

# Software Quality 1

## Workshop 4

**WFSDAD.M018.22**

**Jan Zuur**

**René Prinz**



# Studiehandleiding

## Leerdoelen workshop 4

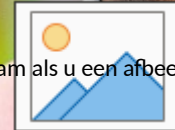
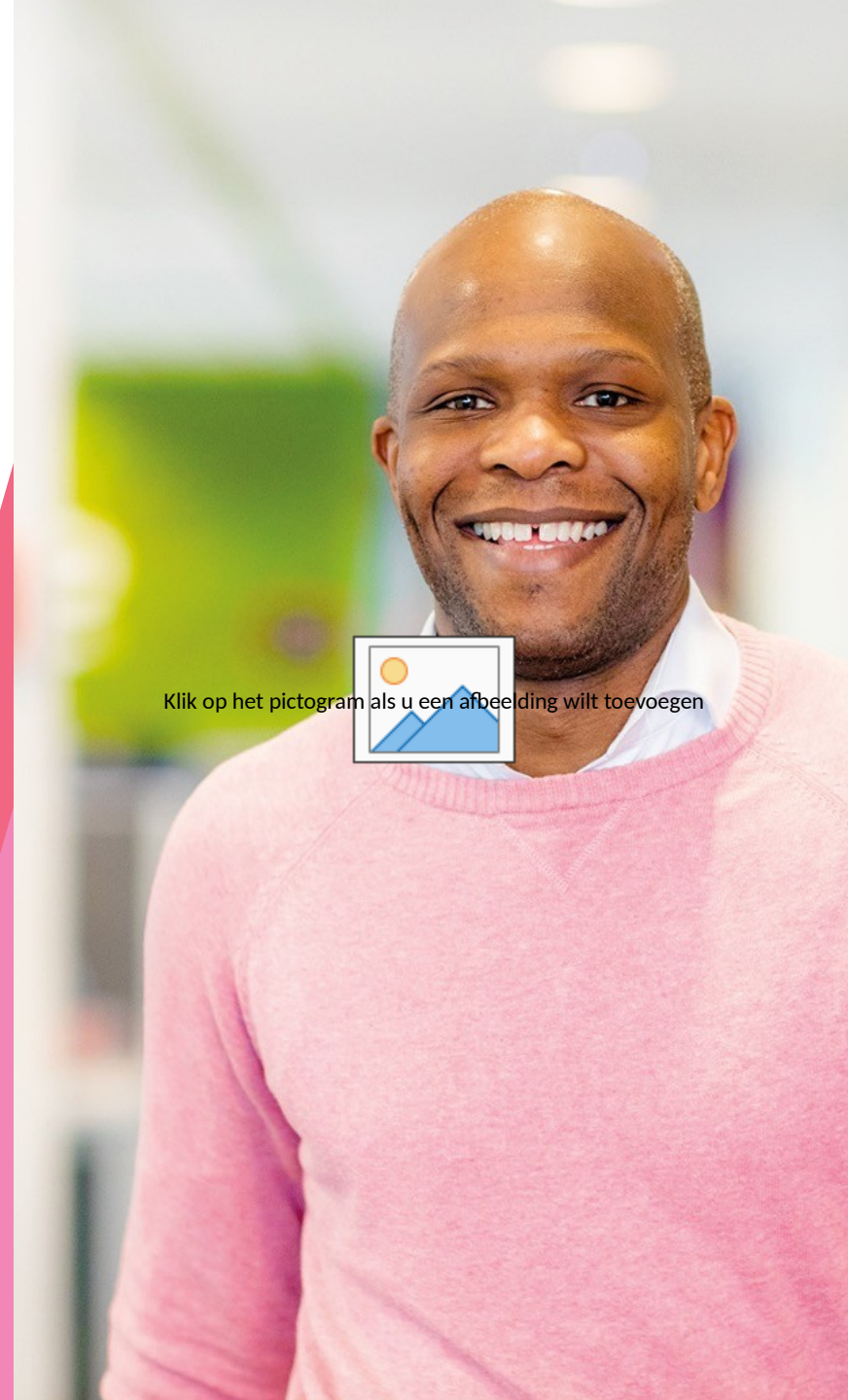
- Je weet wat er bedoeld word met Non Functional requirements
- Je weet wat er bedoeld word met Functional requirements
- Je kunt de verschillen tussen de 2 soorten requirements benoemen en voorbeelden van geven
- Je weet hoe je beide soorten requirements kan testen





