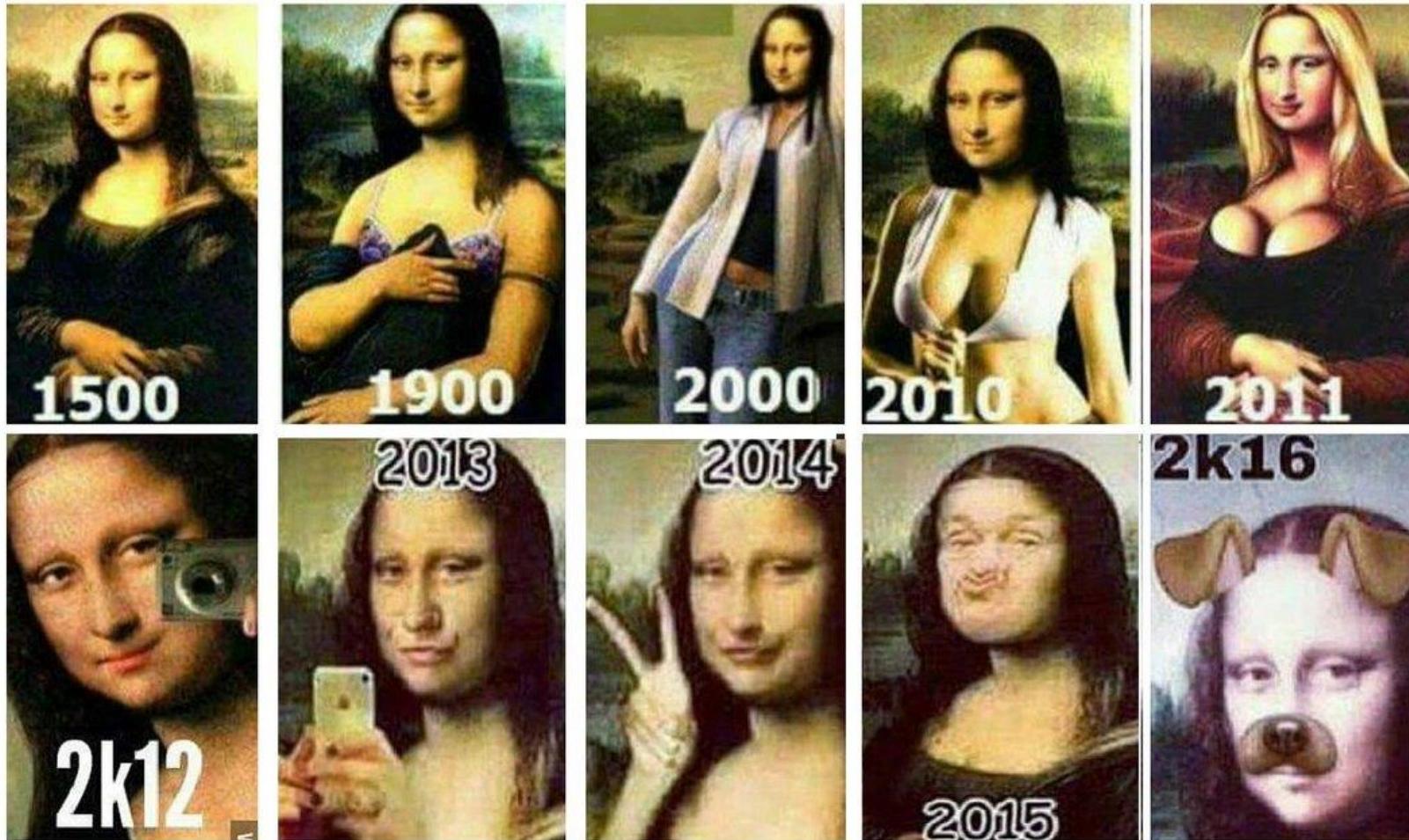
Terminologie en hun betekenis

- **ONTWIKKELING IN PARALEL**



Terminologie en hun betekenis

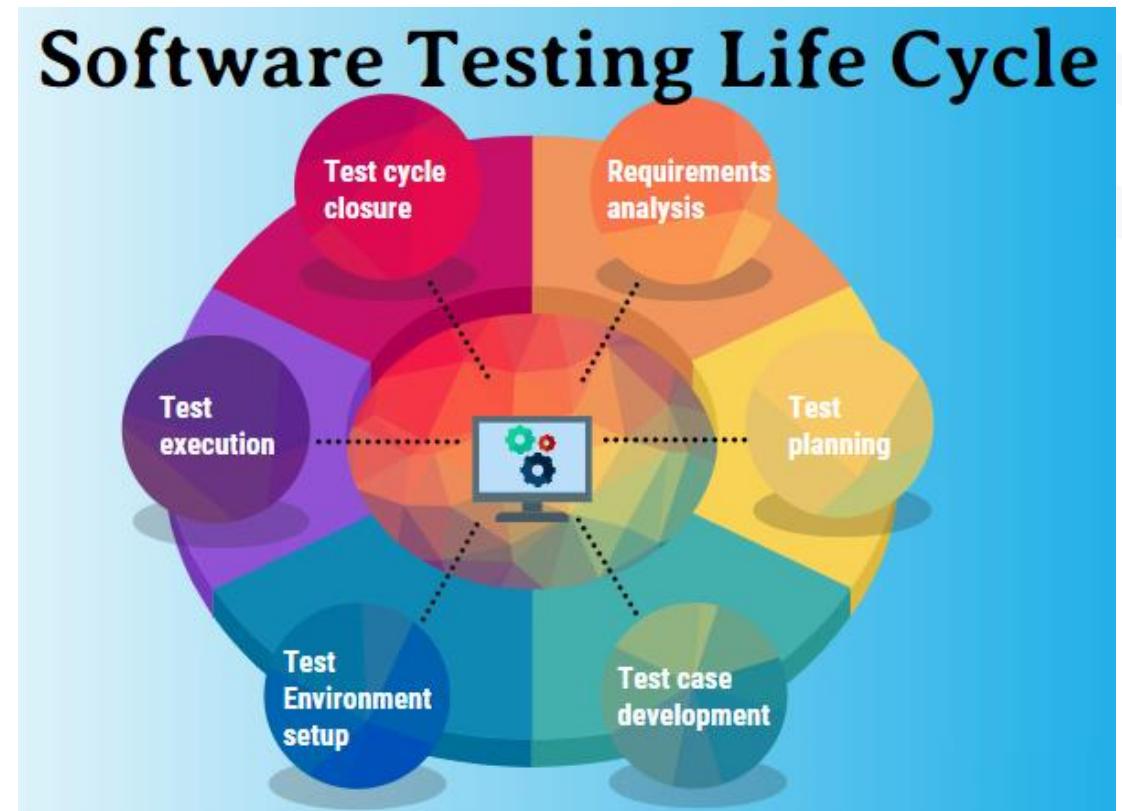
- **TEGENWOORDIG**



Terminologie en hun betekenis

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

- **EEN HOOP VOORBEELDEN**



Wat is een computer programma?

- WAT DENK JE DAT DIT IS?



Wat is een computer programma?

- EEN SET VAN INSTRUCTIES VOOR EEN COMPUTER.



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

```
String hql = "from Person p where 1=1";
int count = 0;
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> list = query.list();
for(Person p : list){
    if(count > 0){System.out.println();}System.out.print(p);
    count++;
}
```

Wat is een computer programma?

- EEN SET VAN INSTRUCTIES VOOR EEN COMPUTER.



© Michael Bolton 2013
How to get what you
really want from testing?

```
String hql = "from Person p where 1=1";
8   if(parameters.contains("name")){
9     hql += " and p.name = :name";
10  }
11  if(parameters.contains("age")){
12    hql += " and p.age = :age";
13  }
14  TypedQuery<Person> query = em.createQuery(hql, Person.class);
15  if(parameters.contains("name")){
16    query.setParameter("name", values[0].toString());
      parameters.contains("age")){
        query.setParameter("age", Integer.valueOf(values[1].toString()));
      }
    list();
  }
```

Een huis

- **EEN SET VAN BOUWMATERIALEN, GEORDEND IN EEN “HUIS” ONTWERP PATROON.**



Een huis

- IETS WAAR MENSEN IN LEVEN



leren. durven. doen.

Cem Kaner's definitie 'computer programma'

© Michael Bolton 2013
How to get what you
really want from testing?

