

# CAFCR Functional View

## *Functional, non- Functional requirements & Constraints*

System Engineering (TCTI-V2SYEN-16)  
week 3

Marius Versteegen

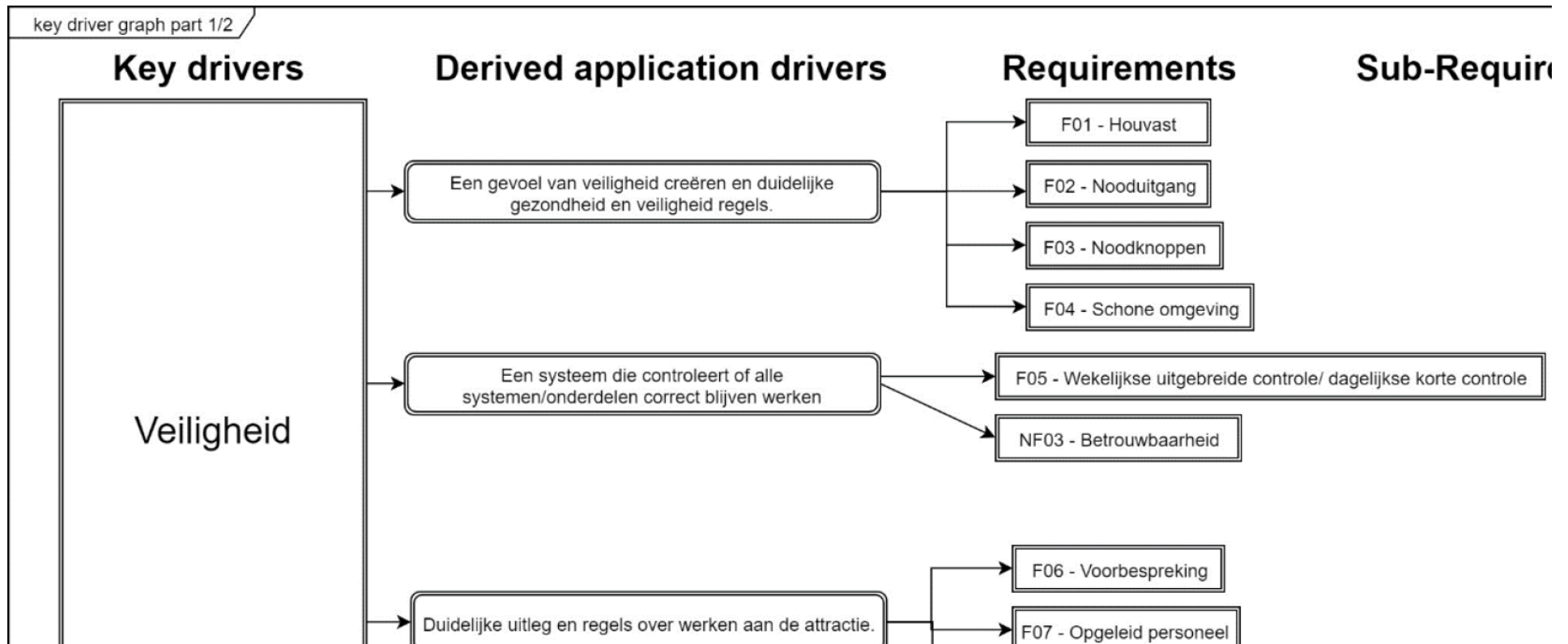
Auteur: Joost Schalken-Pinkster

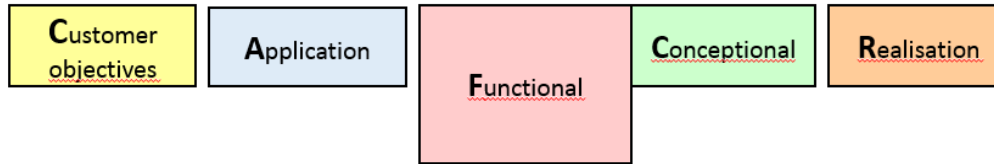


**Terugblik**

**2<sup>40</sup>**

# Key driver graph



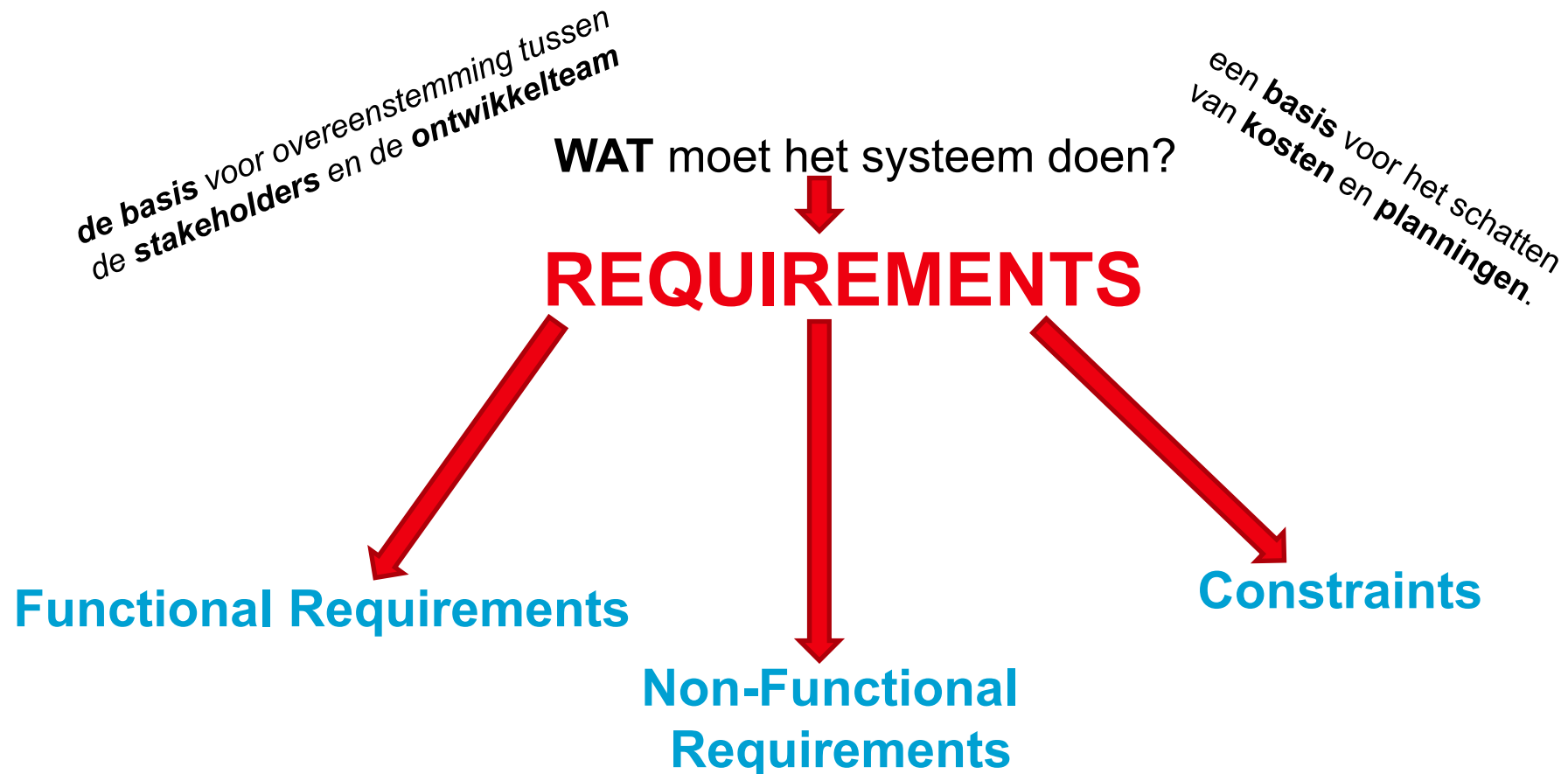


## Voordelen van goed geschreven requirement

- Vormt **de basis** voor overeenstemming tussen de **stakeholders** en de **ontwikkelteam** over wat het product moet doen.
- Vermindert de ontwikkelingsinspanning omdat er minder herstel nodig is vanwege slecht beschreven, ontbrekende en onbegrepen eisen.
- Biedt een **basis** voor het schatten van **kosten** en **planningen**.

Function	Behaviour	Structure
System Context	Functional Requirements	Logische view
Stakeholders	Non-functional requirements	Development view