# Agenda

1. Functional requirements
2. Non Functional requirements
3. Zoek de verschillen



Klik op het pictogram als u een afbeelding wilt toevoegen

Studiehandleiding

# Literatuur / naslagwerk

<https://www.stxnnext.com/blog/software-product-requirements-high-level-project-specifications-guide>

<https://www.stxnnext.com/hubfs/STX%20Next%202020/blog/lean-canvas-example.pdf>

<https://www.softwaretestinghelp.com/test-case-template-examples/>

<https://www.softwaretestinghelp.com/what-is-test-scenario/>

[What are non-functional requirements \(NFRs\) | Planio](#)



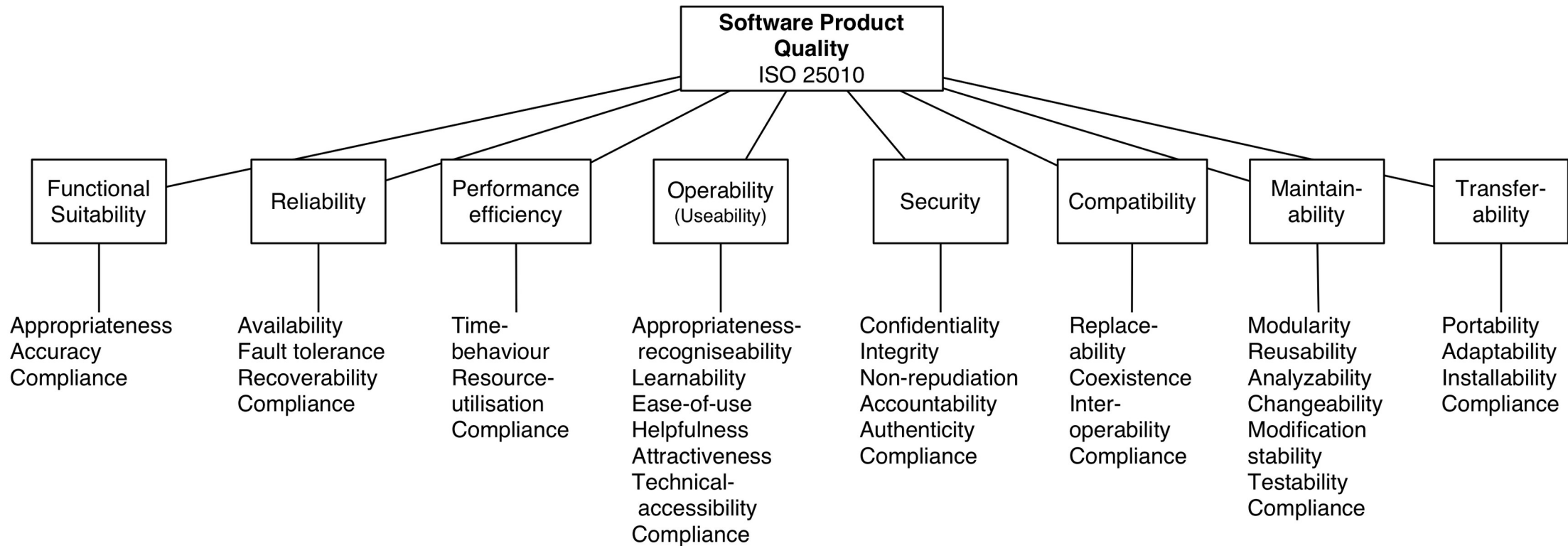
# Even terugblikken....



week ook al weer over?