- **EEN COMPUTER PROGRAMMA IS**

- Een communicatie
- Tussen verschillende personen
- En computers (systemen)
- Over afstand en tijd
- Dat instructies bevat
- Die uitgevoerd kunnen worden op een computer

Cem Kaner's doel 'computer programma'

- **HET DOEL VAN EEN COMPUTER PROGRAMMA IS**
 - Leveren van toegevoegde waarde aan mensen

© Michael Bolton 2013
*How to get what you
really want from testing?*

Wat wil dit nu zeggen?

© Michael Bolton 2013
How to get what you
really want from testing?

- **HET DOEL VAN EEN COMPUTER PROGRAMMA IS**
 - Leveren van toegevoegde waarde aan mensen
- **GEVOLG**
 - Een computer programma is veel meer dan de code
 - Een software product is veel meer dan de instructies voor een computer
 - Kwaliteit is veel meer dan het ontbreken van fouten in de code
 - Testen is veel meer dan het uitvoeren van test scenario's om een bepaald gedrag te bevestigen
 - Testen is veel meer dan een beetje code / programma / scenario dat bevestigt dat een andere code / programma / scenario correct werkt

Wat is testen?

Wat is het doel van testen?

- TESTEN IS BEVESTIGEN VAN KWALITEIT

Kwaliteit is toegevoegde waarde geven aan
één of meerdere personen
— Jerry Weinberg

Software testen is het onderzoek
van systemen bestaande uit
mensen en hun werk, computers,
programma's en de relatie tussen allen

lets meer realistisch

**Testen is een product in vraag stellen
om het resultaat te kunnen evalueren**

Wat is “perfect” testen?

**Testen is het oneindig proces
het onzichtbare te vergelijken
met het onduidelijke
om het ondenkbare te vermijden
dat een onbekende kan tegenkomen**

- **MET ANDERE WOORDEN**
 - Perfect testen is een onmogelijke uitdaging

Wat is testen?

Probeer het en zie of het werkt.
Probeer genoeg te ontdekken over of het kan werken, over hoe het niet zal werken, en leer of het werkbaar (bruikbaar) is

Procedures	Is alles gecoverd	Voorspellingen
Zijn er verschillende versies?	Waar kijken?	Lees de specificaties?
Zet je procedures op	Hoe kijken?	Zijn die er niet? Oh ☺
Probeer simpele dingen	Wat is er?	Ze zijn er. Oh ☺, maar fout
Probeer complexe dingen	Wat is er niet?	Vind inconsistenties
Probeer volgordes	Wat is merkbaar / zichtbaar?	Vind normale problemen
Probeer combinaties	Is het veranderd?	Vind obscure problemen
Probeer vreemde dingen	Zal het veranderen?	Vind echte problemen
Probeer ze opnieuw	En wat is het nu?	Vertel ook hoe ze op te lossen

Checken versus Testen

- **CHECKEN**

- Het proces van evaluaties maken tussen enerzijds hoe een product werkt, en anderzijds genomen algoritmische beslissingen (geschreven code)
 - Dit kan worden geautomatiseerd

- **TESTEN**

- Het proces van een product evalueren door te ermee te experimenteren.
- Hier kan me informatie bekomen en doorgaat men een leerproces
- Bevraag, bestudeer, modeleer, observeer en beïnvloed het project
 - Niet te automatiseren

Wat is testen?

- **ONDER ANDERE**

- Het product en de ideeën en beschrijvingen ervan **herzien**, op zoek naar significante en relevante **inconsistenties**.
- **Experimenteren** met het product om erachter te komen **hoe het problemen kan hebben** - wat niet 'het product kapot maakt'.
- Iets dat **kwaliteitszorg** informeert, maar op zichzelf geen **kwaliteitszorg** is.
- Onze klanten helpen **geïnformeerde beslissingen te nemen** over het product, project of bedrijf.

Wat is testen?

- **ONDER ANDERE**

- Een proces waarbij we **systematisch** elk aspect van het product **onderzoeken** om verrassingen te voorkomen.
- Een proces van **interactie met het product** en zijn systemen op vele manieren die **ongerechtvaardigd optimisme** uitdagen.
- Het **product observeren, verkennen, experimenteren en evalueren**, om te zien waar al die defectpreventie-ideeën mogelijk hebben gefaald.
- Een speciaal onderdeel van het ontwikkelproces was gericht op het **ontdekken van wat er slecht zou kunnen gaan** (of wat er slecht gaat).

