

Software Engineering (100094)

College 2: *Requirements-engineering*

Marko van Eekelen

marko@cs.ru.nl

kamer HG02.074

Inhoud

1. 6 feb: Het systeemontwikkelp proces
2. **13 feb: Requirements-analyse**
3. 6 mar: Documentatie, Kwaliteit
4. ... : Architectuur, Object-oriëntatie
5. ... : Ontwerp
6. ... : Menselijke factoren
7. ... : Testen
8. ... : ...
9. ... : ...

College 2: Leerdoelen

- Hoofdstuk 7 van Pressman
- Wat is requirements-engineering?
- Waarom is het belangrijk?
- Welke stappen?
- Hoe weet je dat je 't goed gedaan hebt?

Requirements

- Meetbare uitspraken
over de **diensten** die een systeem
verwacht wordt aan te bieden,
en de **condities** waaronder deze moeten
worden uitgevoerd

Soorten requirements

- **Functionele** requirements
 - wat het systeem moet doen
- **Niet-functionele** requirements
 - Bijv. performance karakteristieken,
 - Randvoorwaarden t.a.v. ontwikkeling/beheer
 - Gebruik van standaarden

Niet-functionele requirements

- **Product**
 - Efficiëntie (performance, geheugen)
 - Bruikbaarheid
 - Betrouwbaarheid
 - Portabiliteit
- **Organisatie**
 - Oplevering
 - Implementatie
 - Standaarden
- **Extern**
 - Ethisch
 - Interoperabiliteit
 - Juridisch (privacy, veiligheid)

Requirements formuleren

In natuurlijke taal, maar pas op:

- Wees **helder**,
- Maak duidelijk **onderscheid** tussen **functionele** en **niet-functionele requirements**, **doelstellingen** en **ontwerpinformatie**,
- Kies het **juiste abstractieniveau**,
- Houd verschillende requirements **duidelijk gescheiden** (voeg geen verschillende requirements samen)

Vb. van slecht requirement

Om te helpen bij het positioneren van onderdelen van een diagram, kan de gebruiker een grid in cm of inches aanzetten via een optie in het controlpanel. Initieel staat het grid uit. Te allen tijde kan het grid aan of uit worden gezet, en kan er worden overgegaan van cm naar inches of andersom. De gridoptie is ook van toepassing op de 'reduce-to-fit' view, maar dan kan het aantal gridlijnen gereduceerd worden om te voorkomen dat het grid te fijnmazig wordt.

Software requirements specification

- **Voorwoord** – doelgroep, versiehistorie
- **Introductie** – doelstellingen, context, scope
- **Glossary** – definitie van technische termen
- **Globale requirements** – begrijpelijk voor gebruikers
- **Systeemarchitectuur** – globaal overzicht van de functionaliteit over componenten
- **Gedetailleerde requirements**
- **Systeemmodellen** – relatie systeem, componenten en omgeving; object modellen; data modellen;
- **Systeemevolutie** – geanticipeerde wijzigingen
- **Appendices**
- **Index**

Requirement beschrijving

- Beschrijf alleen extern gedrag
- Meetbaar, testbaar
- Rationale
- Bron

RE lijkt zo eenvoudig... maar is moeilijk!

- Vraagt heel veel communicatie
- Mis-interpretaties
- Dubbelzinnigheid

“I know you believe you understood what you think I said, but I am not sure you realize that what you heard is not what I meant...”

Moeilijk!

- **Scope** – vage afbakening van grenzen, of te veel detail
- **Begrip**
 - Uiteindelijk oplossing is moeilijk voor te stellen
 - Taalverschil gebruiker - ontwikkelaar
 - Huidig vs toekomstig systeem ('vastgeroeste' eisen en wensen)
 - Denken in behoeften versus oplossingen
 - "Voor-de-hand-liggende" zaken worden verzwegen
- **Volledigheid**
- **'Wicked problems'** – geen eenduidige oplossing
- **Vluchtigheid** – requirements veranderen in de tijd
- **Conflicterende eisen** bij gebruikers
- Verschil tussen **opdrachtgever** en **gebruiker** (bijv. budget)
- Verschil tussen **'nice-to-have'** en **kritische** functionaliteit

Valkuil

- **Software-centrisch** perspectief

Neem alle elementen van een systeem in ogenschouw voordat je je op de software gaat concentreren.

