

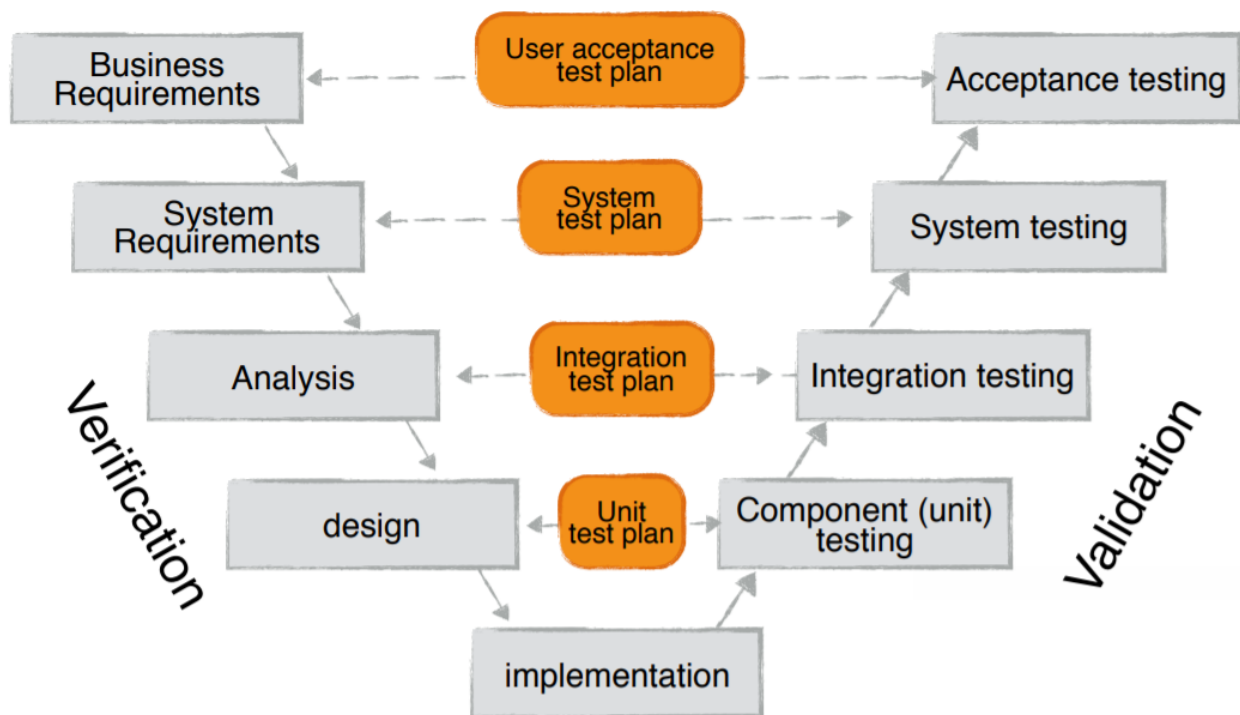
# TESTING

## HERFST SEMESTER 2018

LECTOR: ELKE STEEGMANS

---

### V-Model



### Verificatie fases

Business Requirements: De behoeften van het systeem achterhalen a.d.h.v. de behoeftes van de klant.

System Design: Vereisten worden vervolgens doorgespeeld naar de ontwikkelaars. Deze maken dan een systeemspecificatie. Hierin verandert de focus van wat het systeem moet kunnen naar de manier waarop men dit wil bereiken, dus welke methode, software enz. nodig is om aan de vereisten te kunnen voldoen.

---

Architecture Analysis (Systeem Ontwerp): Hierin worden de verschillende componenten die het systeem zal bevatten om aan de requirements te kunnen voldoen aan elkaar gekoppeld en wordt de relatie tussen de componenten beschreven.

Component Design: Het ontworpen systeem wordt in componenten opgedeeld. Iedere component heeft ook een eigen design, waar in detail wordt beschreven hoe het zijn processen zal verwerken.