Wat is testen?

- **ONDER ANDERE**

- Het product verkennen, ontdekken, onderzoeken, leren en rapporteren om **nieuwe informatie te onthullen**.
- **Informatie verzamelen** over het product, de gebruikers en de gebruiksvoorwaarden om **de waarde van het product te helpen verdedigen**.
- **Vragen stellen** om teams te helpen producten te ontwikkelen die sneller en gemakkelijker hun eigen problemen blootleggen.
- **Programmeurs en het team helpen** om meer te weten over **onverwachte aspecten** van het product dat we ontwikkelen.

Wat is testen?

- **ONDER ANDERE**

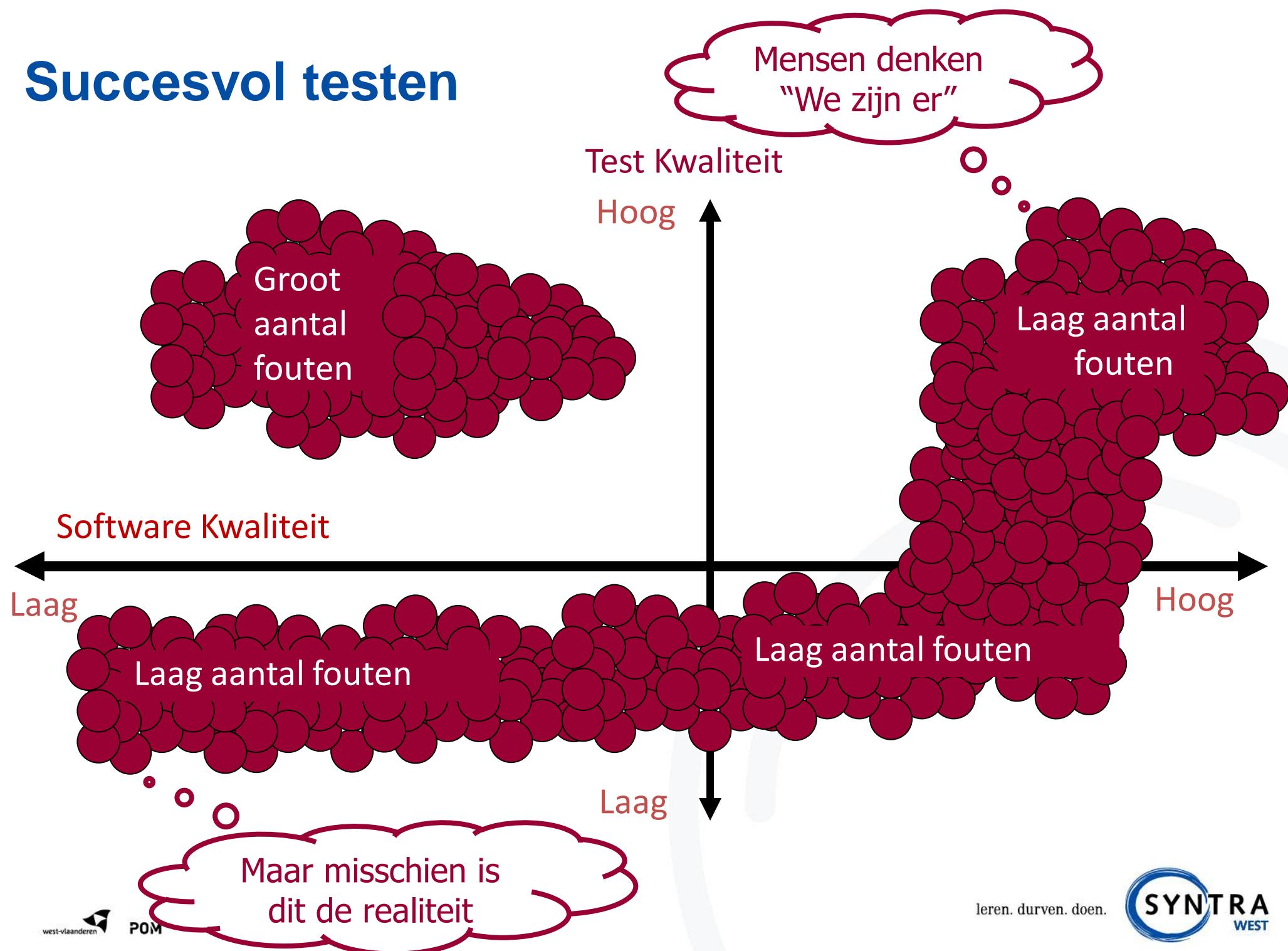
- We helpen onze klanten het product dat ze hebben te begrijpen, zodat ze kunnen beslissen of dit het product is dat ze willen.
- Hulpmiddelen en directe interactie met het product gebruiken om zijn gedragingen en toestanden in twijfel te trekken en te evalueren.
- Producten diepgaand, fantasierijk en achterdochtig onderzoeken om problemen te vinden die de toegevoegde waarde bedreigen.
- Een rommelige zaak.

