

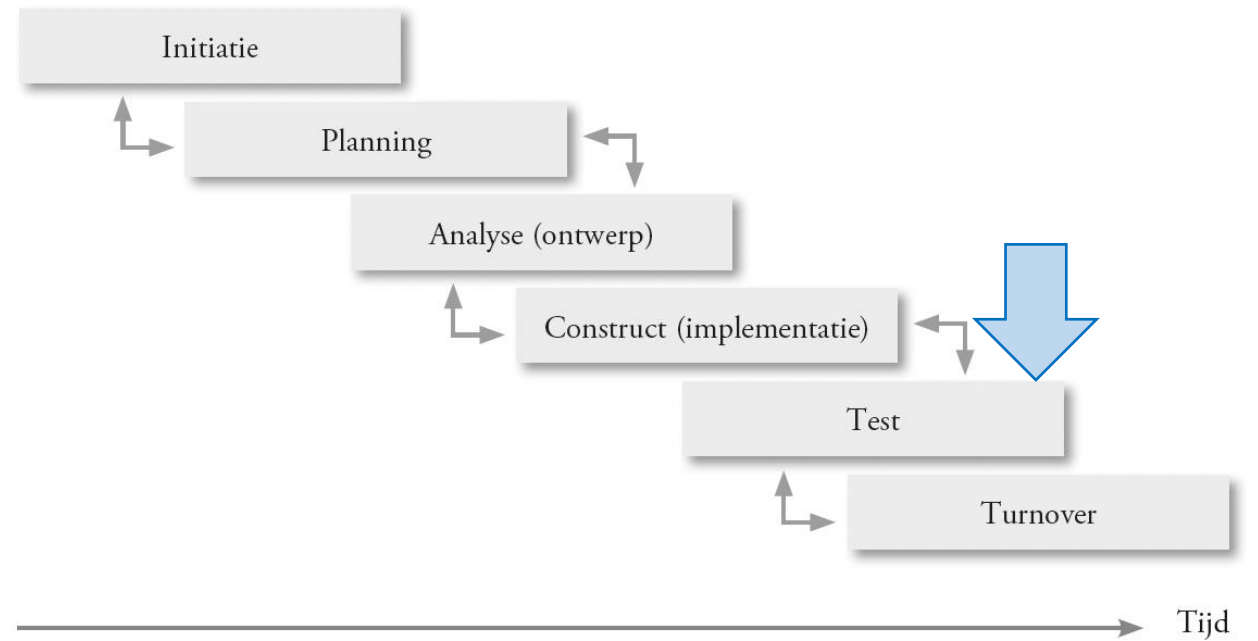


Hoe garanderen we  
de kwaliteit van de  
oplossing?

#Testen

# Projectfases

---



# Leerdoelen – Hoofdstuk 9

Op het einde van dit hoofdstuk:

- ... kan ik een onderscheid maken tussen de **verschillende testtypes en testlevels**
- ... begrijp ik het **belang van testen** en heb ik een volledig beeld van alle onderdelen die getest moeten worden
- ... ken ik de verschillende **onderdelen van een testplan** en weet ik hoe de testresultaten gedocumenteerd moeten worden


# Beschrijving en doel

**Testen** zijn een reeks van handelingen uitgevoerd om te bepalen of de kwaliteit van de geboden oplossing voldoet aan de vereisten en verwachtingen van de uiteindelijke gebruiker.

De testfase is een belangrijke fase gezien het **dé kwaliteitscontrole** is van de ontwikkelde software

- Traditioneel werd testen als minder belangrijk aanzien
- Vroeger werd testen vaak enkel gedaan door de gebruikers
- Testen gebeurt ook nog vaker ongestructureerd dan gestructureerd





370 million dollars worth of fireworks because of a software bug. (Source: ESA)

## Waarom testen?

Beroemde software bugs met grote gevolgen:

- Therac-25 Radiation Therapy Machine (1985 – 1987)
- Ariane 5 Flight (1996)
- Knight Capital Group Trading Loss (2012)
- Healthcare.gov launch (2013)
- Boeing 737 Max Crashes (2018 en 2019)

