



Odisee
DE CO-HOGESCHOOL

Testlandschap – soorten testen



Jens Baetens

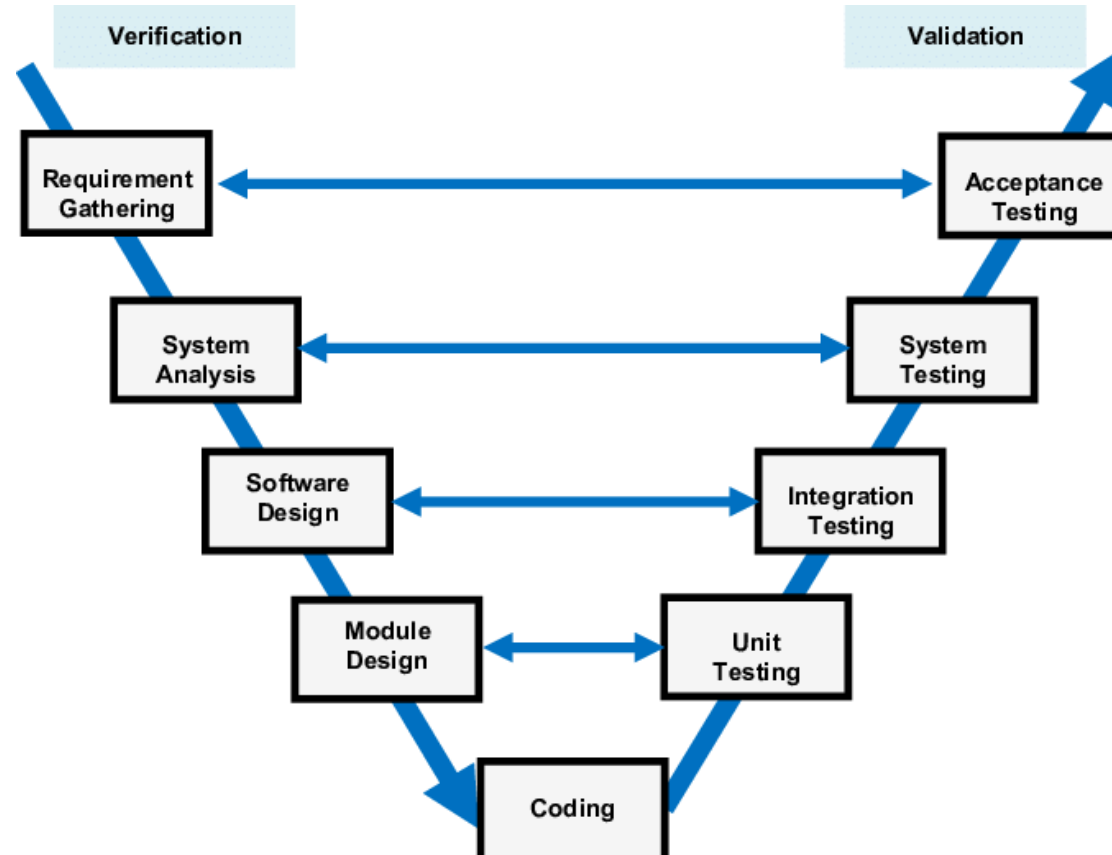
Soorten testen

- ▣ Er zijn meer dan 100 soorten test technieken (<https://www.guru99.com/types-of-software-testing.html>)
- ▣ Niet elk type test moet uitgevoerd worden
 - Hangt van scope en type project
- ▣ 3 types
 - Functional testing
 - Non-functional testing
 - Maintenance testing

Test levels

▣ Naast de 3 types zijn er ook 4 testlevels

- ▣ Unit test
- ▣ Integration test
- ▣ System test
- ▣ Acceptance





Testlevels

- ▣ Een testlevel is een groep van testactiviteiten die samen gepland en beheerd worden
- ▣ Elk level test op een ander niveau
 - Van individuele klassen/componenten tot volledige systemen

Unit testing

- ▣ Component testing
- ▣ Focus op componenten die afzonderlijk getest kunnen worden

- ▣ Doel

- ▣ Risico's verkleinen
- ▣ Vertrouwen opbouwen in de kwaliteit van de component
- ▣ Voorkomen dat defecten doorsijpelen naar hoger niveau

↳ *dependency injection*

↳ *sneller om fouten vroeg te detecteren*

Unit testing

▣ Wat wordt getest?