Wat is testen?

- **ONDER ANDERE**

- Uitvoeren van feitelijke en gedachte-experimenten op producten en ideeën om **problemen en risico's te identificeren**.
- **Kritisch en sceptisch nadenken** over producten en ideeën om ons heen, met als doel niet voor de gek gehouden te worden.
- **Leren** over dingen die iemand wil weten.
- **Uitzoeken of wensen uitkwamen** en daar de **juiste conclusies uit trekken**.

Succesvol testen



De toegevoegde waarde van testen

- IK HEB 25 TESTEN UITGEVOERD.
- IK HEB 100 TESTEN UITGEVOERD EN HET SYSTEEM FAALT 12 KEER
- IK VOND 56 BUGS
- EN DAN?
- TESTEN GEEFT ENKEL WAARDE ALS HET INFORMATIE OPLEVERT. DAT IS VERSCHILLENDE VAN DATA GENEREREN

Testers zijn geen zwartkijkers

- HOEWEL WE “NEGATIEF” KUNNEN OVERKOMEN TEGEN OVER:
 - Ontwikkelaars
 - Project leiders
 - Business mensen
 - Marketing (verkopers)



The tester's role

© Michael Bolton / James Bach
Rapid Software Testing

- **TESTERS VERLICHTEN DE WEG**

- We zien dingen zoals ze zijn.
- We maken gemotiveerde beslissingen mogelijk rond kwaliteit, omdat we kritisch denken over software.

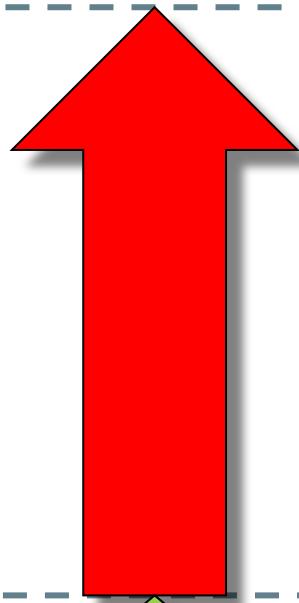


Testers verlichten de weg

© Michael Bolton / James Bach
Rapid Software Testing

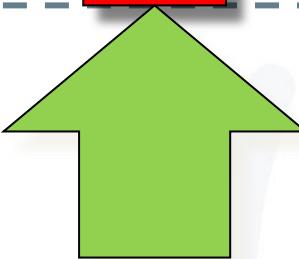
- **DE “RISICO” AFGROND**

Wat we zouden moeten weten



De reden van testen is **het sluiten van** de “Risico” afgrond.
Hoe groter de afgrond is,
Hoe lastiger het product te testen is.

Wat we weten



Wat weten we van de status van ons product



Kwaliteit / Fout (Bug) / Probleem

- **QUALITY / BUG / ISSUE**
- **KWALITEIT**
 - Waarde voor een bepaalde persoon (wie trekt het zich aan)
- **BUG**
 - Alles van het op te leveren product (de op te leveren software) dat de waarde bedreigt
- **PROBLEEM**
 - Alles dat de waarde van een project, de business, of het testen zelf in de weg staat.

Testers, het brein achter een project?