- Mensen
- Procedures
- Bedrijfsdoelen

Van geheel naar detail

- **Begin** met **begrip** opbouwen van het bedrijf of organisatie als **geheel**, en **zoom** dan **in op** bepaalde **delen** ervan

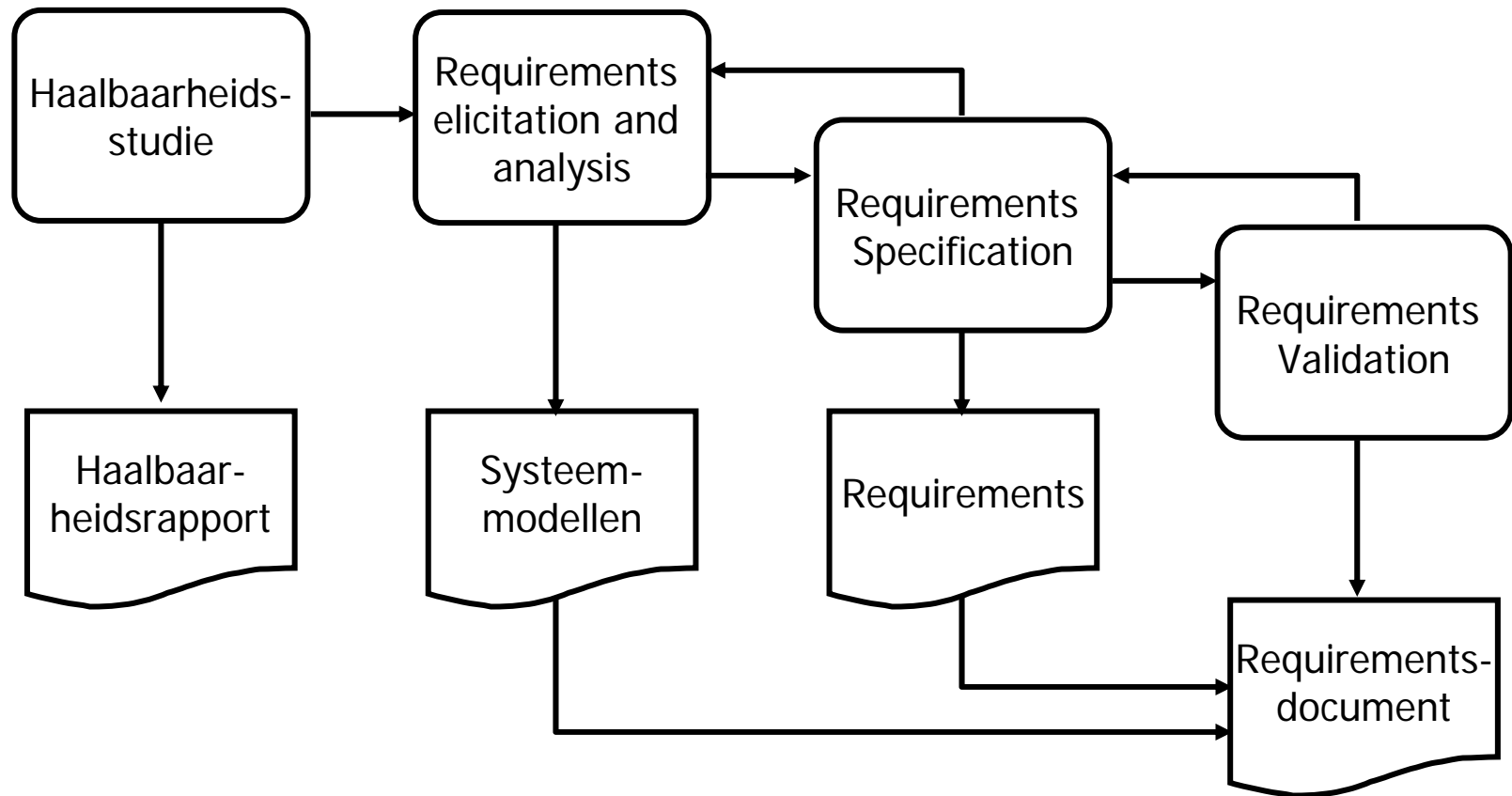
Kritische succesfactoren

- Beoordeel business en technische **haalbaarheid**
- Identificeer **stakeholders**, en verzeker je van hun commitment. Bijv. opdrachtgever, gebruikers(-vertegenwoordigers)
- Identificeer de personen die requirements kunnen vaststellen (met **mandaat** én **kennis**)
- Betrek zoveel mogelijk verschillende **partijen** in het proces
- Definieer **technische omgeving** (architectuur, operating system, netwerkomgeving)
- Identificeer '**domain constraints**' die de functionaliteit en performance van het systeem beïnvloeden
- **Kies methoden** die gebruikt worden om de requirements boven water te krijgen (interviews, workshops, prototyping)
- Documenteer de **rationale** achter elk requirement