- Code en datastructuur
- Classen/methoden
- Databasemodules

▣ Types defecten/bugs gedetecteerd

- Logische fouten → *code fouten*
- Verkeerde functionaliteit → *acceptance requirements*

Integration testing

■ Focus op interactie tussen verschillende componenten of systemen

■ Doel

- Risico's verkleinen
- Vertrouwen opbouwen in de kwaliteit van de component
- Voorkomen dat defecten doorsijpelen naar hoger niveau

— → ook in
↑ Unit Testing

■ Kan toegepast worden op twee niveau's

- Component integration testing: interactie en interface tussen componenten
- System integration testing: interactie en interface tussen systemen

of klassen

Integration testing

▣ Wat wordt getest?

- ▬ Infrastructuur
- ▬ Interfaces
- ▬ API's
- ▬ Databases

▣ Gedetecteerde defecten/bugs

- ▬ Verkeerde/ontbrekende gegevens
- ▬ Verkeerde volgorde/timing van interface calls
- ▬ Interface mismatch
- ▬ Problemen bij communicatie tussen componenten

System testing

- ▣ Focus op het gedrag van een volledig system
- ▣ Doel
 - ▬ Nagaan of het systeem voldoen aan (non-)functionele vereisten
 - ▬ Valideren of het system volledig is en werkt zoals verwacht
- ▣ Wat wordt getest?
 - ▬ Applicaties
 - ▬ Software/hardware systemin
 - ▬ Operating systems
 - ▬ Systeemconfiguratie

System testing

- ▣ Opgespoorde defecten/bugs
 - Foutief/onverwacht gedrag van het system
 - Systeem werkt niet in een omgeving
 - Systeem werkt niet zoals beschreven in handleiding

Acceptance testing

- ▣ Focus typisch op gedrag van volledig system of product
- ▣ Nagaan of system klaar is en gebruikt kan worden
- ▣ Vier vormen
 - User acceptance tests: Kunnen eindgebruikers het gebruiken om hun doel te bereiken
 - Operational Acceptance Testing: Vertrouwen opbouwen dat administrators het werkende kunnen houden
 - Contractueel: Voldoet software aan criteria in het contract
 - Alpha en Beta Testing: Alpha testing door development team, beta door eindgebruikers

Test types

- ▣ Test types is een groep van testactiviteiten die gericht zijn op het testen van specifieke eigenschappen van het system
- ▣ Test types in deze cursus
 - Functionele testen
 - Niet-functionele testen
 - Change-related testing

Functionele testen

- ▣ Evalueren van de functionaliteiten van het system

- ▬ Wat het moet doen
- ▬ Bais: functionele vereisten / business requirements

- ▣ Uitgevoerd op alle levels

→ unit testing → ook requirements checken

- ▣ Vergt vaak kennis van het domein waarin de software werkt

- ▬ Bvb Boekhoudingsplatform vereist kennis van boekhoudingsregels, best practices, ...

Niet-functionele testen

- ▣ Valideert niet-functionele vereisten
 - Gebruiksvriendelijkheid, performance, security, ...
- ▣ Hoe goed presteert het systeem?
- ▣ Zo snel mogelijk in het development process
- ▣ Op alle testlevels noodzakelijk

Change-related testing

- ▣ Testen gerelateerd aan wijzigingen in het system
- ▣ Twee vormen
 - ▬ Confirmation testing
 - Na een fix tests schrijven die bevestigen dat het probleem opgelost is
 - ▬ Regression testing
 - Testen die nagaat of na een wijziging de bestaande code nog werkt