=> belang van diepgaande software testen in verschillende industrieën

=> zorgen voor betrouwbare, veilige, functionele en schaalbare software systemen

# Redenen voor testen

- Verhoogde **complexiteit** van projecten, processen en technologieën
- Grotere concurrentie op de softwaremarkt gecombineerd met lagere budgetten
  - Bedrijven stappen sneller over naar een andere leverancier als ze het gevoel krijgen dat er prijs/kwaliteit iets niet in orde is
  - Goede **kwaliteit** opleveren zorgt voor een goede naam
- Steeds meer **audits** op softwaresystemen
  - Vooral in de financiële sector
  - GDPR
  - Isonormen





# Redenen voor testen

- Digitale oplossingen zijn **geïntegreerd in het dagelijkse leven**
  - Wanneer de digitale oplossing faalt, kan het dagelijks leven verstoord geraken
  - Treinen rijden niet meer op tijd, overschrijvingen kunnen niet meer gebeuren, lonen worden niet meer uitbetaald, communicatie valt weg,...
- **Eindgebruikers** worden steeds veeleisender
  - Digitale oplossingen zijn alomtegenwoordig
  - Klant heeft veel voorbeelden van goed werkende oplossingen
- Redenen voor gestructureerd testen
  - Testmanagers behouden het **overzicht** en kunnen zorgen dat alles getest wordt
  - Maakt deel uit van het **risicomanagement** en zorgt ervoor dat een deel van de geïdentificeerde risico's kunnen worden afgewend

# Wat wordt er getest?

- Software
- Hardware
- Organisatie en procedures
- Documentatie
- Cutover
- Conversie
- ....





# Testtypes

- Een testtype **test een specifiek deel van de oplossing** en heeft dus **een specifiek doel**:
- **Functionaliteitstesten**
  - Werken de functionaliteiten correct (verificatie)
  - Voldoen de functionaliteiten aan de verwachtingen en de vereisten (validatie)
  - Gebeuren door de eindgebruikers!
- **Cross-platform testen (portabilitytesten)**
  - De software moet kunnen functioneren op verschillende platformen
  - Kan gebeuren door geautomatiseerde tools

# Testtypes

- Performantietesten

- Snel werken is een must vandaag
- Trage applicaties wekken frustratie op, hetgeen de weerstand weer groter zal maken
- Indien nodig moet de applicatie ook getest worden op een groot aantal gebruikers of piekmomenten

- Gebruiksvriendelijkheidtesten / usability testen

- Gebruikers willen resultaat met een minimaal aantal handelingen
- Eindgebruikers moeten dit zeker testen