Requirements engineering

- Het systematisch gebruik van bewezen principes, technieken, talen en gereedschappen voor een kosten-effectieve analyse, documentatie, en voortschrijdende evolutie van gebruikersbehoeften en de specificatie van het gedrag van een systeem afgestemd op die behoeften.
- **Focus** op **wat?**, niet op *hoe?*

Requirements engineering – het proces



Haalbaarheidsstudie

Beantwoord in ieder geval de volgende vragen:

- Draagt het systeem bij aan de **doelstellingen** van het bedrijf of de organisatie?
- Kan het systeem worden geïmplementeerd met de beoogde technologie en binnen het opgegeven **budget** en **tijdsframe**?
- Kan het systeem worden **geïntegreerd** binnen de bestaande omgeving?

Haalbaarheidsstudie

Vragen die ook kunnen worden gesteld:

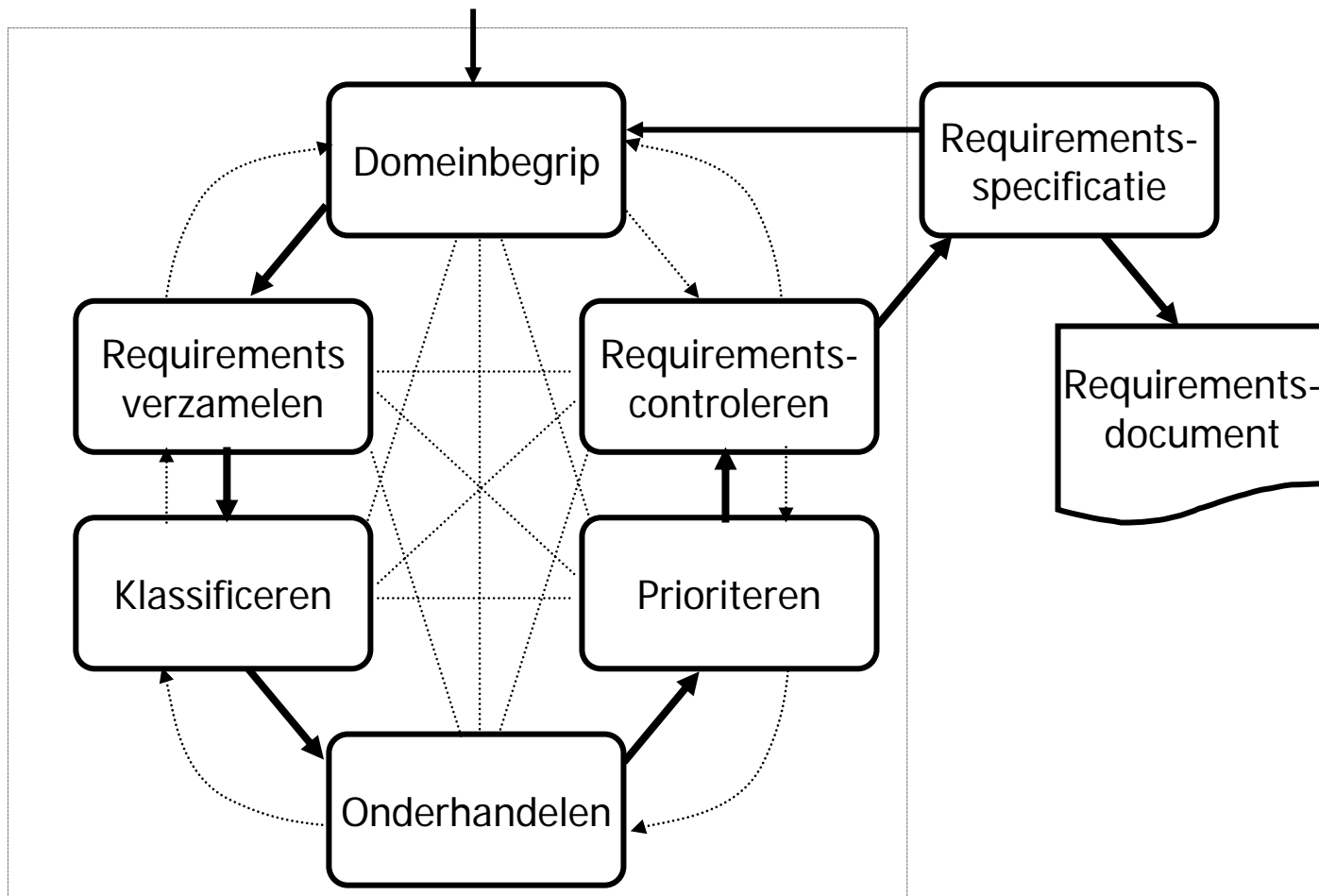
- Hoe redt de organisatie het **zonder** het systeem?
- Wat zijn de **problemen** die het systeem op moet lossen?
- Kan de benodigde informatie uit de **bestaande systemen** worden **overgedragen** naar het nieuwe systeem?
- Moet er gebruik worden gemaakt van **technologie** die **nog niet eerder** in de organisatie is **toegepast**?
- **Wat** moet er **wel** en **wat niet** worden ondersteund door het systeem?

