

SMART requirements schrijven

Reverse Engineering als aanpak voor leren

Requirements Kenniscentrum

27 maart 2012, 18:50 – 19:30 uur

Hossein Chamani, docent en trainer bij Hogeschool Rotterdam

Introductie

- Gewerkt in Telecom als projectmanager, testmanager, Network Manager en trainer testmanagement
- Docent software testing, analytisch schrijven en telecommunicatie bij Hogeschool Rotterdam

Introductie

- Verwerf graag kennis uit de praktijk en verspreid deze kennis
- Momenteel bezig met een trainingsboek over SMART requirements schrijven
- De onderdelen worden via een weblog geleidelijk gepubliceerd

De volgende foto's geven een indruk hoe ik als docent graag:

- **studenten train voor de praktijk**
- **vakkennis verwerf uit de praktijk**



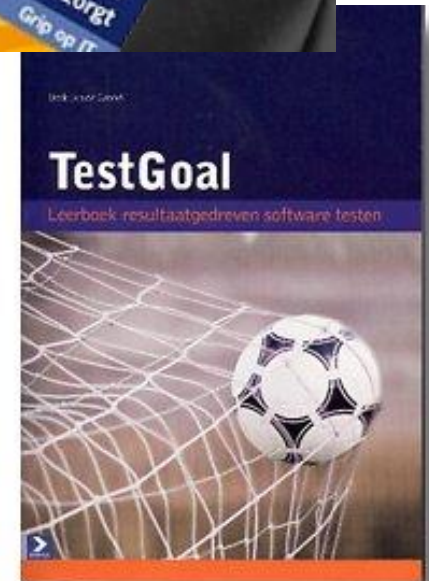
Studenten worden getraind bij ICT-bedrijven om de motivatie en praktijkgerichtheid te verhogen.⁵



Studenten krijgen bij bedrijven trainingsopdrachten en ze voeren deze uit op de werkplek van ICT'ers.



Experts worden uitgenodigd voor het trainen
en inspireren van jonge talenten



Manuscripten reviewen en auteurs feedback geven tijdens het schrijfproces



Inhoud presentatie

SMART requirements schrijven
Wat en waarom?

SMART requirements schrijven
Waarom Reverse Engineering als aanpak voor trainingen?

Enkele aanbevelingen voor leren SMART requirements schrijven volgens Reverse Engineering



