

# **Key Drivers – *Application drivers*** ***Requirements***

System Engineering (TCTI-V2SYEN-16)  
week 3, les A

Marius Versteegen

Auteur: Joost Schalken-Pinkster

# **Terugblik Klantgesprekken**

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	

# Inhoudsopgave



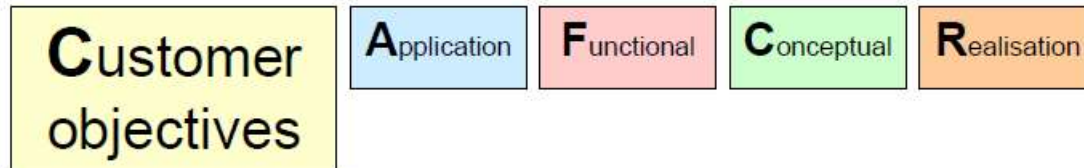
Key Drivers

Application Drivers

Requirements

Key Driver Graph

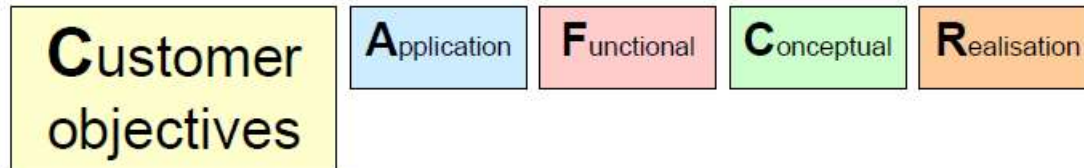
# Key Drivers



**WAT** wil de klant ?

- De essentiële doelstellingen van de klant
- Bied richting voor het vastleggen van de requirements
- Vastleggen van de relatie tussen klant de key drivers en de requirements en design

# Key Drivers



**Naam** van een

**Driver**

is bij voorkeur een

**Zelfstandig Naamwoord**

Bijvoorbeeld

Klanttevredenheid

# Tips voor het opstellen van key driver



- **Beperk** het aantal key drivers
  - minimaal 4, maximaal 8
- Vergeet niet de voor **de hand liggende key drivers**
  - bijvoorbeeld de bekende belangrijkste functie van het product
- Gebruik **korte namen**, die herkenbaar zijn voor de klant
- Gebruik zoveel mogelijk markt-en klantspecifieke namen
  - vervang "rendement" door "integrale kosten per patiënt"

**Opdracht:** Verfieer je het lijstje met stakeholders en belangen (key drivers) uit de vorige les nog eens.

Vergewis dat ze voldoen aan de onderste 3 criteria van de bovenstaande lijst. Verbeter indien nodig.