# Testtypes

- Veiligheidstesten

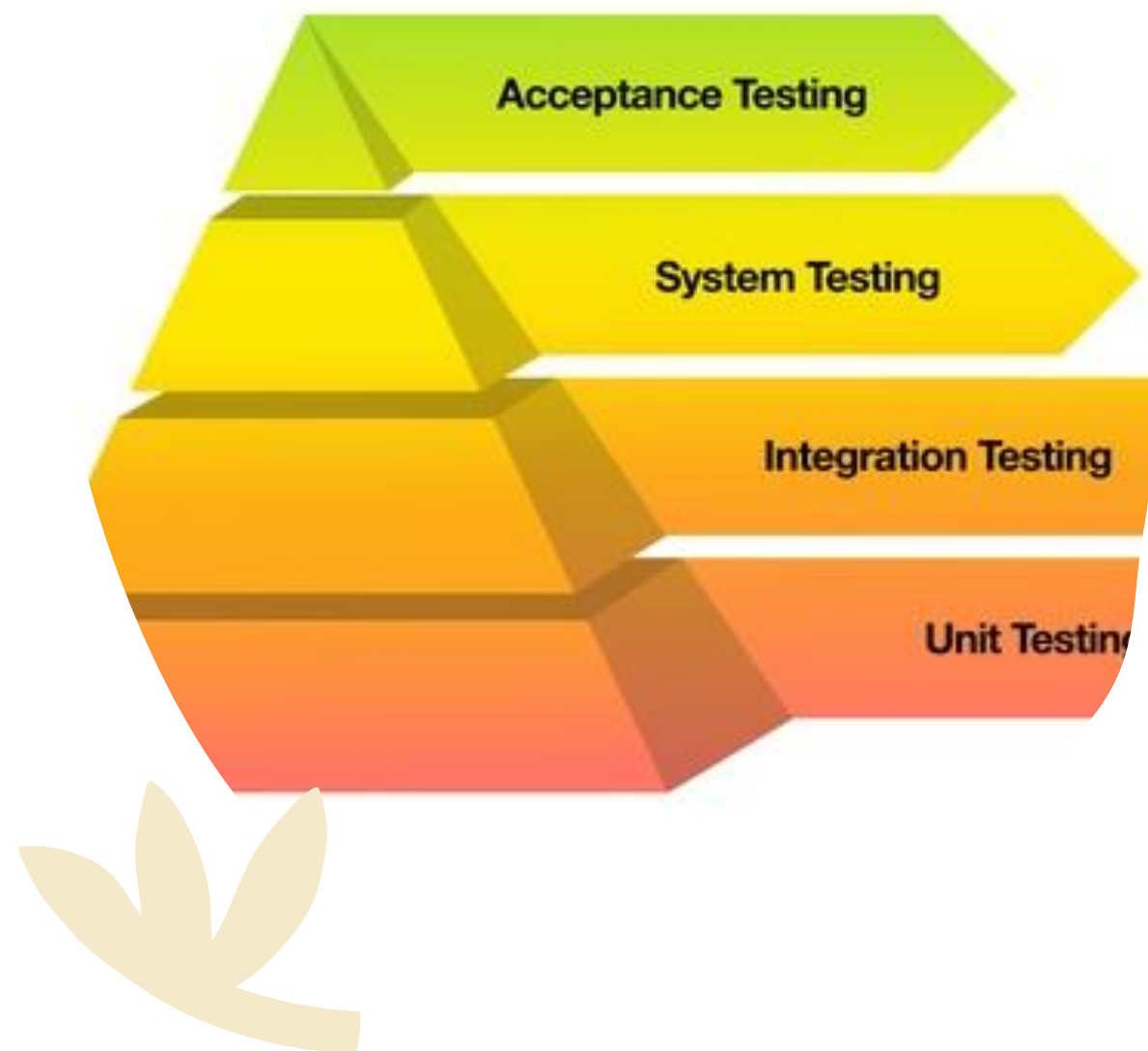
- **Cybersecurity** wordt meer en meer belangrijk
- Niet enkel de gegevens moeten beschermd worden, ook de rollen en autorisaties moeten getest worden
- Rollen kunnen ook in de functionaliteitstesten meegenomen worden

- Regressietesten

- Wanneer er wijzigingen gebeuren, moet de impact op andere functionaliteiten getest worden
- Herbruikbaarheid is een belangrijk principe in software
- Hertesten van alle functionaliteiten en niet enkel diegene die gewijzigd is

# Testlevels

- Een testlevel groepeert een aantal testactiviteiten binnen een bepaalde fase in de ontwikkeling:
- **Unittesten**
  - Controleren afzonderlijke delen van de oplossing
  - Uitgevoerd door ontwikkelaars
  - Testen technische correctheid van de oplossing
- **Integratietesten**
  - Samenwerking tussen de verschillende componenten