Key drivers	Constraints	Beslissingsmatrices
Application drivers		FMEA
		Process View
		Physical View
Key-driver graph		
	Tracability diagram	

# Requirements



De requirements moeten **toepasbaar** zijn, **testbaar**, gerelateerd aan **identificeerbare bedrijfsbehoefte** of kansen, en gedefinieerd op een niveau dat voldoende gedetailleerd is voor het systeemontwerp.

# Inhoudsopgave



## Requirements opstellen

Functional Requirements

Non Functional Requirements

Constraints

Conclusies

# Requirements Attributen

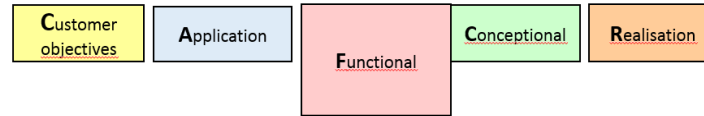
<b>Naam</b>	<i>&lt; Naam van de vereiste &gt;</i>
<b>Omschrijving</b>	<i>&lt;formulering <b>SMART</b>&gt;</i>
<b>ID</b>	<i>&lt;identificatie&gt; = &lt;type requirements&gt;&lt;nr&gt;</i>
<b>Rationale</b>	<i>&lt;Verklaring&gt;</i>
<b>Business prioriteit</b>	<i>&lt;MoSCoW&gt;</i>