## **Validatie fases**

Component/Unit Testing: Unit Test Plans worden ontwikkeld tijdens de component design fase. Bij unit testing worden de individuele componenten getest: voldoen zij aan de voorwaarden beschreven in component design?

Integration Testing: Wanneer componenten afzonderlijk zijn getest begint de integratietest, dit is in feite kijken of de componenten wanneer zij aan elkaar gekoppeld zijn ook nog naar wens functioneren.

System Testing: Hierin wordt gekeken of alle vereisten kunnen worden gehaald. Dit is niet het testen van verschillende componenten, maar meer het volledige systeem.

Acceptance Testing: De acceptatietest test feitelijk de business case: levert het systeem wat de klant ervan verwacht om voor hem een voordeel op te leveren in zijn werkveld? Met andere woorden zijn alle vereisten bereikt. Deze test moet uiteraard worden gedaan door de klant / gebruiker.

## **Voordelen van het V-model**

- Elke fase van integratie wordt getest.

## **Nadelen van het V-model**

- Het V-model gaat ervan uit dat vereisten niet veranderen.
- Het ontwerp wordt niet geverifieerd.
- Vereisten worden niet geverifieerd.
- In elke fase bestaat de kans op fouten.

---

# Test Design Techniques

## Black-box testing

Black-box testing neemt een extern perspectief van de testobjecten om testcases van af te leiden. Deze tests kunnen functioneel of niet functioneel zijn. De test-ontwerper selecteert goede en foute invoer en bepaalt de goede uitvoer. Er is geen kennis van de interne structuur van de te testen objecten. Bij de black box test is niets (of maar een klein deel) van de werking bekend bij de tester.

De black box test wordt meestal na de white box test uitgevoerd door mensen die ook niet ontwikkeld hebben en niets of weinig (hoeven te) weten van de interne structuur.

## Equivalence partitioning

Equivalentie partitioneren: Een gedeelte van een invoer- of uitvoerdomein waarvan op basis van de specificatie wordt aangenomen dat het gedrag van een component of systeem hetzelfde is.

Techniek: Waarmee testgevallen worden ontworpen om representanten van elke equivalentie partitie uit te voeren. In principe zijn de testgevallen ontworpen om elke partitie minstens een keer af te dekken.

Bvb:

- < 0
- 0 - 9
- 10 - 13
- 14 - 20
- > 20

## Boundary value analysis

Bij een grenswaardenanalyse wordt gekeken of het systeemgedrag verandert zodra een waarde van een parameter een bepaalde grens overschrijdt, terwijl bij equivalentieklassen meer wordt gekeken naar het systeemgedrag bij een voorkomen van een vastgesteld waardebereik.

De normale dekking betreft drie testsituaties: op de grens, direct boven en direct onder. Wat verstaan wordt onder direct is afhankelijk van de situatie. In geval van leeftijd in jaren (bijvoorbeeld <18) gaat het om 17, 18, 19 etc., terwijl in geval van bedragen met twee cijfers achter de komma het gaat om bedrag, bedrag + 0,01 en bedrag - 0,01.

Partition	Boundari
< 0	..., -1
0 - 9	0, 9
10 - 13	10, 13
14 - 20	14, 20
> 20	21, ...

## Decision table testing

Basically:

Condition				
<10	Y	Y	N	N
Cheated	Y	N	Y	N
Outcome				
Pass	N	N	N	Y

		Rules							
<b>Conditions</b>	Printer prints	No	No	No	No	Yes	Yes	Yes	Yes
	A red light is flashing	Yes	Yes	No	No	Yes	Yes	No	No
	Printer is recognized by computer	No	Yes	No	Yes	No	Yes	No	Yes
<b>Actions</b>	Check the power cable			✓					—
	Check the printer-computer cable	✓		✓					—
	Ensure printer software is installed	✓		✓		✓		✓	—
	Check/replace ink	✓	✓				✓		—
	Check for paper jam		✓		✓				—

---

## Functional testing

Testen gebaseerd op een analyse van de functionele specificaties van een component of systeem.