# Testlevels

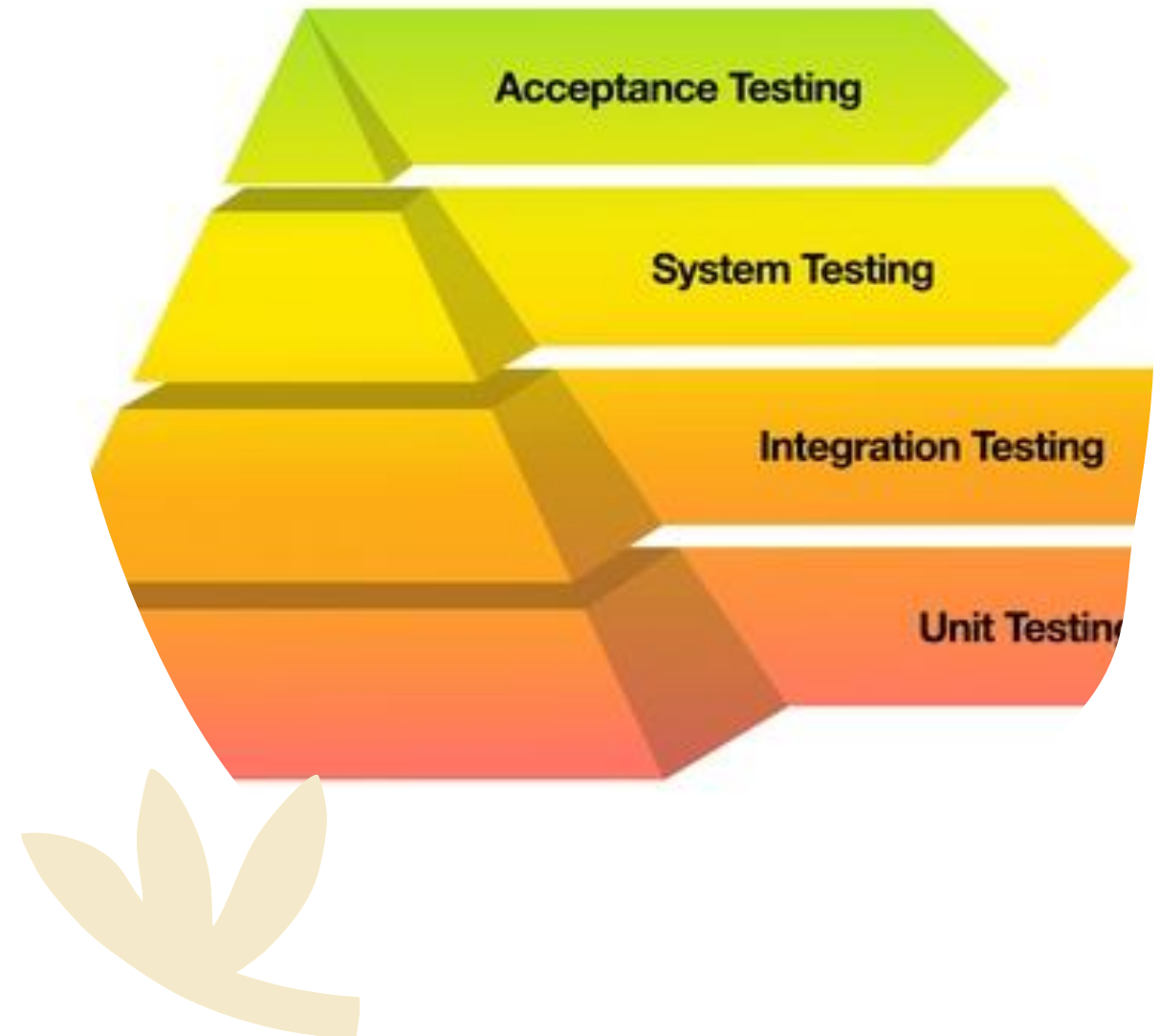
Zie ook: <https://reqtest.com/en/knowledgebase/different-levels-of-testing/>