# Kennen jullie deze nog?





# ***W Functional requirements***

# Voordat we de diepte ingaan...

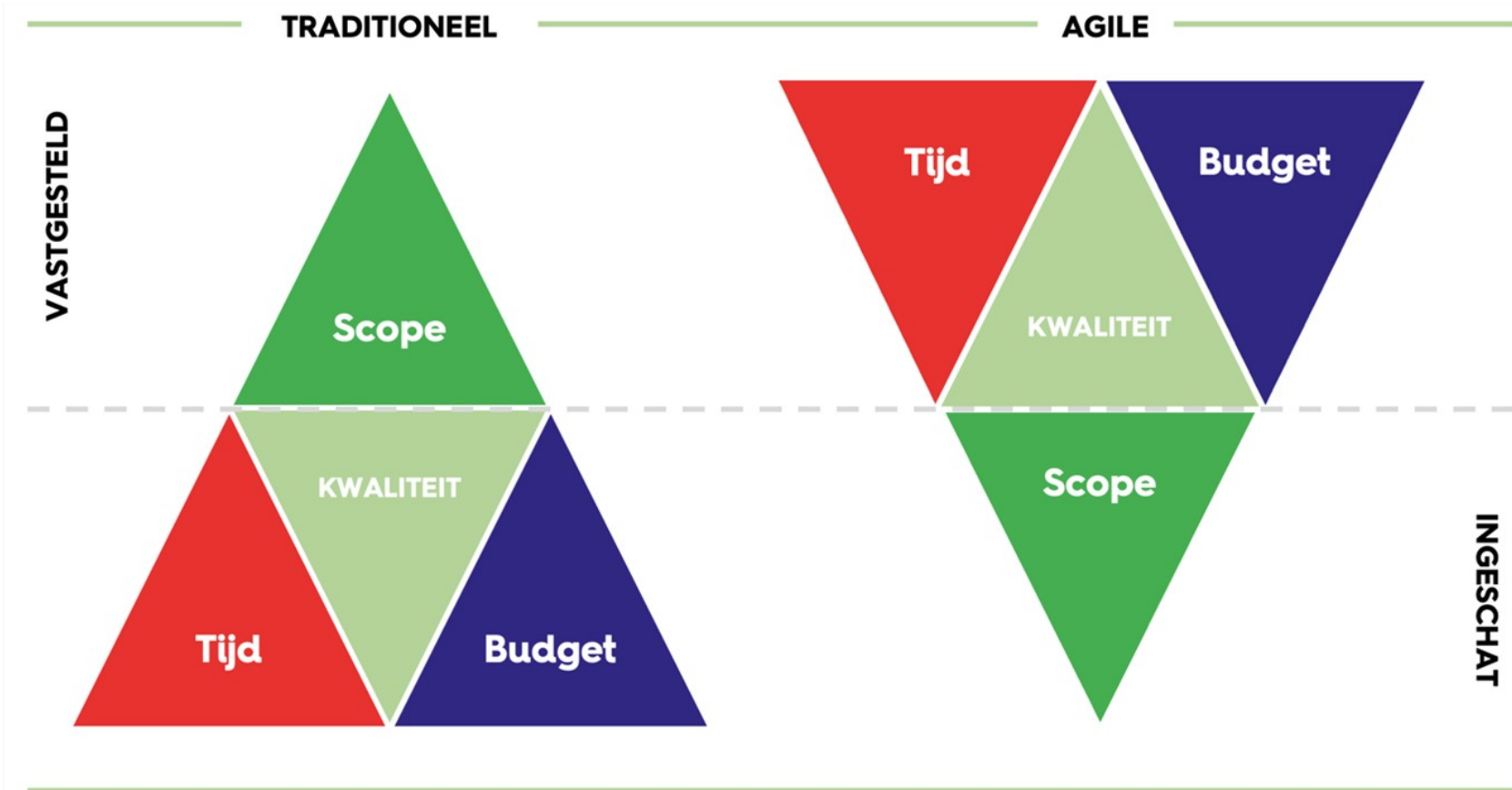
Functional  
requirements





Wie heeft er wel eens gehoord van....