# Een voorbeeld



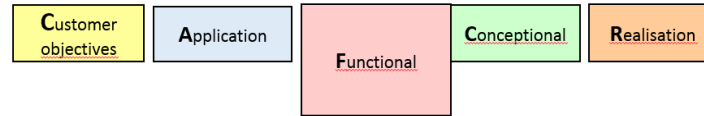
F01 - Houvast	
<b>Omschrijving</b>	Gebruikers moeten zich ergens aan vast kunnen houden.
<b>Rationale</b>	De gebruikers moeten zich vast kunnen houden, zodat zij niet vallen als ze hun evenwicht verliezen.
<b>Business Priority</b>	Must have



# SMART Requirements

Zijn de requirements ...

- **S**pecifiek *Is de requirement eenduidig?*
- **M**eetbaar *Onder welke meetbare voorwaarden of vorm is het te bereiken?*
- **A**ceptabel *Is het acceptabel voor de opdrachtgever/gebruiker?*
- **R**ealiseerbaar *Is het haalbaar?*
- **T**ijdgebonden *wanneer moet het bereikt worden?*

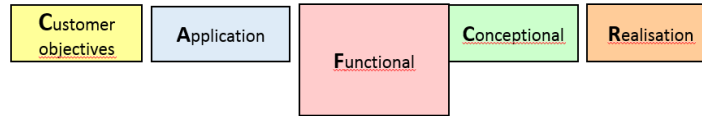


# Rationale

- Rationale is een verklaring van de redenering achter een beslissing.

Bijvoorbeeld:

Als een requirement een methode kiest, moet in de Rationale vermeld worden waarom die keuze is gemaakt.