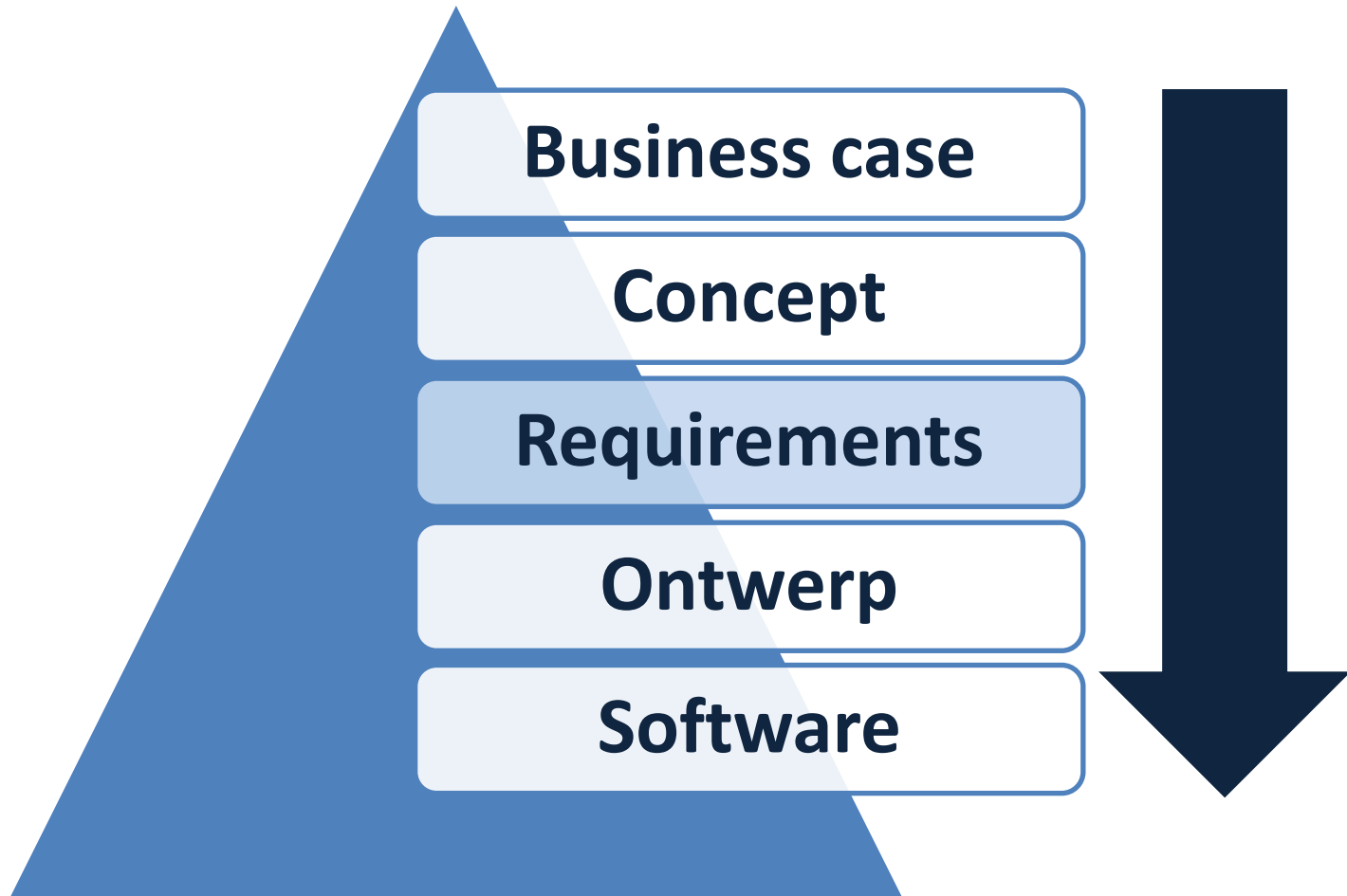
Inhoud presentatie

SMART requirements schrijven
Wat en waarom?

SMART requirements schrijven
*Waarom Reverse Engineering als aanpak
voor trainingen?*

Enkele aanbevelingen voor leren
SMART requirements schrijven volgens
Reverse Engineering

Requirements in de context van softwareontwikkeling



Requirements in de context van softwareontwikkeling

Requirements beschrijven een behoefte. Ze maken de behoefte expliciet en vormen de informatiebasis voor het ontwerpen, ontwikkelen en testen van softwareproducten.

SMART Requirements: *Wat?*

SMART Requirements zijn eenduidig beschreven en bevatten geen logische fouten.

- Specifiek
- Meetbaar
- Acceptabel
- Realistisch
- Tijdsgebonden

SMART requirements schrijven: *Waarom?*

SMART requirements helpen de ontwerpers, ontwikkelaars of testers om:

```
graph TD; A[SMART requirements helpen de ontwerpers, ontwikkelaars of testers om:] --> B[doeltreffende producten te kunnen leveren]; A --> C[efficiënter te kunnen werken];
```