# Inhoudsopgave



Key Drivers

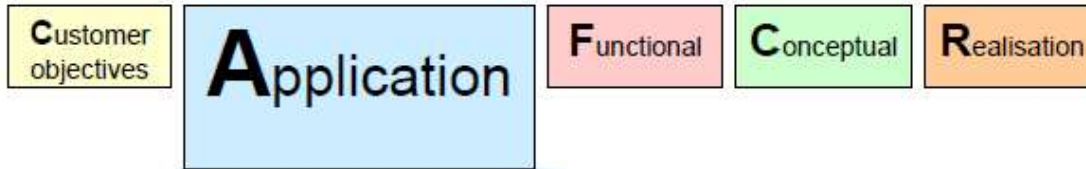
Application Drivers

Requirements

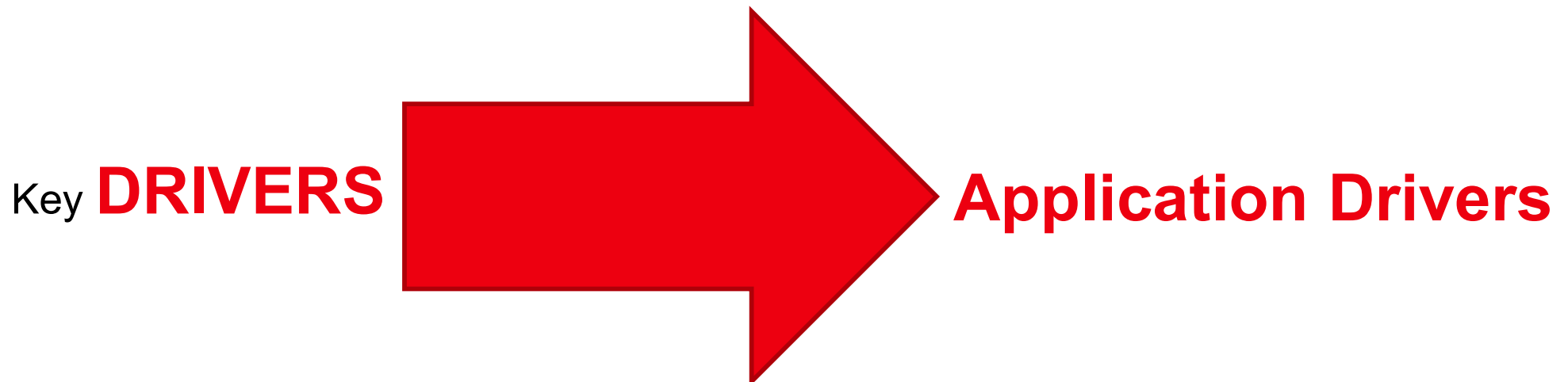
Key Driver Graph



# Application Drivers

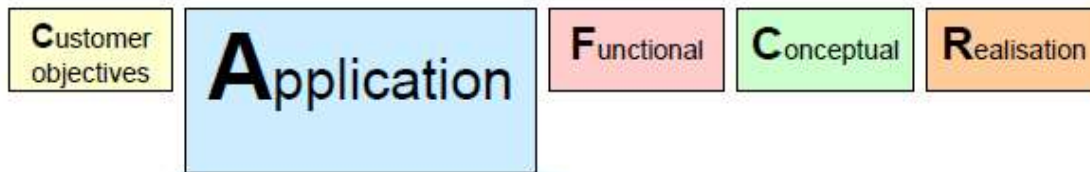


**WAT** wil de klant?



**WAT** willen de stakeholders?

# Application Drivers



- Zijn middelen om de key drivers te bereiken
- Zijn de diensten die het systeem moet leveren zodat de klant/stakeholder (een deel van) zijn doelen kan bereiken.

## Key drivers

- Veiligheid en verantwoordelijkheid
- Rapportage-kwaliteit

## Application drivers

- Patiënt Identificatie, Procedures volgen, Diagnose-informatie beveiligen
- Selecteren van relevant materiaal, gebruik maken van standaarden

**TIP:** Maak je geen zorgen over de exacte **Grens** tussen Doelen/key drivers en Application Drivers  
Maak vooral **de relaties** tussen **Doel** en **Middel** helder

- Application drivers zijn de **diensten** die het **systeem** moet leveren zodat de klant (een deel van) zijn doelen kan bereiken. **(sub-doelen)**

Je komt ze vaak net voor de key-drivers tegen tijdens het doorvragen met “waarom”

## Opdracht:

Vind bij elk van je key drivers de bijbehorende application drivers en orden die in een tabel **op prioriteit**. Beargumenteer in begeleidende tekst (buiten de tabel) de keuze van de prioritering

Key Drivers	Application Drivers	Toelichting / Rationale
<b>Tijd besparen</b>	Snel en makkelijk bij te vullen Automatisch de katten voeren. Trainingsfaciliteiten bieden.	Het zou tijd besparen als er faciliteiten worden geboden waardoor de katten snel met het apparaat overweg kunnen gaan.
<b>Gemak</b>	De katten automatisch de juiste hoeveelheid voer geven. Een dieetplan over een langere periode kunnen instellen. Makkelijk te reinigen	Zo zou bijvoorbeeld de hoeveelheid voer die de katten gewend zijn te eten over een periode langzaam kunnen worden afgebouwd.
<b>Prijs</b>	Goedkope onderdelen Goedkope assemblage Lage Verzendkosten Lage Verzekeringskosten	
<b>Winst</b>	Zelfde als prijs en compatibiliteit. Verder:  Onderscheidend vermogen	Naarmate de kattenvoermachine zich (op nuttige manier) meer van de concurrentie onderscheidt, is de kans groter dat de klant bereid is er een grotere prijs voor te betalen
<b>Veiligheid</b>	Voldoet aan internationale veiligheidsnormen.  De gebruiker wordt in kennis gesteld van eventuele gevaren.  Er mag geen risico gelopen worden dat een kat ondervoed raakt.  Het apparaat mag geen letsel aan mens of dier veroorzaken.	
<b>Comfort voor Kat</b>	Kan geen pijn veroorzaken bij de kat.  De kat hoeft niet te veel moeite te doen om te eten.  De kat wordt niet aan het schrikken gebracht.	
<b>Compatibiliteit</b>	Werkt met de meest gangbare typen kattenvoer van de grootste Kattenvoerfabrikanten  Werkt met de meest gangbare afmetingen van huiskatten.  Werkt met de meest gangbare RFID-chips voor huiskatten.	
<b>Betrouwbaarheid</b>	Het apparaat mag bij de goedgekeurde typen kattenvoer niet te vaak vastlopen.  Het actueel uitgevoerde dieetschema mag niet te veel afwijken van het ingestelde dieetschema.	
<b>Geen overlast</b>	Het apparaat moet handige vorm en afmetingen hebben voor in-huisgebruik.  Het apparaat mag niet te veel storende geluiden voortbrengen.  Het apparaat mag niet te veel energie verbruiken.	