- **Systeemtesten**

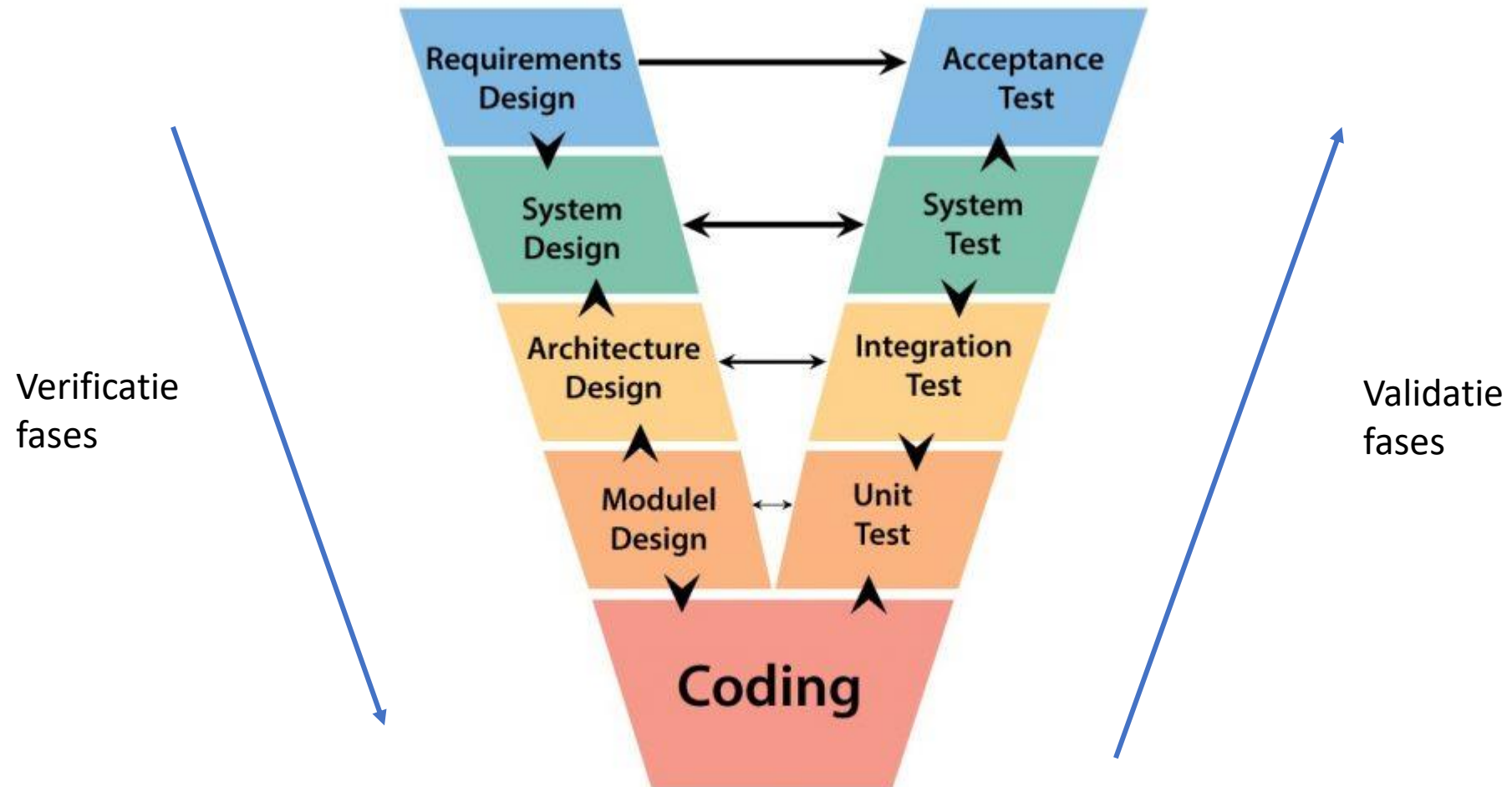
- Testen van de oplossing in zijn geheel
- Niet enkel software maar alle componenten van de oplossing

- **Acceptatietesten**

- Testen door de eindgebruikers om te kijken of de oplossing in gebruik kan worden genomen
- Eindgebruikers testen zowel de correcte werking als het voldoen aan vereisten en verwachtingen