# Business Prioriteit

## **M**ust Have → Essentieel

Deze vereisten moeten in het eindresultaat terugkomen, Het niet halen van deze vereisten betekent dat het systeem niet voldoet aan de key drivers en is niet bruikbaar.

## **S**hould Have → Belangrijk

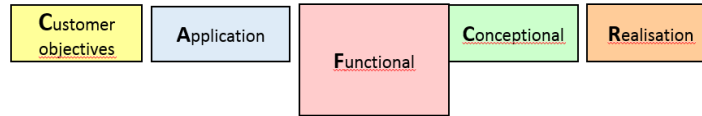
deze vereisten zijn zeer gewenst, maar zonder is het product wel bruikbaar. Het weglaten van deze vereisten kan klant tevredenheid beïnvloeden.

## **C**ould Have

Deze vereisten zullen alleen aan bod komen als er tijd genoeg is.

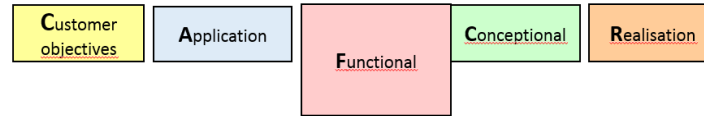
## **W**on't Have

deze eisen zullen in dit project niet aan bod komen maar kunnen in de toekomst, bij een vervolgproject, interessant zijn.



## Tips requirements opstellen

- ✓ Maak actieve zinnen
- ✓ Vermijd onduidelijke woorden
- ✓ Kwantificeer waar mogelijk (meetbaar)
- ✓ Geen onnodige keuzes voor implementatie of keuzes die alleen een sub-systeem aangaan.



# Tracability

- ✓ Iedere requirement heeft een eigenaar
- ✓ Requirements zijn achterwaarts traceerbaar naar application drivers en key drivers en voorwaarts traceerbaar naar ontwerp, code en testcases

# Inhoudsopgave

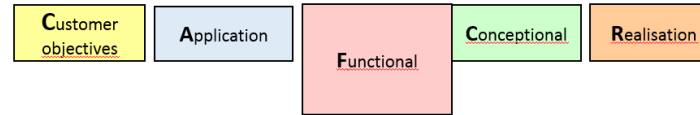
Requirements opstellen

Functional Requirements

Non Functional Requirements

Constraints

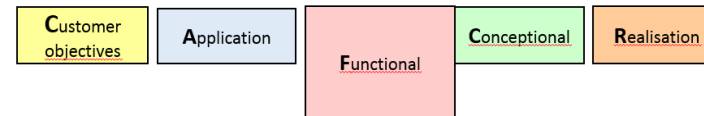
Conclusies



# Functional requirements

- Definiëren **WAT** het systeem voor de gebruikers moet doen.
- Zijn **acties** die het systeem moet kunnen uitvoeren.
- Dragen direct bij aan het bereiken van het doel van de klant.





# Functional Requirements

## Voorbeelden

- “Applicatie stuurt een email wanneer een gebruiker zicht aanmeldt”

OK

### F01 – Camerabeelden

<i>Omschrijving</i>	De robot moet actuele beelden terug kunnen geven aan de robotoperator.
<i>Rationale</i>	De operator moet actuele beelden ontvangen, zodat deze hierop acties kan uitvoeren.
<i>Business Priority</i>	Must have

### F03 – Rupsbandenbesturing

<i>Omschrijving</i>	De robot moet met zijn rupsbanden kunnen voortbewegen.
<i>Rationale</i>	Rupsbanden geven meer grip in moeilijk begaanbaar terrein dan andere alternatieven.
<i>Business Priority</i>	Must have

kan misschien beter..