# Inhoudsopgave

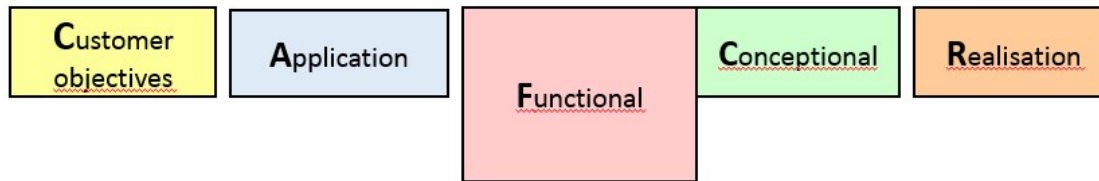
Key Drivers

Application Drivers

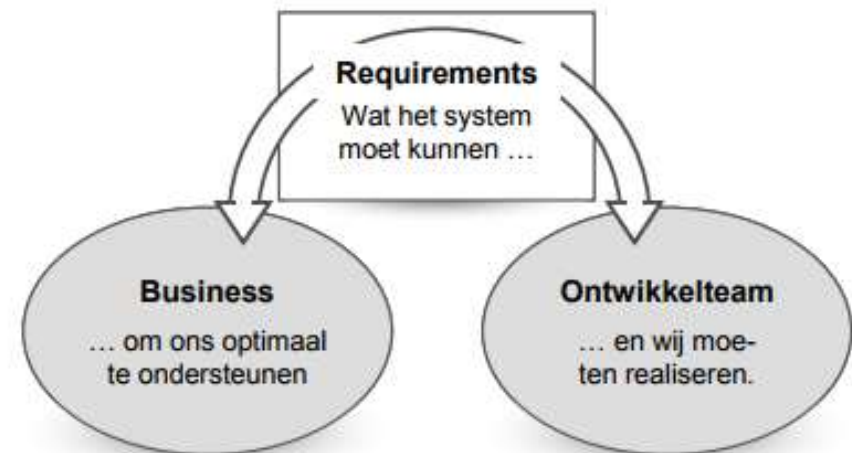
Requirements

Key Driver Graph

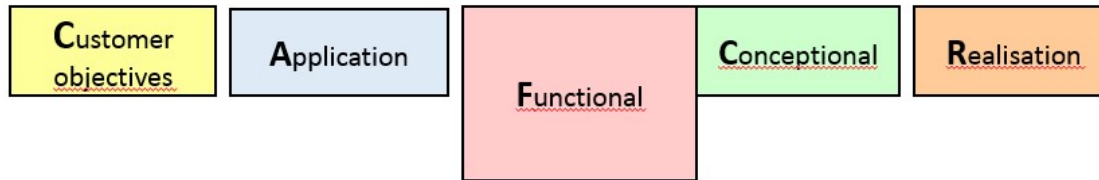
# Wat zijn Requirements?



- Requirements zijn een beschrijving van wat de **stakeholders** van het systeem **verwachten**.
- .. En wat het **ontwikkelteam** moet **opleveren**.



# Requirements



Om conflicten achteraf te voorkomen:

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

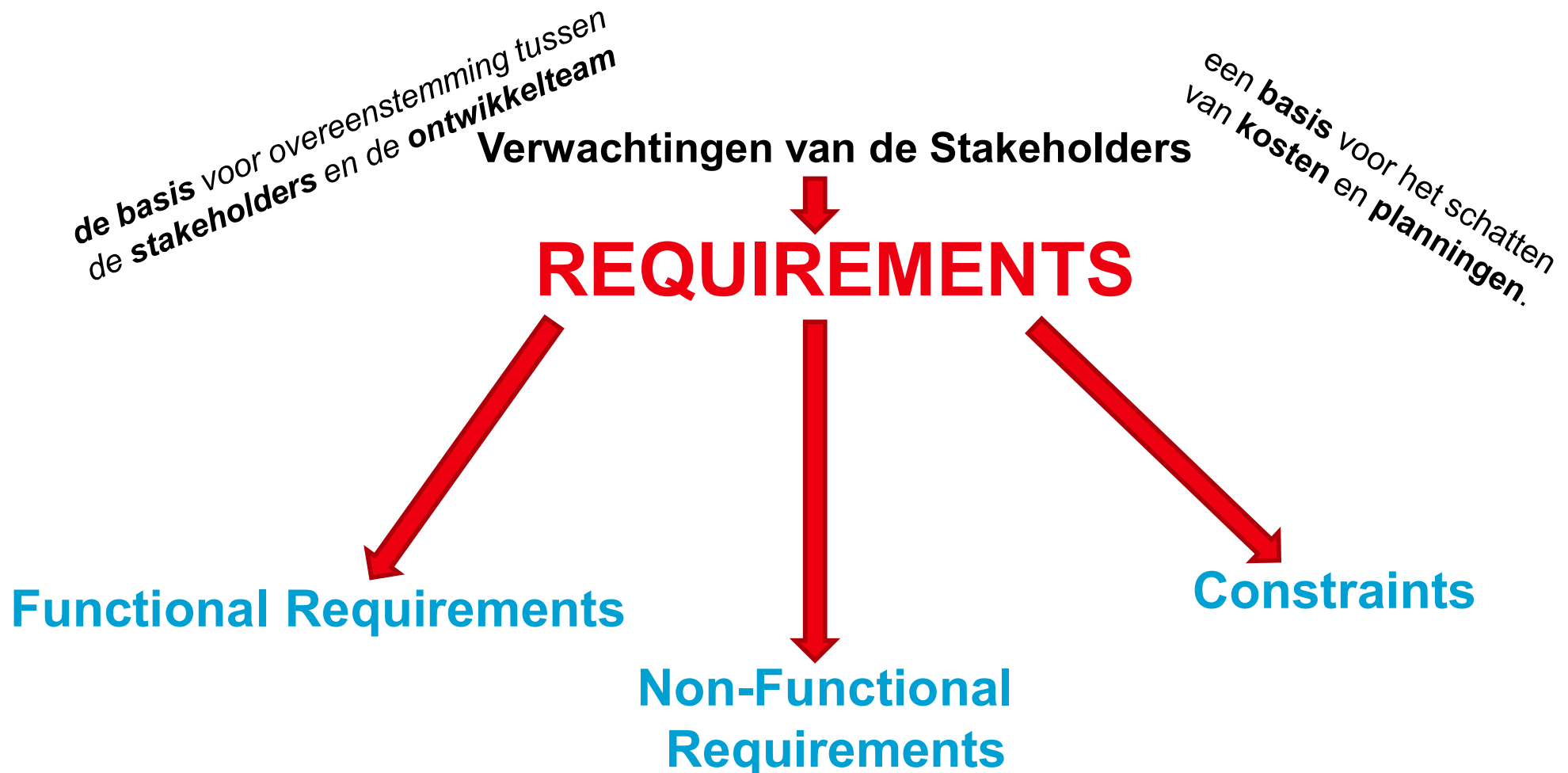


## Voordelen van goed geschreven requirements

- Vormt **de basis** voor overeenstemming tussen de **stakeholders** en het **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**.



# Drie soorten 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.*

# Constraints



- **Beschrijven beperkingen van je systeem**
- **Beperkingen van buitenaf** – waar je niets aan kunt doen  
(door wetgeving, budget, natuurwetten, e.d)
- *Voorbeeld:*  
“De snelheid waarmee de sonar objecten kan detecteren is  
gelimiteerd door de geluidssnelheid:  
 $\text{minimaleDetectieTijd} = \text{objectAfstand} / \text{snelheidVanGeluidInWater}$ ”

# Functional Requirements



- Beschrijven **wat** het systeem moet **doen**
- *Voorbeeld:*  
“Het sonarsysteem moet onderscheid kunnen maken tussen vriendelijke en vijandelijke onderzeeers die in bereik zijn.”

# Non-Functional Requirements



- Beschrijven de **overige kenmerken** van het systeem.
- **Kwaliteitsaspecten** (performance, onderhoudbaarheid, etc)
  - bijvoorbeeld **de mate waarin / hoe** functionele requirements vervuld moeten worden.

*Voorbeeld: “Het herkennen van een vriendelijke onderzeeer als vriendelijk binnen een afstand van 1000 meter moet in minstens 99% van de gevallen correct zijn.”*

- Beschrijving van **interfaces**:

*Voorbeeld: “Het sonar-systeem is via een ethernet verbinding verbonden met het navigatie systeem van de onderzeeer”*

- Beschrijving van algemene **fysieke kenmerken**:

*Voorbeeld: “Het sonar-systeem neemt niet meer dan 5 kubieke decimeter aan ruimte in beslag”*