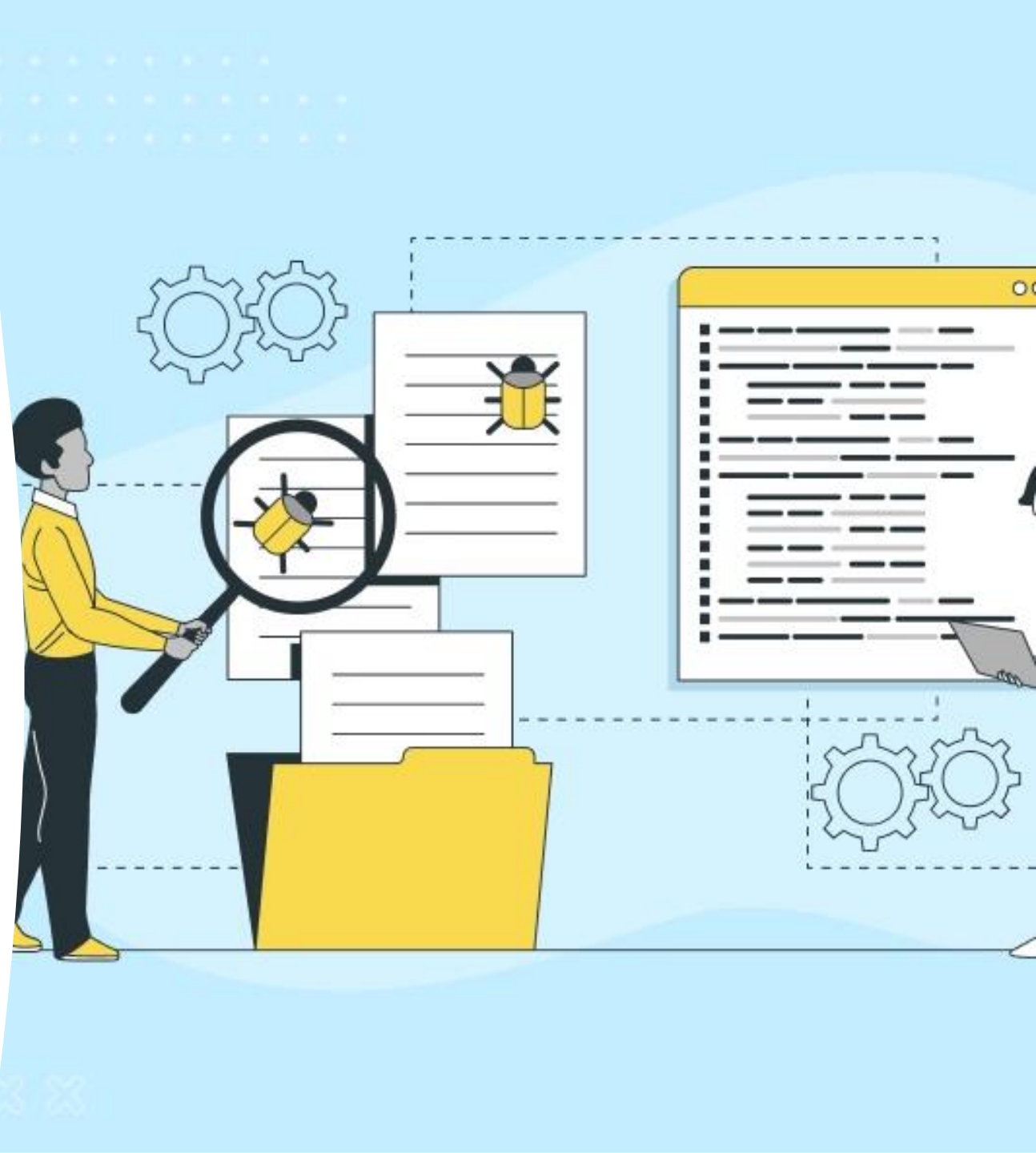
# V-model software testing lifecycle



# Voorbeelden

Bespreek de **testtypes** bij deze cases:

- Een bank lanceert een nieuwe mobile banking app
- Social media platform: redesign van user interface
- Software bedrijf: web browser extension
- E-commerce platform breidt uit naar een nieuwe internationale markt
- Online streaming service verwacht een piek in het aantal gebruikers tijdens een popular live event