# Opdracht Functional Requirements: Schrijf je Functional Requirements uit in blokken als onderstaand:



F01 – Camerabeelden	
<i>Omschrijving</i>	De robot moet actuele beelden terug kunnen geven aan de robotoperator.
<i>Rationale</i>	De operator moet actuele beelden ontvangen, zodat deze hierop acties kan uitvoeren.
<i>Business Priority</i>	Must have

- Definiëren **WAT** het systeem voor de gebruikers moet doen.
  - Zijn **acties** die het systeem moet kunnen uitvoeren.
  - Ze dragen direct bij aan het bereiken van het doel van de klant.
- ✓ Actieve zinnen
  - ✓ Kwantificeer waar mogelijk
  - ✓ Requirement bevat geen aanwijzingen over de implementatie

# Inhoudsopgave

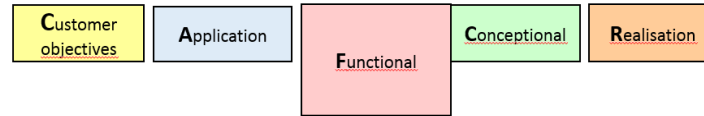
Requirements opstellen

Functional Requirements

Non Functional Requirements

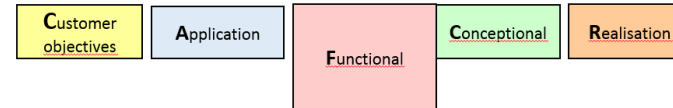
Constraints

Conclusies



# Non-Functional requirements

- Beschrijven **HOE** het systeem werkt en zich moet gedragen
- Specificeren de **kwaliteitsattributen** en kenmerken van het systeem (eigenschappen of kwaliteiten die het systeem moet hebben)
- Specificeren de criteria's om de kwaliteit van de werking van het systeem te beoordelen
- Ze worden ook **Architecturally Significant Requirements** genoemd en gaan vaak over de kwaliteit van een systeem (daarom ook wel kwaliteitseisen genaamd).



# Non-Functional requirements

- Ze bevatten informatie over:
  - **Betrouwbaarheid** van het systeem
  - **Bruikbaarheid** van het systeem
  - **Informatiebeveiliging** van het systeem
  - **Performance** van het systeem
  - **Onderhoudbaarheid** van het systeem

Voorbeelden:

- De maximale tijd die nodig is om een uitgevallen onderdeel van het systeem te vervangen is 3 uur.
- Het duurt maximum 5 minuten tot de gebruiker een mail ontvangt nadat die zich heeft aangemeld.

# ISO 25010 - Kwaliteit kenmerken



## Productkwaliteit

- Betrouwbaarheid (Reliability)
- Bruikbaarheid (Usability)
- Beveiligbaarheid (Security)
- Prestatie-efficiëntie (Performance efficiency)
- Uitwisselbaarheid (Compatibility)
- Onderhoudbaarheid (Maintainability)
- Overdraagbaarheid (Portability)

## Kwaliteit tijdens gebruik

- Effectiviteit (Effectiveness)
- Efficiëntie (Efficiency)
- Voldoening (Satisfaction)
- Vrijwaring tegen risico (Freedom from risk)
- Context dekking (Context coverage)

ISO\_25010, Wikipedia. [https://nl.wikipedia.org/wiki/ISO\\_25010](https://nl.wikipedia.org/wiki/ISO_25010)

# Productkwaliteit



Functionele geschiktheid	Prestatie-efficiëntie	Uitwisselbaarheid	Bruikbaarheid
Functionele compleetheid Functionele correctheid Functionele toepasbaarheid	Snelheid Middelenbeslag Capaciteit	Beïnvloedbaarheid Koppelbaarheid	Herkenbaarheid van geschiktheid Leerbaarheid Bedienbaarheid Voorkomen gebruikersfouten Volmaaktheid gebruikersinteractie Toegankelijkheid
Betrouwbaarheid	Beveiligbaarheid	Onderhoudbaarheid	Overdraagbaarheid
Volwassenheid Beschikbaarheid Foutbestendigheid Herstelbaarheid	Vertrouwelijkheid Integriteit Onweerlegbaarheid Verantwoording Authenticiteit	Modulariteit Herbruikbaarheid Analyseerbaarheid Wijzigbaarheid Testbaarheid	Aanpasbaarheid Installeerbaarheid Vervangbaarheid