## De duivelsdriehoek



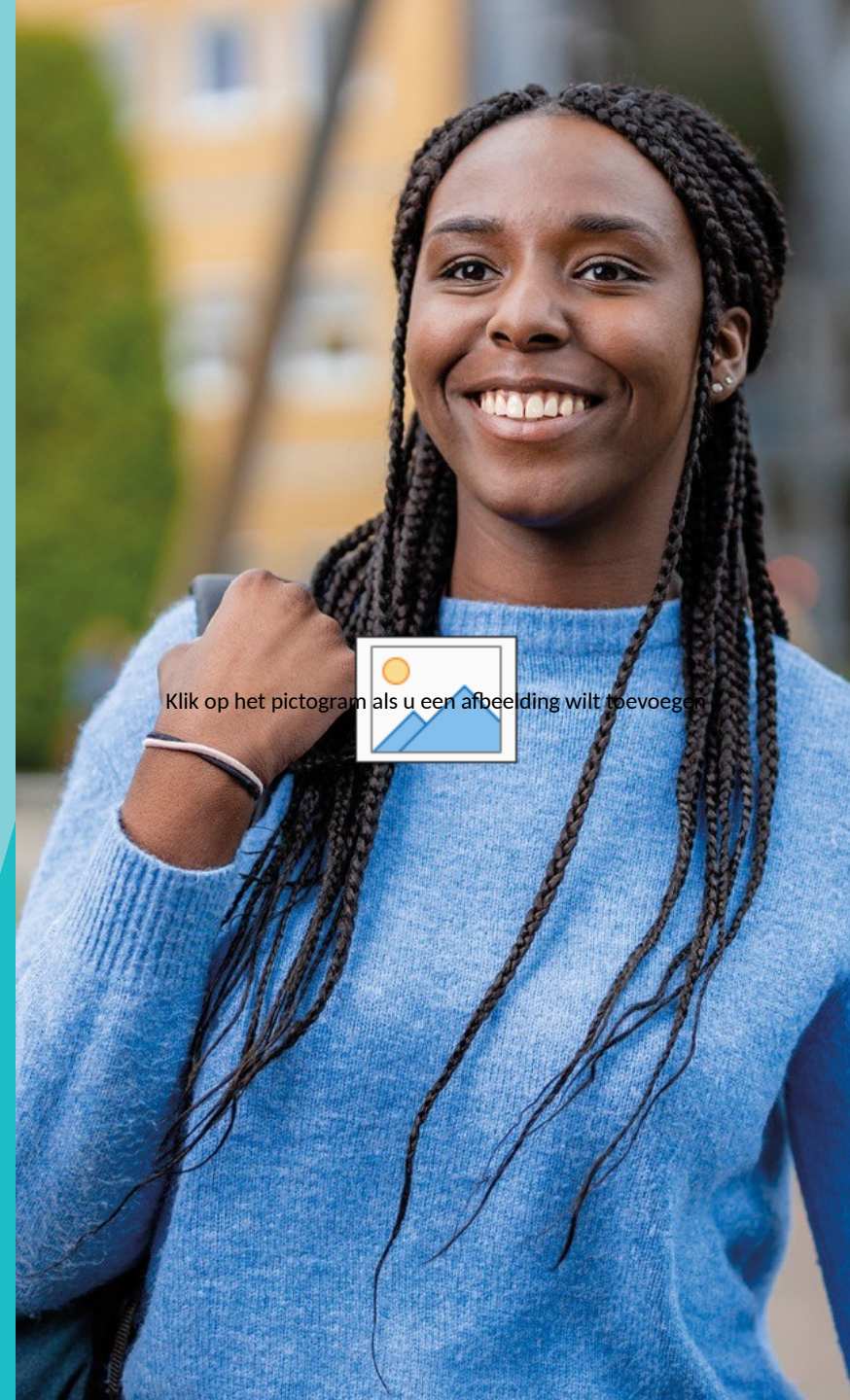
# Wat zijn functional requirements

Functional Requirements (FRs) zijn de basisvereisten die de functionaliteit van een softwareproduct definiëren. Ze beschrijven de taken die het systeem moet uitvoeren, de acties die het moet ondernemen en de reacties die het moet geven in verschillende situaties.



# Kenmerken

1. Uniek identificeerbaar: Een FR moet een uniek nummer of label hebben, zodat deze gemakkelijk traceerbaar en referentieerbaar is.
2. Beschrijft een specifieke functie: Elke FR moet specifiek beschrijven wat het systeem moet doen. Dit betekent dat het niet alleen een algemene beschrijving mag zijn, maar ook een gedetailleerde beschrijving van de functionaliteit die vereist is.
3. Meetbaar: FRs moeten meetbaar zijn, zodat het mogelijk is om vast te stellen of de functionaliteit correct werkt en aan de verwachtingen voldoet.
4. Realistisch: Een FR moet realistisch zijn en rekening houden met de beperkingen van het systeem, de technologie en het budget.
5. Traceerbaar: Het is belangrijk om de oorsprong en de geschiedenis van een FR te kunnen traceren om de besluitvorming rondom wijzigingen te ondersteunen.