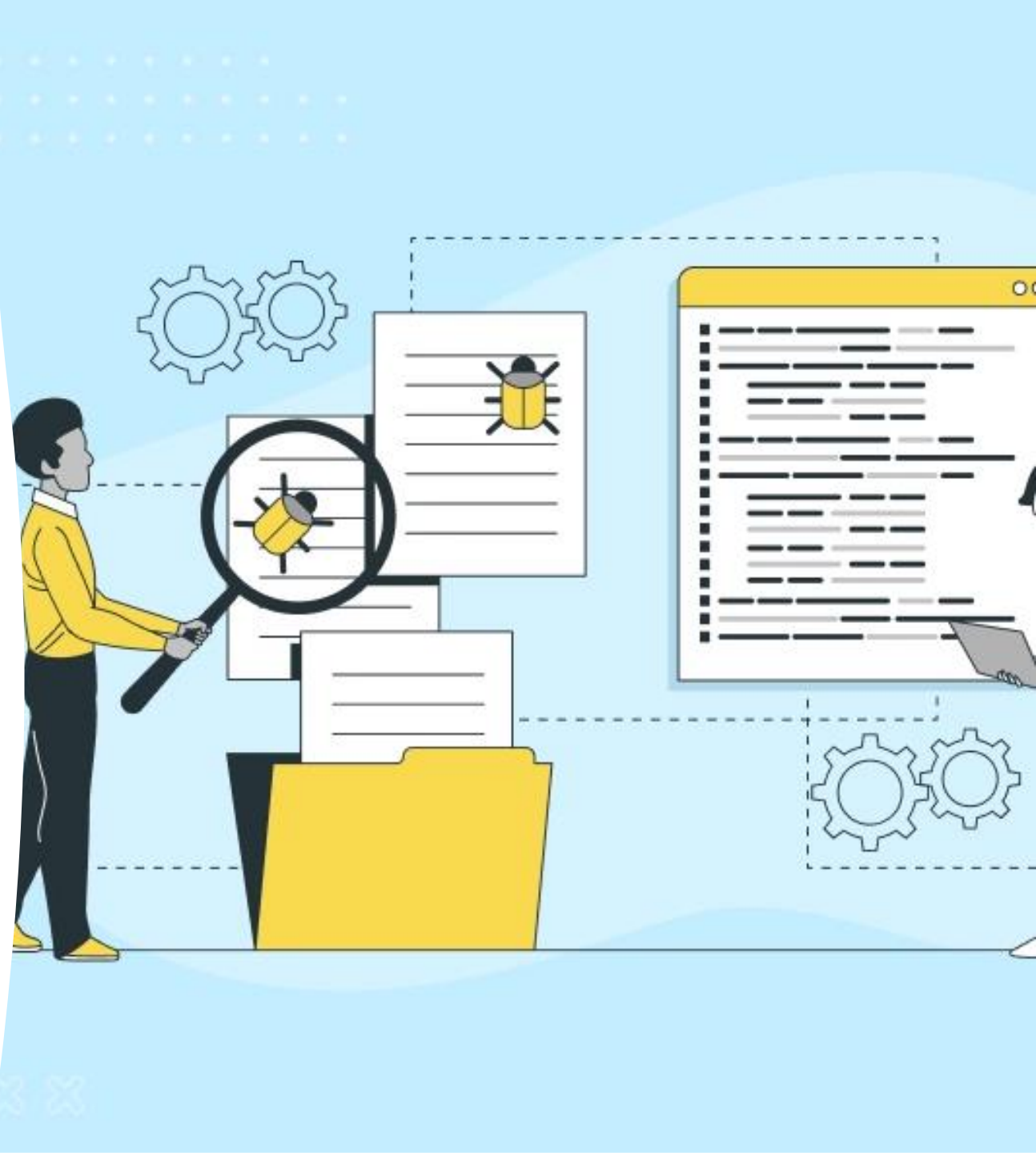




# Voorbeelden

Bespreek de **testlevels** bij deze cases:

- Software development team maakt een math library voor wetenschappelijke toepassingen
- E-commerce website integreert een nieuw betalingsmiddel.
- Een vliegtuig reservatiesysteem wordt getest, het maakt beheer van boekingen van vluchten mogelijk (boeking, annulatie, toekennen van zetels in het vliegtuig, ...)
- Een mobile app voor een fitness tracker
- Een bedrijf zal een aangepaste versie releasen van de software waarin de offertes van klanten beheerd worden





roles and responsibilities

# Rollen en verantwoordelijkheden

- Ontwikkelaars

- Unit- en integratietesten
- Verbeteren van software bij bugs



- Testers

- Integratie- en systeemtesten op basis van voorafbepaalde scenario's
- Documenteren van de resultaten
- Objectieve manier van testen



- Eind- en sleutelgebruikers

- Acceptatie van de oplossing d.m.v. acceptatietesten
- Testen functionaliteiten zoals ze die ook dagelijks zouden gebruiken



# Rollen en verantwoordelijkheden

- Testmanager/Projectmanager

- Coördineren testactiviteiten
- Opvolgen resultaten
- Opstellen testplan en resources selecteren
- Opstellen en valideren testscenario's