**doeltreffende producten
te kunnen leveren**

**efficiënter te
kunnen werken**



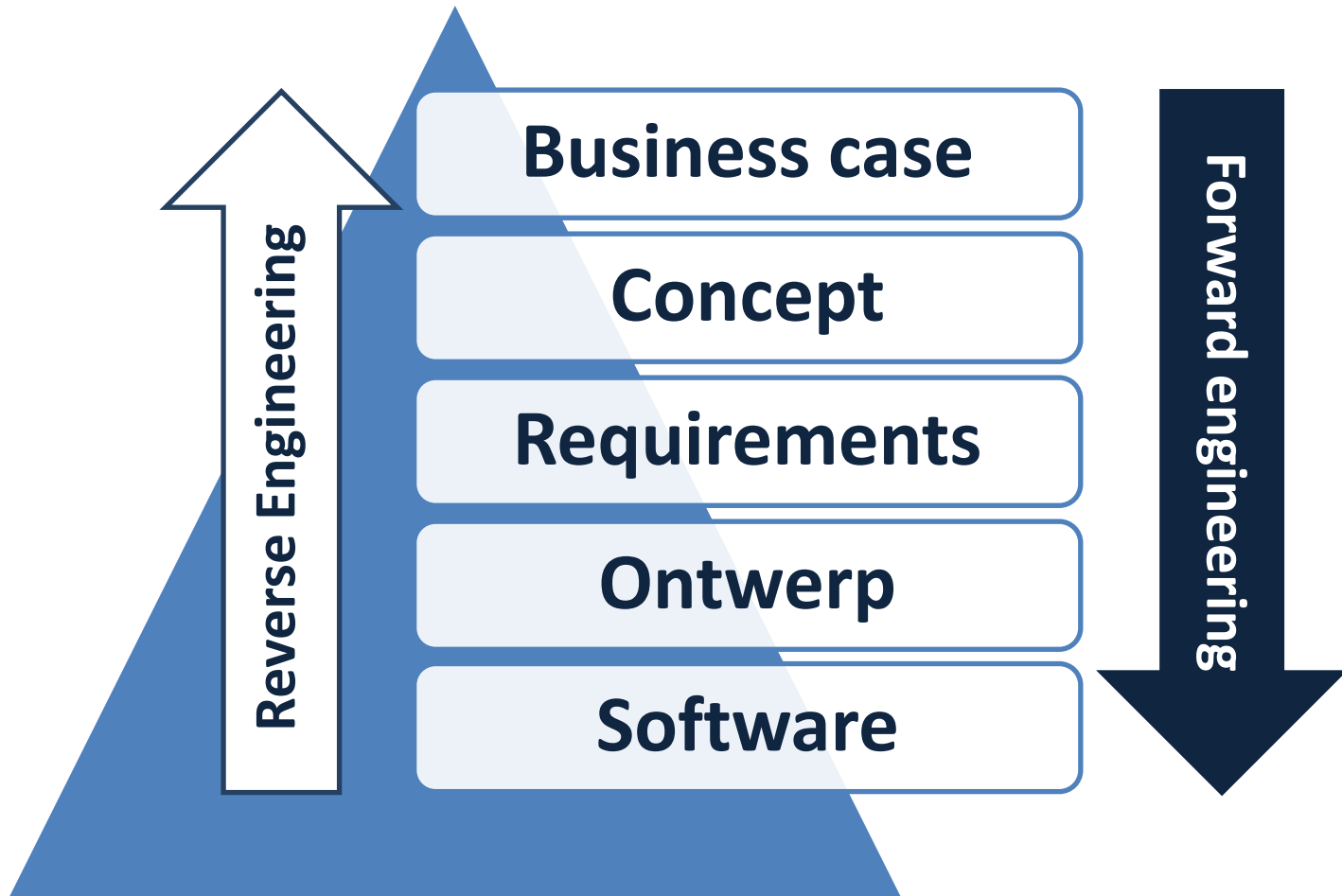
Inhoud presentatie

SMART requirement schrijven
Wat en waarom?

SMART requirement schrijven
*Waarom Reverse Engineering als aanpak
voor trainingen?*

Enkele aanbevelingen voor leren
SMART requirement schrijven volgens
Reverse Engineering