**Laten we eens een poging samen doen om...**

- **Opdracht 1a:**

- Bedenk een aantal functional requirements voor jouw eigen business case en schrijf deze op

**Opdracht 1b:**

- Bedenk hoe je deze requirement zo slecht mogelijk zou kunnen uitwerken...



# Naslagwerk voor zelfstudie ter voorbereiding voor testen

## **Soorten testen functional requirements**

- 1 Acceptatietesten: deze testen worden uitgevoerd om te bepalen of de software voldoet aan de functionele vereisten die zijn vastgelegd in de gebruikersvereisten en specificaties.
- 2 Integratietesten: deze testen worden uitgevoerd om te bepalen of de software correct integreert met andere systemen en applicaties die in het ecosysteem van de software draaien.
- 3 Systeemtesten: deze testen worden uitgevoerd om te bepalen of de software correct functioneert als een geheel en of alle functionaliteiten werken zoals verwacht.
- 4 Unit testing: deze testen worden uitgevoerd op individuele onderdelen van de software om te bepalen of ze correct werken volgens de specificaties.
- 5 Functionele testen: deze testen worden uitgevoerd om te bepalen of de software voldoet aan de functionele vereisten en doet wat het zou moeten doen.
- 6 Black box testing: deze testen worden uitgevoerd zonder kennis van de interne werking van de software. Hierbij wordt de software getest door input te geven en te controleren of de output aan de verwachtingen voldoet.
- 7 Grey box testing: deze testen worden uitgevoerd met beperkte kennis van de interne werking van de software. Hierbij wordt de software getest door input te geven en te controleren of de output aan de verwachtingen voldoet, waarbij rekening wordt gehouden met enige kennis van de interne werking.
- 8 Regression testing: deze testen worden uitgevoerd om te bepalen of wijzigingen in de software geen onbedoelde gevolgen hebben voor bestaande functionaliteiten.



# **Pauze**





# Non Functional requirements

# Wat zijn Non Functional requirements

NFR's zijn softwarevereisten die niet direct gerelateerd zijn aan de functionaliteit van de software, maar eerder aan de prestatie, betrouwbaarheid, schaalbaarheid, veiligheid en andere kenmerken van de software. Voorbeelden van NFR's zijn responstijden, beschikbaarheid, schaalbaarheid, beveiliging, betrouwbaarheid en onderhoudbaarheid.



## Waarom zijn Non Functional Requirements belangrijk?

- Wie heeft er een idee?





Voorbeelden van NFR's zijn:

1. Prestatie-eisen zoals responstijd, verwerkingssnelheid en belastingstolerantie.
2. Veiligheidseisen zoals autorisatie, authenticatie, vertrouwelijkheid en integriteit van gegevens.
3. Betrouwbaarheidseisen zoals beschikbaarheid, hersteltijd, fouttolerantie en veerkracht.
4. Bruikbaarheidseisen zoals gebruiksvriendelijkheid, toegankelijkheid en compatibiliteit met verschillende apparaten en platforms.
5. Schaalbaarheidseisen zoals de mogelijkheid om meer gebruikers, gegevens of transacties te verwerken zonder prestatieverlies.



## Interne **stakeholders**

- Werknemers
- Managers
- Aandeelhouders
- Investeerders
- Ondernemingsraad

## Externe **stakeholders**

- Leveranciers
- Klanten
- Media
- Crediteuren
- Concurrenten
- Business experts

## Interface **stakeholders**

- Overheid (nationaal)
- Lokale overheid
- Maatschappij
- Consumentenbond
- Kamer van Koophandel
- Opleidingen & scholen

WE  
THINK  
NEXT



# Opdracht

## Opdracht 1a:

Probeer de non functional requirements voor jouw business case eens uit te schrijven

## Opdracht 1b:

Probeer bij elke Non functional requirement eens te bedenken welke stakeholder daar het meeste belang bij heeft





# Naslagwerk voor zelfstudie en voorbereiding

- NFR's kunnen worden geïdentificeerd door middel van een reeks technieken, zoals:
  1. Interviews met belanghebbenden: Door met belanghebbenden te spreken, kan men een beter beeld krijgen van de NFR's die belangrijk zijn voor hun specifieke toepassingen.
  2. Brainstormen: Brainstormen is een effectieve manier om de belangrijkste NFR's te identificeren.
  3. Vergelijking met andere software: Door de prestaties van andere software te analyseren, kan men bepalen welke NFR's belangrijk zijn.
  4. Analyse van gebruiksscenario's: Door gebruiksscenario's te analyseren, kan men bepalen welke NFR's belangrijk zijn voor de specifieke toepassingen.