Source: [http://www.gripoprequirements.nl/downloads/iso-25010-2011-een-introductie-v1\\_0.pdf](http://www.gripoprequirements.nl/downloads/iso-25010-2011-een-introductie-v1_0.pdf)

# Kwaliteit tijdens het gebruik



Effectiviteit	Efficiëntie	Voldoening	Vrijheid van risico	Context dekking
Effectiviteit	Efficiëntie	Bruikbaarheid Vertrouwen Tevredenheid Welzijn	Economisch risico beperking Gezond- en veiligheidsrisico beperking Omgevingsrisico beperking	Context compleetheid Flexibiliteit

Source: [http://www.gripoprequirements.nl/downloads/iso-25010-2011-een-introductie-v1\\_0.pdf](http://www.gripoprequirements.nl/downloads/iso-25010-2011-een-introductie-v1_0.pdf)



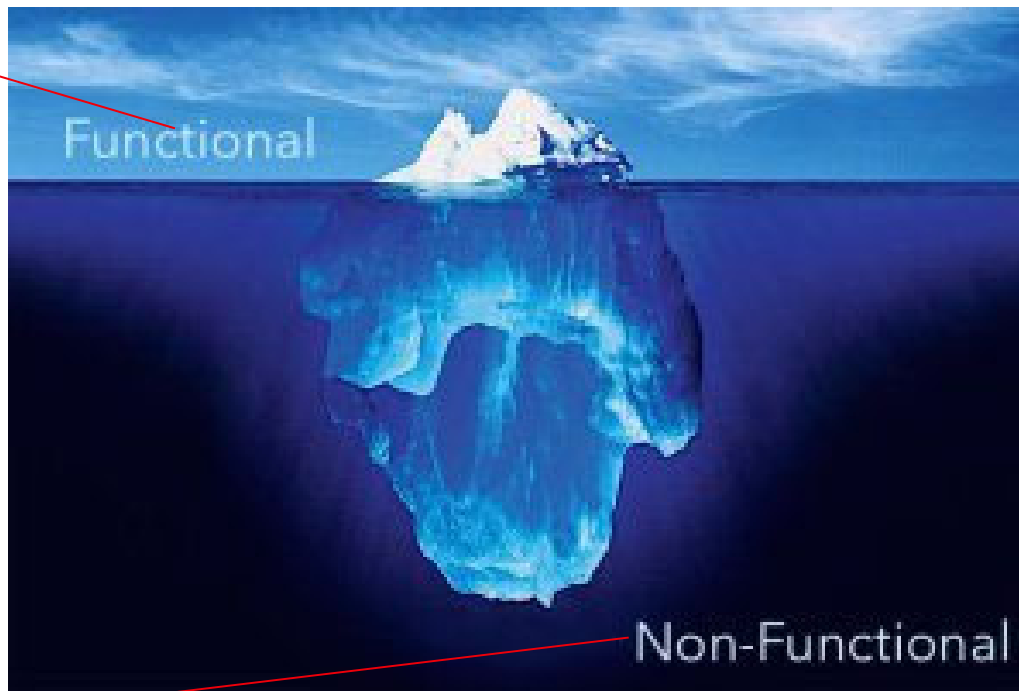
# FURPS – software kwaliteit eigenschappen

- **F**unctionality - Features, mogelijkheden, beveiliging
- **U**sability - Menselijke factoren, esthetisch eigenschappen, consistentie, documentatie
- **R**eliability - Faalfrequentie, betrouwbaarheid, faalimpact, herstelbaarheid, voorspelbaarheid, nauwkeurigheid, mean time to failure
- **P**erformance - Snelheid, efficiëntie, resourceverbruik, throughput, responsetijd
- **S**upportability - Testbaarheid, uitbreidbaarheid, aanpasbaarheid, onderhoudbaarheid, compatibiliteit, configureerbaarheid, serviceability, installeerbaarheid, localizability, overdraagbaarheid