Reverse Engineering: *Wat?*



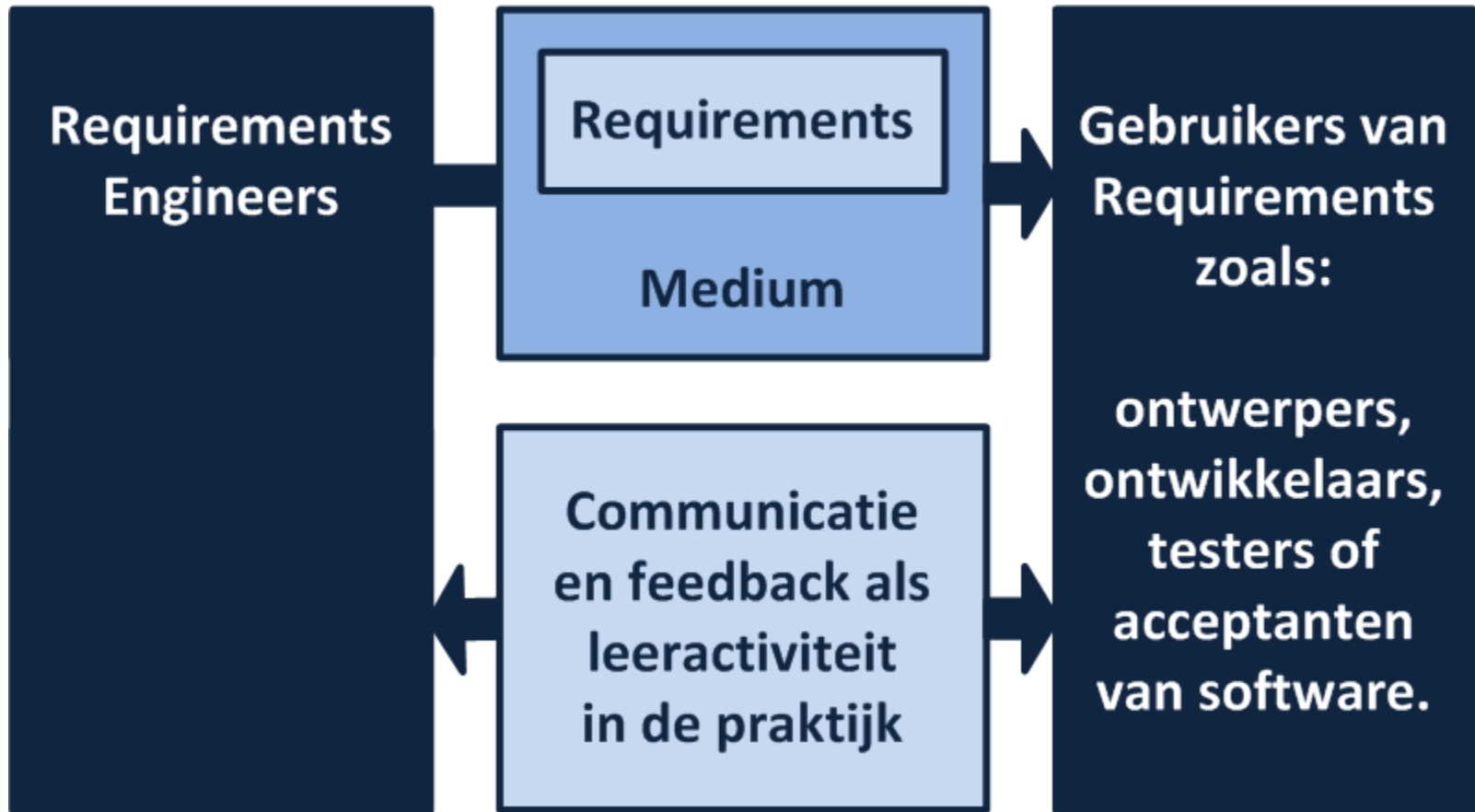
Reverse Engineering: *Waarom?*

Drie scenario's als voorbeeld

De klant wil een softwareproduct laten nabouwen omdat:

1. de leverancier niet meer bestaat
2. de gebruikte technologie is verouderd
3. de klant niet tevreden is over de leverancier.

Communicatiemodel voor leren van en met andere experts



Copyright Hossein Chamani

Waarom Reverse Engineering als invalshoek voor leren SMART requirements schrijven?

Requirements Engineers kunnen SMART requirements leveren als ze meer inzicht hebben in de denk- en werkwijze van de gebruikers van requirements zoals:

- ontwerpers
- ontwikkelaars
- testers

Waarom Reverse Engineering als invalshoek voor leren SMART requirements schrijven?

Door Reverse Engineering krijgen Requirements Engineers meer inzicht in de denk- en werkwijze van de gebruikers van requirements zoals:

- ontwerpers
- ontwikkelaars
- testers

Waarom Reverse Engineering als invalshoek voor leren SMART requirements schrijven?

Reverse Engineering maakt het mogelijk om:

- trainingen snel en makkelijk te concretiseren
- flexibel te leren
- exploratief te leren
- meer plezier te hebben van het leren



Inhoud presentatie

SMART requirements schrijven
Wat en waarom?

SMART requirements schrijven
*Waarom Reverse Engineering als aanpak
voor trainingen?*

**Enkele aanbevelingen voor leren
SMART requirements schrijven volgens
Reverse Engineering**

Leren SMART requirements schrijven volgens Reverse Engineering: *voorbeeldscenario*

**Analyseer
een bestaand softwareproduct**

```
graph TD; A[Analyseer een bestaand softwareproduct] --> B[Schrijf business case]; A --> C[Schrijf concept]; A --> D[Schrijf SMART requirements];
```

**Schrijf
business case**

**Schrijf
concept**

**Schrijf
SMART
requirements**

Wanneer bestaande softwareproducten analyseren en SMART requirements schrijven?

U kunt bestaande softwareproducten analyseren en SMART requirements schrijven wanneer u zich wilt voorbereiden op bijvoorbeeld:

- opdrachten bij (nieuwe) klanten
- opdrachten in nieuwe branches
- nieuwe systemen of technologieën