# Inhoudsopgave

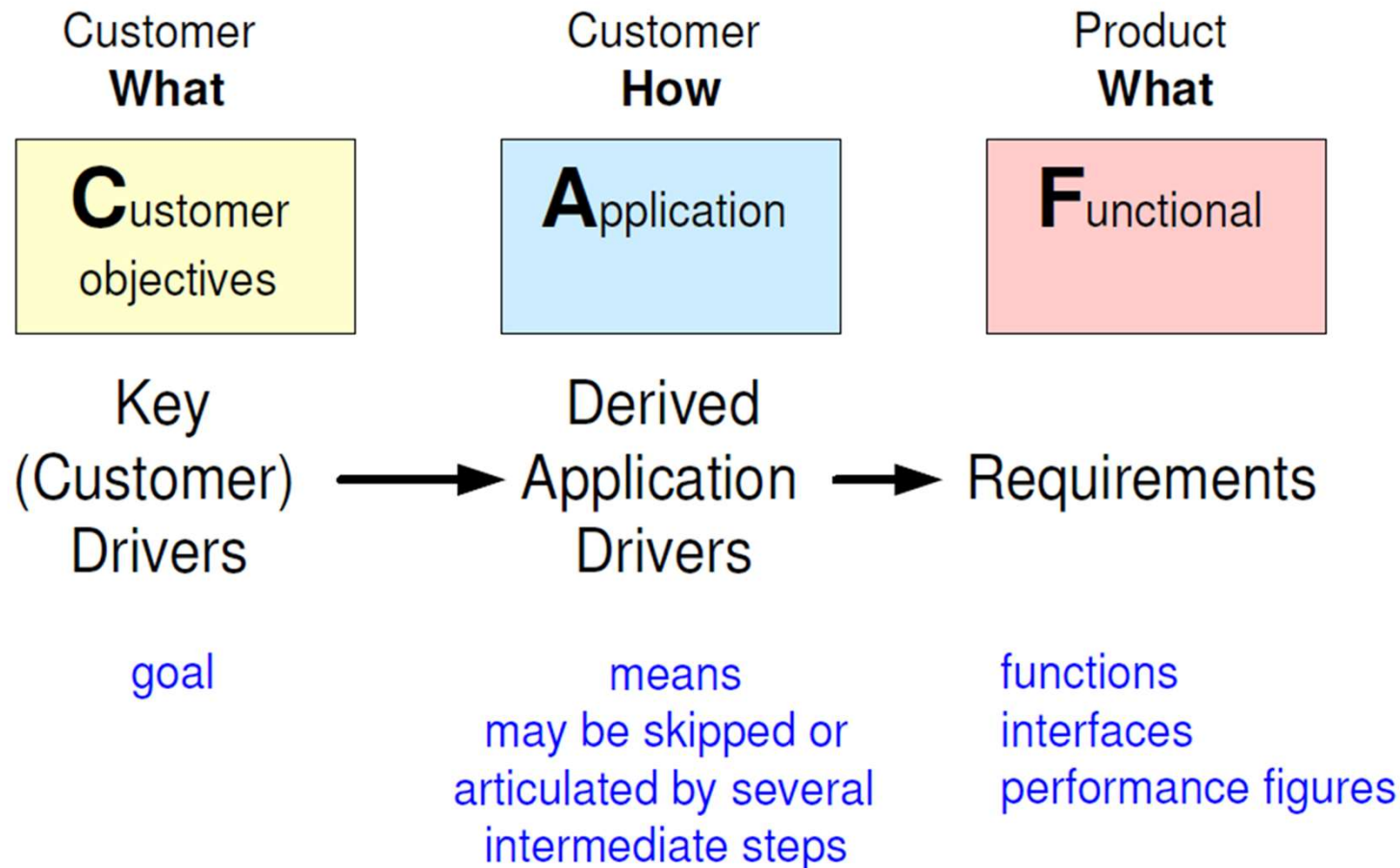
Key Drivers

Application Drivers

Requirements

Key Driver Graph

# CAF View



# Key Driver Graph



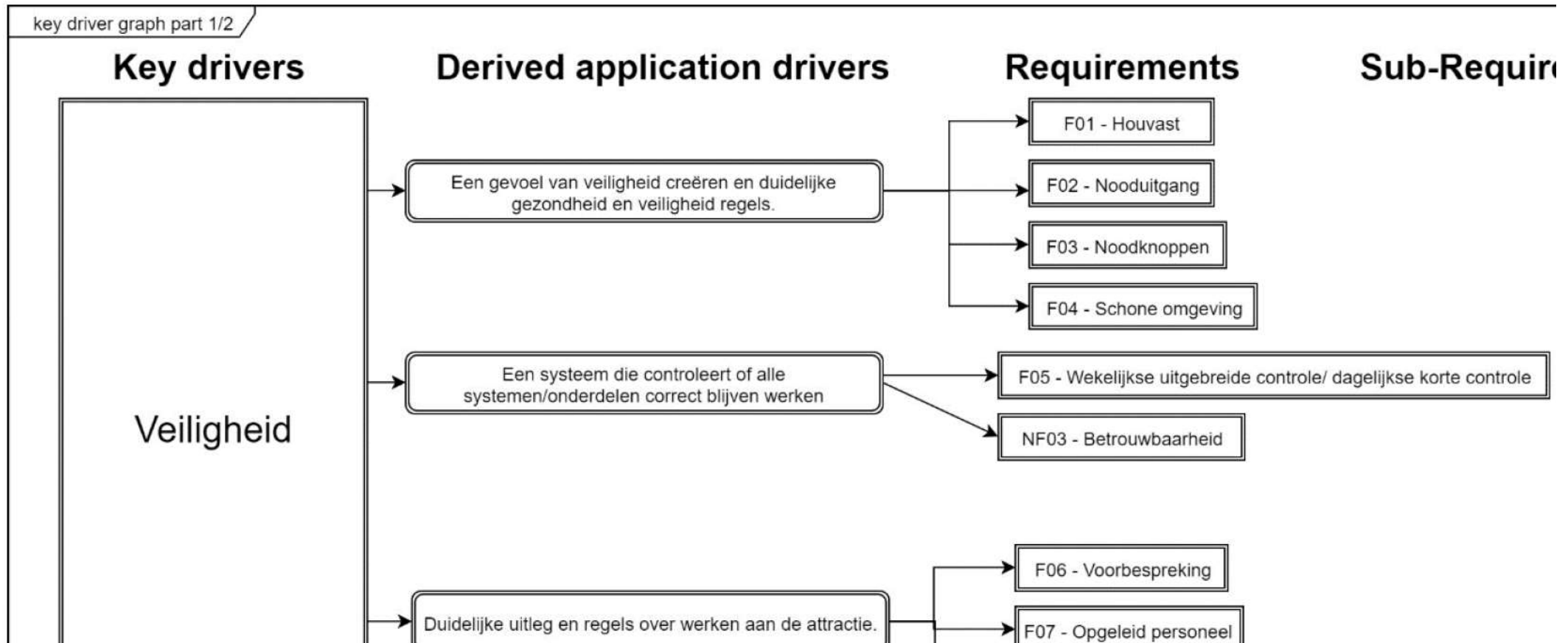
**De Key Driver Graph** is een grafiek die de **koppeling** tussen de **key drivers** en de daarvan afgeleide **requirements** visualiseert