<https://www.youtube.com/watch?v=huv8zOGZ85E>



# Deliverables: Testplan

- Bevat een gedetailleerde beschrijving van de **teststrategie** en de uit te voeren **testscenario's**
- Wordt in principe al deels opgesteld tijdens de analyse
- Resources moeten vrijgemaakt worden
- Verschillende onderdelen:
  1. Testscenario's
  2. Testdata
  3. Verantwoordelijkheden en planning
  4. Testtools
  5. Afspraken rond acceptatie

# Onderdelen testplan



## 1. Testscenario's

- Beschrijving van de te testen scenario's
- Op basis van de business processen
- Zowel functionele als technische scenario's
- Een testscenario bevat de verschillende stappen die moeten worden doorlopen, de inputdata en het verwachte resultaat
- Ook beschrijven wat er niet zal getest worden

## 2. Testdata

- Voldoende en relevante testdata nodig
- Vaak geanonimiseerde operationele data, maar soms ook aangemaakte data

## 3. Verantwoordelijkheden en planning

- Wie gaat de test uitvoeren, wanneer?
- Rollen van de users

## 4. Testtools

- Beschrijving van de tools die zullen gebruikt worden bij het testen
- Verschillende tools die testen kunnen automatiseren
- Ook tools om testen op te volgen en te documenteren

## 5. Afspraken rond acceptatie

- 100% foutloos is niet mogelijk
- Er moeten afspraken gemaakt worden wanneer een oplossing vrijgegeven kan worden





# Deliverables - Testresultaten

- **Testers** documenteren de resultaten zodat de testmanager deze kan opvolgen en de nodige acties kan coördineren
- **Bij negatief resultaat:**
  - Wanneer de fout een technische oorzaak heeft, zal de ontwikkelaar deze fout herstellen
  - Wanneer de fout functioneel is, kijkt men naar de beslissing die werd genomen in de blueprint
    - Indien het anders is gebeurd dan in de blueprint werd omschreven, wordt de fout hersteld
    - Indien de functionaliteit wel werkt zoals voorzien, zal er een change request procedure moeten worden opgestart
  - Oplossingen moeten ook gedocumenteerd worden

# Afhankelijkheden

- De **resultaten** van de testfase hebben een **grote impact op het project**
  - Wanneer er teveel fouten zijn, kan het project worden stopgezet of go-live worden uitgesteld
  - In de projecten vandaag wordt er veel vroeger getest
- **Kosten van fouten** liggen vaak veel **hoger dan de kosten van de testen**
  - Wanneer eindgebruikers merken dat het systeem niet werkt, zal het vertrouwen dalen en de weerstand toenemen
  - Zeker wanneer de applicatie extern gebruikt wordt
- **Testfase is nauw verbonden met de ontwerpfase**
  - Testscenario's worden gebaseerd op de processen
  - Een deel van het testplan wordt al opgesteld bij het ontwerp



# Afhankelijkheden

- Testen is niet altijd beperkt tot de testfase
  - Ook tijdens de bouwfase en de voorbereidende fases wordt er al één en ander getest
  - Ook na go-live zal er nog getest moeten worden bij wijzigingen of eventuele bug-fixes



# Aandachtspunten

- Testen is een discipline op zich
  - Vereist nauwkeurigheid en het kunnen opvolgen van een scenario
  - Belangrijk om de juiste mensen te selecteren om te testen
  - Combinatie van externe en interne testers
- Testen is een repetitieve taak
  - Voor veel testen kan men testtools gebruiken
  - Minder testers nodig dus lagere kosten?
  - Initiële opzet en investering is nodig maar de tools kunnen nadien herbruikt worden
  - Wanneer men werkt met specifieke testbedrijven zijn deze tools inbegrepen



# Aandachtspunten

- **Voldoende tijd voorzien voor testen**
  - Vooral onverwachte bugs kosten veel tijd
  - Initiële scenario's inschatten is vrij eenvoudig, het aantal fouten inschatten niet
  - Uitgaan van een pessimistisch scenario
- **Niet elk testtype of testlevel** zal voorkomen op elk **project**
  - Bij een hoog niveau van standaardisatie zullen er minder/andere testen uitgevoerd worden

# Tools voor testing

- Manuele of geautomatiseerde testen
- Voorbeeld toepassing van testen [Testing XRay Jira](#)



# Vragen?