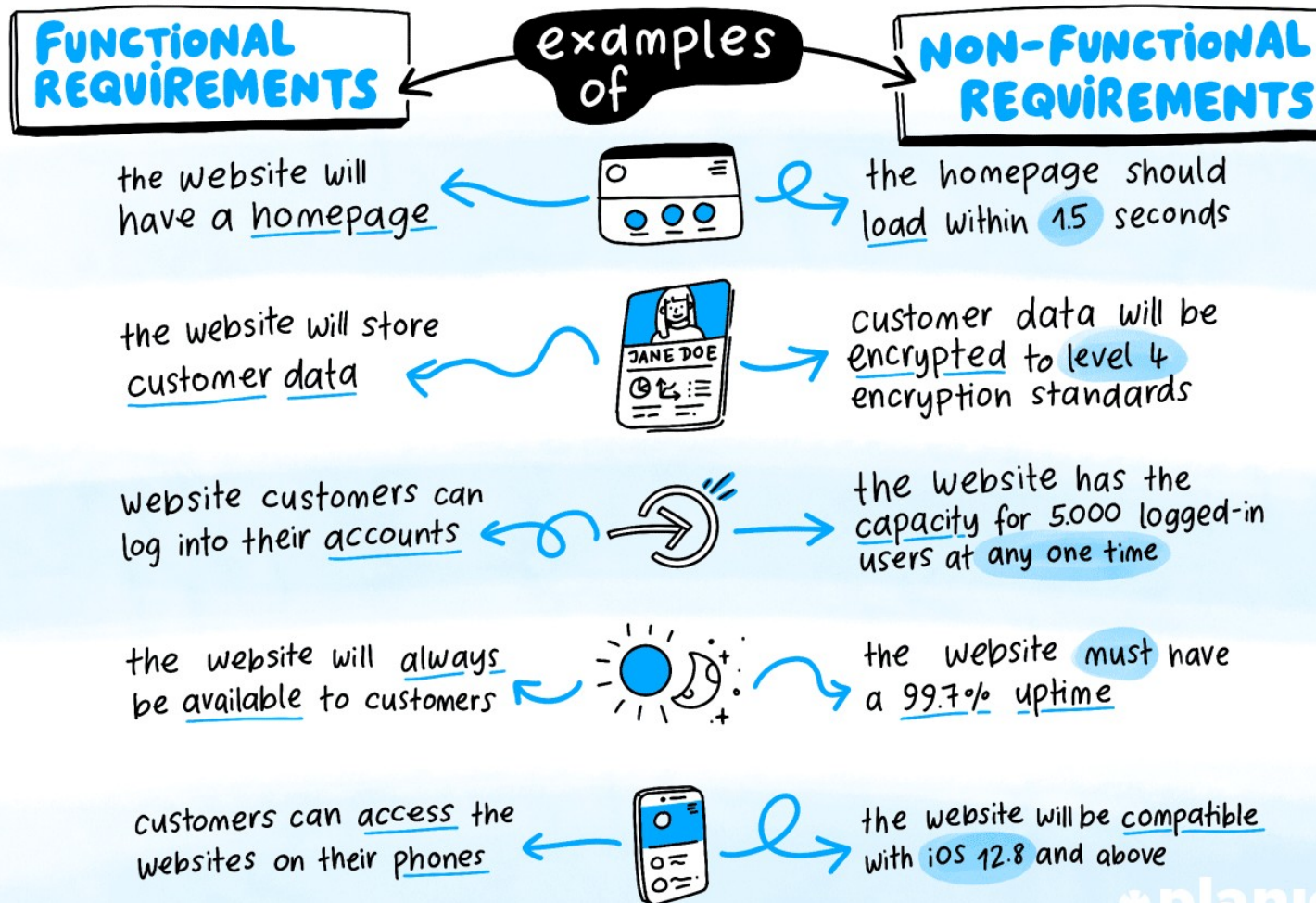


# Naslagwerk voor zelfstudie en voorbereiding

1. **Prestatietests:** NFRs die betrekking hebben op prestatie kunnen worden getest met behulp van prestatietests. Dit kan worden gedaan door het simuleren van een bepaald aantal gebruikers, transacties of gelijktijdige verzoeken om te zien hoe het systeem presteert onder stressvolle omstandigheden.
2. **Beveiligingstests:** NFRs die betrekking hebben op beveiliging kunnen worden getest met behulp van beveiligingstests. Dit omvat het testen van de software op kwetsbaarheden en het valideren van de beveiligingsfuncties, zoals toegangscontroles en encryptie.
3. **Usability tests:** NFRs die betrekking hebben op gebruiksvriendelijkheid kunnen worden getest met behulp van gebruikerstests. Dit omvat het testen van de software door de eindgebruiker om te bepalen hoe gemakkelijk de software is te gebruiken en hoe goed deze aansluit bij de gebruikersbehoeften.
4. **Compatibiliteitstests:** NFRs die betrekking hebben op compatibiliteit kunnen worden getest met behulp van compatibiliteitstests. Dit omvat het testen van de software op verschillende hardware- en softwareconfiguraties om te zien of de software correct werkt onder verschillende omstandigheden.
5. **Schaalbaarheidstests:** NFRs die betrekking hebben op schaalbaarheid kunnen worden getest met behulp van schaalbaarheidstests. Dit omvat het testen van de software om te zien hoe het zich gedraagt wanneer het wordt uitgebreid naar een grotere gebruikersbasis, grotere gegevenssets of meer transacties.
6. **Onderhoudbaarheidstests:** NFRs die betrekking hebben op onderhoudbaarheid kunnen worden getest door de software te evalueren op factoren zoals de structuur van de code, de documentatie en de gemakkelijheid van het maken van wijzigingen.



# Zoek de verschillen..



# Naslagwerk voor zelfstudie en voorbereiding

Hieronder volgt een overzicht van de belangrijkste verschillen tussen functional en non-functional requirements:

1. **Doel:** Functional requirements beschrijven wat het systeem moet doen en welke functionaliteit het moet bieden, terwijl non-functional requirements beschrijven hoe het systeem moet presteren en welke kwaliteiten het moet hebben.