Dankzij die grafiek kunnen we:

- De **klant** beter te **begrijpen**  
(ook door er met de klant over te discussieren)
- Beter het **belang** van elke **requirement** beoordelen.
- Iteratief **verbeteren**  
(Bijvoorbeeld door in te zien dat bepaalde application drivers eigenlijk requirements zijn, en vica versa)

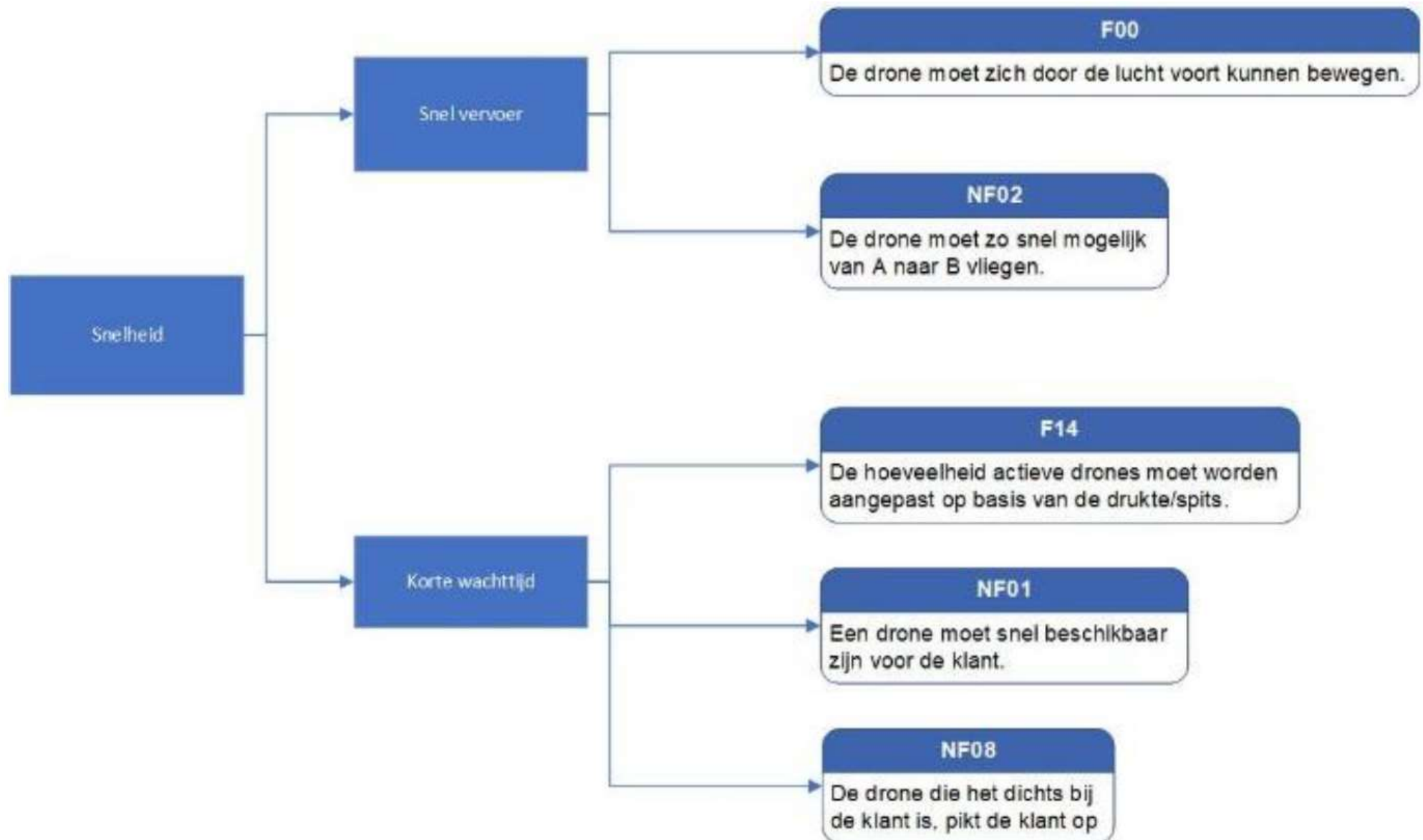
Daardoor kunnen we beter **met de klant afstemmen** wat voor product hij nodig heeft en het project **gerichter leiden**.