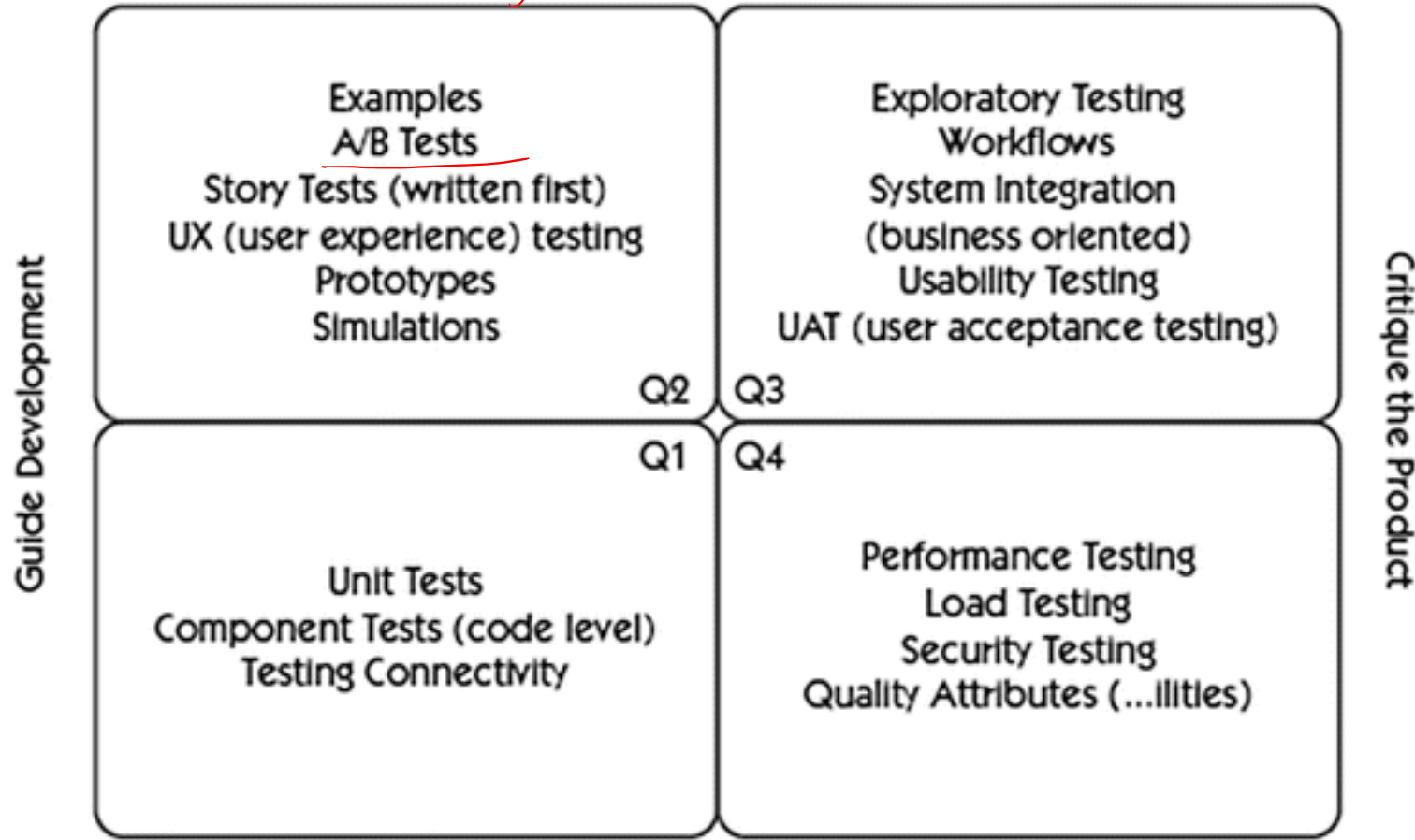
Levels vs Types

		Type →			
Level ↓		Functional	Nonfunctional		Regression
			Performance	Security	
	Unit	✓	×	✓	×
	Integration	×	×	×	✓
	System	×	×	×	×
	Acceptance	×	×	×	×

Maintenance Change-related

Agile testing quadrants

Subjectief Business Facing *Blond ↑ gericht*



→ achteraf

In het begin
Application architecture evalueren

Gelijektig Technology Facing *Coole gericht*

Andere test types

▣ Smoke testing

- ▬ Triviale zaken testen (enkel meest belangrijke)

▣ End-to-end testing

- ▬ Meerdere systemen samen testen

▣ Characterization testing

- ▬ Bij legacy projecten

→ unit testen voor bestaande code
→ Wat doet deze functie

▣ Positive & negative testing

- ▬ Testen met correcte data en werking verifiëren
- ▬ Testen met incorrecte data om te kijken of de fouten gedetecteerd worden



Andere test types

■ Small, medium & large tests

- Systeem van Google

- Small test = unit test

- Klein en snel, geen externe systemen, binnen 60 seconden gedaan

omdat er veel data is

- Medium test

- Interactie testen tussen verschillende lagen, binnen 5 minuten gedaan

- Large test

- Geen beperkingen