2. **Meetbaarheid:** Functional requirements kunnen vaak gemakkelijker worden gemeten dan non-functional requirements, omdat ze zich richten op wat het systeem moet doen. Non-functional requirements zijn daarentegen vaak moeilijker te meten omdat ze zich richten op hoe het systeem moet presteren.
3. **Objectiviteit:** Functional requirements zijn vaak objectiever dan non-functional requirements omdat ze zich richten op wat het systeem moet doen en minder afhankelijk zijn van interpretatie. Non-functional requirements zijn vaak afhankelijker van interpretatie en subjectiever.
4. **Scope:** Functional requirements richten zich vaak op specifieke functies of onderdelen van het systeem, terwijl non-functional requirements betrekking hebben op het systeem als geheel en de kwaliteiten die het moet hebben.
5. **Prioriteit:** Functional requirements zijn vaak belangrijker dan non-functional requirements omdat ze direct bijdragen aan de functionaliteit en bruikbaarheid van het systeem. Non-functional requirements zijn echter ook belangrijk omdat ze de kwaliteit en prestaties van het systeem bepalen.
6. **Testbaarheid:** Functional requirements zijn vaak gemakkelijker te testen dan non-functional requirements omdat ze zich richten op de functionaliteit van het systeem. Non-functional requirements zijn vaak moeilijker te testen omdat ze betrekking hebben op de prestaties en kwaliteit van het systeem, wat vaak afhankelijk is van omgevingsfactoren en moeilijker te repliceren is in een testomgeving.
7. **Veranderlijkheid:** Functional requirements zijn vaak meer vatbaar voor veranderingen dan non-functional requirements, omdat ze vaak meer afhankelijk zijn van de specifieke behoeften en wensen van de gebruikers. Non-functional requirements zijn vaak minder vatbaar voor veranderingen omdat ze betrekking hebben op de kwaliteiten en prestaties van het systeem als geheel.





## Afsluiting

# Zijn de leerdoelen behandeld?

- Je weet wat er bedoeld word met Functional requirements
- Je weet wat er bedoeld word met Non Functional requirements
- Je kunt de verschillen tussen de 2 soorten requirements benoemen en voorbeelden van geven
- Je weet hoe je beide soorten requirements kan testen



Wat gaan wij volgende  
week doen?