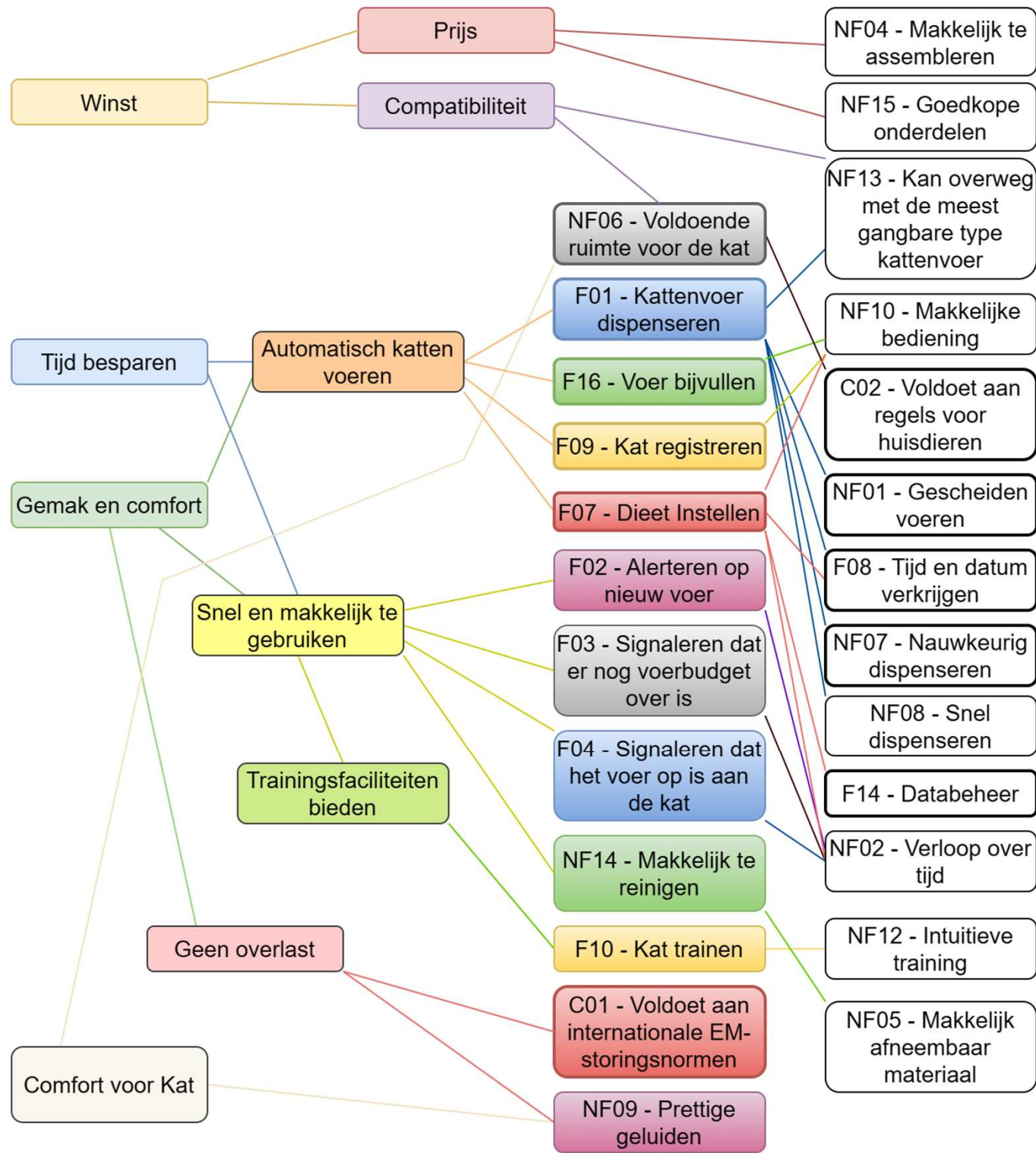
# Voorbeeld van een key-driver graaf





# Nog een voorbeeld van een key-driver graaf

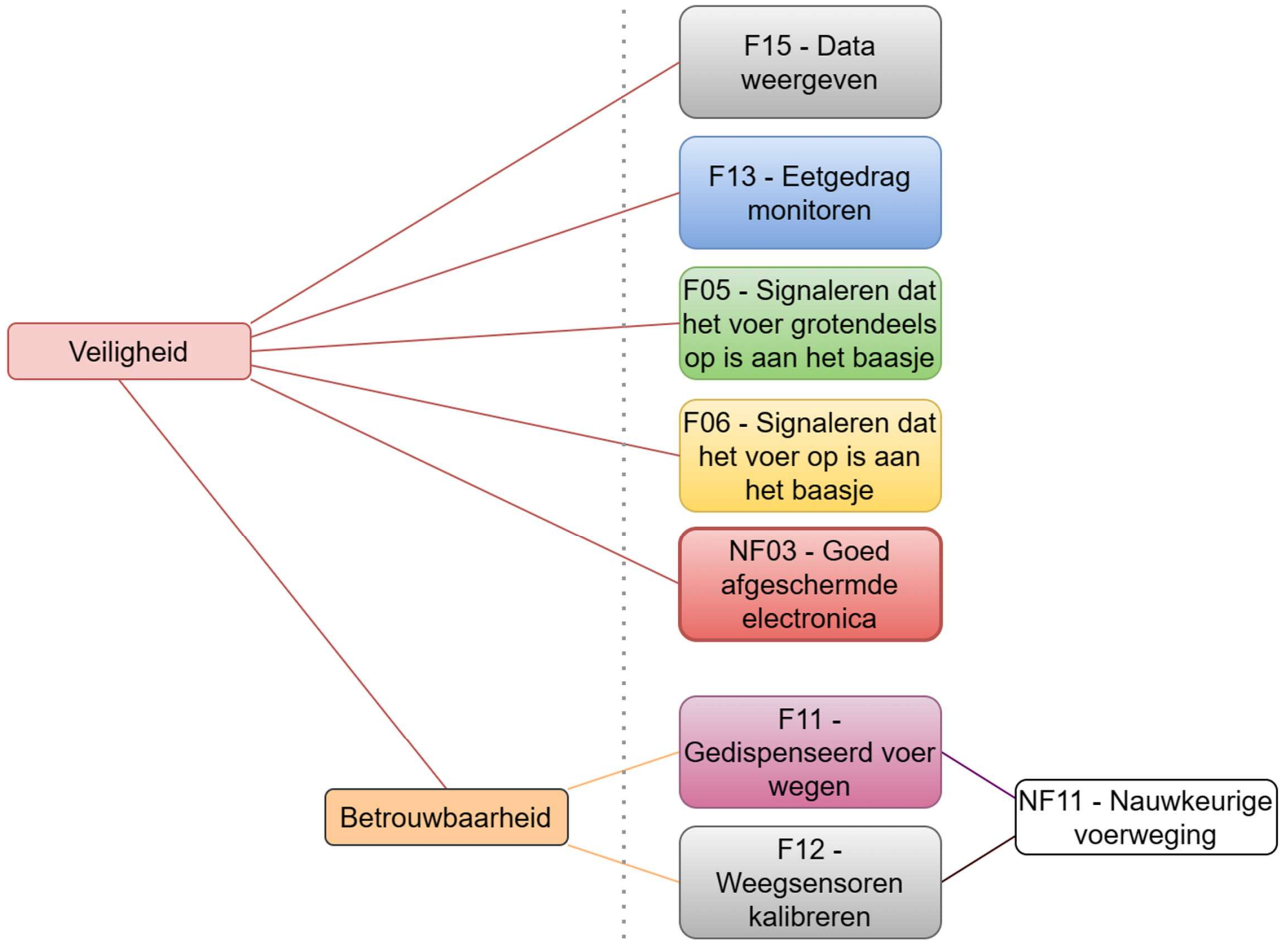




## Key drivers

## Application Drivers

## Requirements



# Key Drivers Graaf



**C**ustomer  
objectives

**A**pplication

**F**unctional

1. Definieer scope voldoende specifiek
  - in termen van belanghebbenden of marktsegmenten
2. Verwerf en analyseer alle feiten
  - haal feiten uit specificaties van bestaande producten
  - vraag waarom vragen over specificaties van bestaande producten
3. Bouw een graaf met de relaties tussen key drivers en eisen d.m.v. brainstormen en discussies
  - soms hebben eisen meerdere drivers
4. Feedback verkrijgen
  - bespreken met klanten, observeren van reacties
5. Vaak itereren
  - beter begrip leidt vaak tot de verhuizing van kwesties van application driver naar functionele eis of vice versa.

Source:

Gerrit Muller (2013) Key Drivers How To. preliminary draft. version: 0.2. Buskerud University College, maart, 2013.  
<http://www.gaudisite.nl/KeyDriversHowToPaper.pdf>

# Requirements Types



## Constraints

- Beschrijven beperkingen van je systeem
- Beperkingen van buitenaf – waar je niets aan kunt doen
- (door wetgeving, budget, natuurwetten, e.d)

## Functional Requirements

- Beschrijven wat het systeem moet doen

## Non-Functional Requirements

- Beschrijven de overige kenmerken van het systeem.
- Kwaliteitsaspecten
- Beschrijving van interfaces
- Beschrijving van algemene fysieke kenmerken:

**Opdracht:** Probeer aan de hand van de application-drivers die je al gevonden hebt wat bijbehorende requirements te vinden. Noteer het geheel in een **Key-Driver-Diagram**.

# Huiswerk



- **Verder uitwerken** van de opdrachten van vandaag
- De lesstof voor de volgende les **voorbereiden**.