# Functional & non-Functional ijsberg



**Specificeren gedrag of functie van het systeem.**



**Beschrijf hoe het systeem moet werken en zich gedragen**

Resource: <https://kencourses.com/tc1019fall2016/syndicated/the-functional-and-non-functional-iceberg/>

# Opdracht

## Non-Functional Requirements:



**Noteer de non-functional requirements uit het ISO25010:2011 document die van toepassing zijn voor je project**

NF01 – Onderhoud Interval	
<i>Omschrijving</i>	De robot moet drie keer achter elkaar ingezet kunnen worden zonder enige vorm van onderhoud.
<i>Rationale</i>	De robot moet betrouwbaar zijn in een operatie.
<i>Business Priority</i>	Should have



# Inhoudsopgave

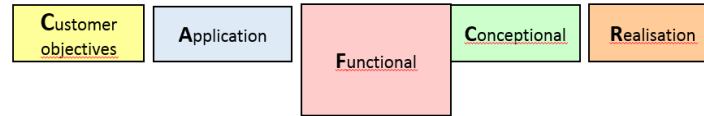
Requirements opstellen

Functional Requirements

Non Functional Requirements

Constraints

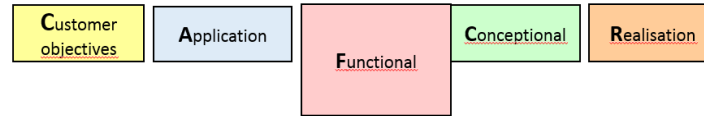
Conclusies



# Constraints

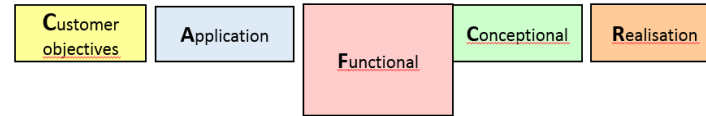
- **Constraint**
  - Is een voorwaarde van buitenaf aan het systeem opgelegd die het aantal mogelijke oplossingen beperkt
  - Toont de grenzen waarbinnen het systeem moet worden gerealiseerd.

Bijvoorbeeld : Een zeker apparaat mag vanwege wetgeving niet meer dan 80dB aan geluid produceren.



# Constraints

- Constraints kunnen economische, politiek, technisch of milieutechnisch aard zijn
- Constraints kunnen betrekking hebben op projectplan, project resources en het systeem zelf.



# Requirements vs Constraints

## Requirements

**Wat moet  
systeem doen**

## Constraints

**Voorwaardes die de  
oplossing beperken.**

# Opdracht Constraints:

## Leg de belangrijkste constraints vast zoals op onderstaande manier (voeg een rationale toe indien nodig)



*Na het gesprek met de klant, is er een budget vastgesteld van wat de robot uiteindelijk mag gaan kosten. Dit is hieronder vastgelegd in constraints.*

C00 – Productiekosten	
<i>Omschrijving</i>	Productiekosten moeten onder €25.000 per unit vallen.
<i>Business Priority</i>	Must have

C01 – Onderhoudskosten	
<i>Omschrijving</i>	Onderhoudskosten moeten onder de €2.000 per onderhoudsbeurt vallen
<i>Business Priority</i>	Must have

- Constraints kunnen economische, politiek, technisch of milieutechnisch aard zijn
- Constraints kunnen betrekking hebben op projectplan, project resources en het systeem zelf.



# Referenties

Hoofdstuk 7 van Muller (2013)  
*Architectural Reasoning Explained.*

[http://www.gripoprequirements.nl/downloads/iso-25010-2011-een-introductie-v1\\_0.pdf](http://www.gripoprequirements.nl/downloads/iso-25010-2011-een-introductie-v1_0.pdf)