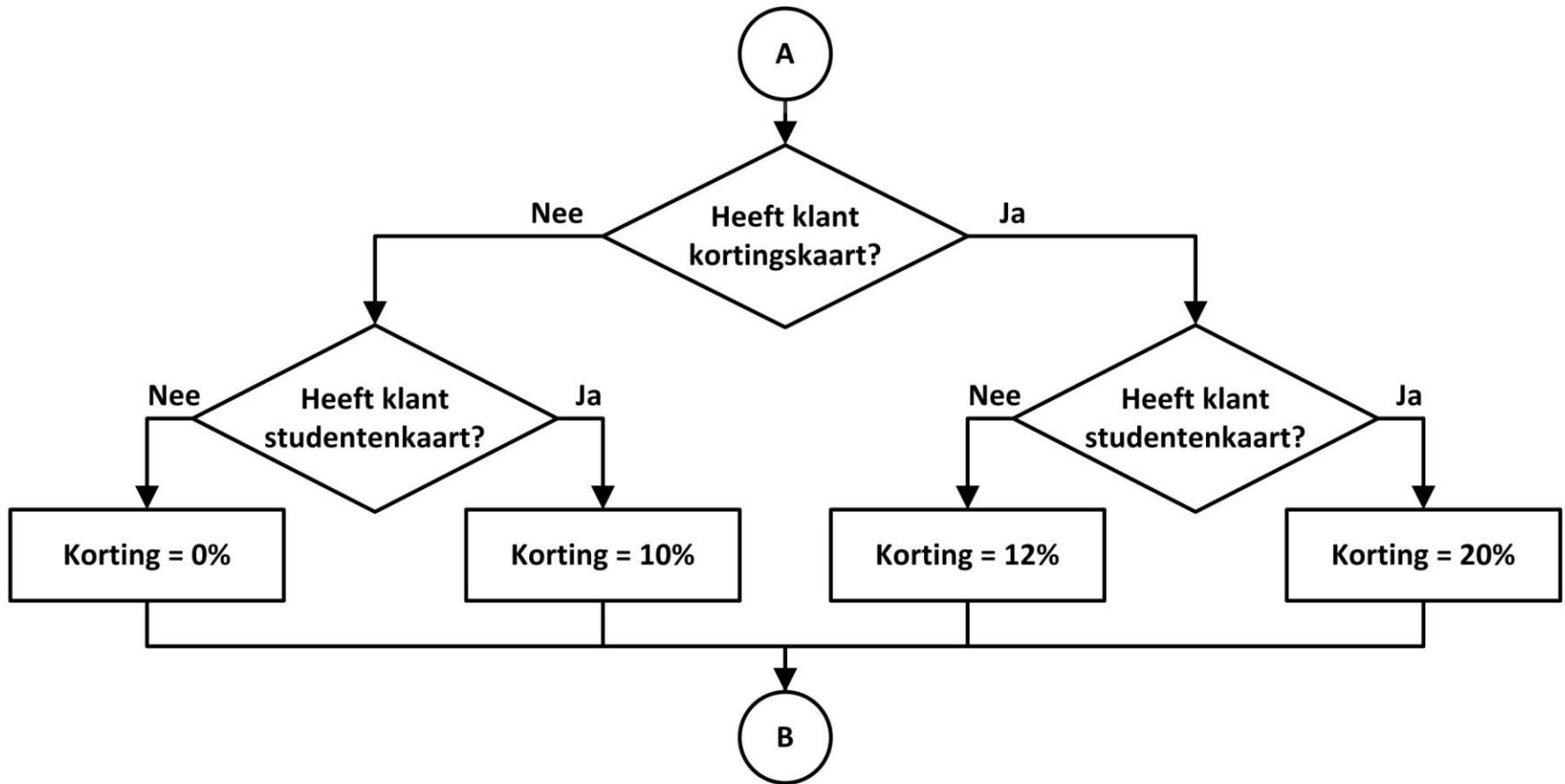
Vertaal het ontwerp van een bestaand softwareproduct naar SMART requirements

Analyseer
het ontwerp van een bestaand softwareproduct



Schrijf
SMART requirements

Vertaal het ontwerp van een bestaand softwareproduct naar SMART requirements



Vertaal het algoritme of de code van een bestaand softwareproduct naar SMART requirements

**Analyseer
het algoritme of de code van een softwareproduct**



**Schrijf
SMART requirements**

Vertaal het algoritme of de code van een bestaand softwareproduct naar SMART requirements

Read *Dienstjaren*

If *Dienstjaren* > 20 Then *Vakantiedagen* = 35

Else

 If *Dienstjaren* > 10 Then *Vakantiedagen* = 30

 Else

 If *Dienstjaren* > 5 Then *Vakantiedagen* = 25

 Else

Vakantiedagen = 20

 Endif

 Endif

Endif

Print *Vakantiedagen*



Samenvatting

SMART requirement schrijven
Wat en waarom?

SMART requirements schrijven
Waarom Reverse Engineering als aanpak voor trainingen?

Enkele aanbevelingen voor leren SMART requirements schrijven volgens Reverse Engineering

SMART requirements schrijven

Reverse Engineering als aanpak voor leren

**Kritische reflectie of vragen
over de voor- en nadelen?**