## Non-functional testing

Het testen van een component of systeem op niet functionele kwaliteitsattributen, zoals betrouwbaarheid, efficiëntie, bruikbaarheid, onderhoudbaarheid en overdraagbaarheid.

## State transition testing

Een functionele test specificatie techniek waarmee de testgevallen ontworpen worden om de geldige en ongeldige toestandsovergangen uit te voeren

	Event 1 (Insert card)	Event 2 (Enter Pin)	Event 3 (Pin OK)	Event 4 (Pin not OK)
S1:Start	S2	-	-	-
S2:Wait for Pin	-	S3	-	-
S3: 1st try	-	-	S6	S4
S4: 2nd Try	-	-	S6	S5
S5: 3rd Try	-	-	S6	S7
S6: access to account	-	-	-	-
S7: eat card	S1	-	-	-

## Use case testing

Een testspecificatietechniek waarin de testgevallen worden ontworpen om gebruikers scenario's uit te voeren.

---

## Seven Testing Principles

1. **Testing shows presence of defects:** Testing kan niet bewijzen dat code correct is (dat er geen defects zijn). Het vermindert wel de kans.
2. **Exhaustive testing is impossible:** Je kan niet alles testen. Ipv dit te proberen zou je risicoanalyse en prioriteiten moeten gebruiken om betere tests te schrijven.
3. **Early testing:** Zo vroeg mogelijk met test activiteiten beginnen.
4. **Defect clustering:** Vaak bevat maar een klein aantal van de modules het grootste deel van de fouten
5. **Pesticide paradox:** Als dezelfde tests opnieuw en opnieuw herhaalt worden zal deze set van test cases geen nieuwe fouten meer vinden. Om deze "pesticide paradox" te overkomen moet je huidige test cases herbekijken of herschrijven en nieuwe of andere tests schrijven.
6. **Testing is context dependent:** Je kan niet elke programma/systeem op dezelfde manier testen. (webshop vs vliegtuig-software)
7. **Absence of errors fallacy:** Fouten vinden en fixen gaat niet helpen als het systeem niet bruikbaar is en niet doet wat de gebruiker verwacht.

## Static vs dynamic testing

### Static testing

Statisch testen wordt uitgevoerd zonder het systeem daadwerkelijk uit te voeren. Zoals proofreading, code reviews, code syntax en data flow worden static getest door de compiler/IDE.

### Dynamic testing

Bij het dynamisch testen wordt het systeem uitgevoerd. Er kan een onderverdeling worden gemaakt naar impliciet testen of expliciet testen en naar de mate hoeveel de tester weet van de interne structuur van de software; black box test of white box test.

---

## **White-box testing**

De white box test is een teststrategie die gebruikmaakt van kennis van de interne structuur of code van een programma of systeem. Het is daarmee heel anders dan de black box test, die geen kennis heeft van de implementatie. Bij een white box test mag de implementatie gemodificeerd worden om effectief te testen. Een bekend voorbeeld hiervan is het compileren van de code met debug statements (of macro's).

Het primaire doel is vaak niet het controleren op fouten (verificatie), maar het opsporen van fouten (debuggen). Een bekend voorbeeld hiervan is de unittest, vaak in combinatie met een Test-Driven Development-aanpak.

## **Smoke test**

Een deel van alle beschreven / geplande testgevallen die de belangrijkste functionaliteiten van een component of systeem afdekken, om zeker te stellen dat de meest kritische functies van een programma werken, zonder verdere details in beschouwing te nemen. Een dagelijkse opleveringstest en een proeftest behoren tot de best practices uit de industrie.

## **Regression testing**

Het testen van een eerder getest programma na een wijziging, om aan te tonen dat er geen fouten (defecten) zijn geïntroduceerd of ontdekt in ongewijzigde gebieden van de software als gevolg van die wijzigingen. Het vindt plaats wanneer de software of de omgeving is gewijzigd.

[Standaard verklarende woordenlijst van software testtermen Vertaling Engels - Nederlands.pdf](#)