Requirements elicitation

Het **boven water krijgen** van de eisen, wensen en behoeften t.a.v. het systeem

- Functioneel
- Niet-functioneel

Requirementselicitatie en -analyse



Requirements-**analyse**

- **Consistentie** met algemene doelstelling
- Juiste niveau van **abstractie**
- Maak onderscheid tussen **essentiele** requirements en **nice-to-haves**
- **Rationale** en **bron** (persoon) van requirements
- **Conflicterende** requirements
- **Haalbaarheid** binnen de technische context
- **Testbaarheid** van requirements
- **Ondubbelzinnig**
- **SMART**
(**S**pecifiek, **M**eetbaar, **A**ceptabel, **R**ealistisch, **T**ijdgebonden)

Onderhandelen

- Bij conflicterende requirements moet worden onderhandeld
- Ga op zoek naar achterliggende belangen
- Onderscheid essentiële zaken en implementatiedetails
- Stel prioritering op...

Prioritering

- Zorg dat je van alle requirements weet welk belang erachter zit; hoe belangrijk ze zijn
- **MoSCoW**
 - M**ust have,
 - S**hould have,
 - C**ould have,
 - W**on't have

Requirements-**validatie**

- Zijn de requirements
 - ‘Smart’, consistent, etc?
- Methoden:
 - Formele technische reviews
 - Prototyping
 - Test-cases genereren
 - Automatische consistentie-analyse

Systemmodelling

- Informatiemodelling
- Data-flow
- Gedrag
- Use-cases

Use-cases

- **Scenario's** over typische interacties met het systeem beschreven vanuit actor-perspectief
- **Actor** is bijv. gebruiker of device
- Beschrijft
 - **Taken/functies** van een actor
 - **Informatie** die wordt verkregen, gegeven, of gewijzigd door een actor
 - **Toestandsveranderingen** van het systeem
 - (Onverwachte) **events** waarvan de actor op de hoogte moet zijn

Requirements engineering – een **sociaal** leerproces

- Ga een relatie aan met de klant
- Wees je bewust van wij/zij-denken
- Creëer een sfeer van samenwerken

De eerste **verkenning**

- Algemene vragen
 - Wie heeft de opdrachtomschrijving opgesteld?
 - Wie zal de software gaan gebruiken?
 - Wat zijn de economische voordelen van een succesvol systeem?
- Inzoomen; detailvragen
 - Hoe ziet 'goede' output eruit?
 - Welke problemen lost het op?
 - Hoe ziet de omgeving eruit?
 - Performance? Randvoorwaarden?

De eerste **verkenning**

- Meta-vragen
 - Juiste persoon? Hoe 'officieel' zijn je antwoorden?
 - Zijn mijn vragen relevant?
 - Stel ik teveel vragen?
 - Zijn er vragen die ik nog zou moeten stellen?
 - Zijn er nog anderen die zinvolle informatie hebben?
 - Wat is essentieel aan een requirement?

Workshops

- Vermijd vraag-antwoordformat (interviews), behalve evt. bij een eerste contact.
- Zijn de juiste personen aanwezig?
- Faciliteer
 - samen probleemoplossen,
 - samen onderhandelen,
 - samen specificeren
- Brainstorm, creatief proces
- Maak de discussie zichtbaar (flapover, post-its, diagrammen, ...)

Requirements-documentatie

- Beschrijft de systeembehoefte en de haalbaarheid;
- Begrenzing, scope;
- Een lijst met stakeholders (klant, gebruikers), die geholpen hebben om de requirements boven water te krijgen;
- De technische omgeving van het systeem;
- Een lijst met functionele en niet-functionele requirements;
- Een verzameling gebruiksscenario's (use-cases);
- Beschrijving van gebruikte prototypes

Traceerbaarheid

- De relatie tussen **requirements** en **bedrijfsdoelstellingen** dient helder te zijn.
- De relatie tussen **ontwerpbeslissingen** en **requirements** dient helder te zijn. Requirements vormen het waarom voor ontwerpbeslissingen.

Werkstuk 2 (van de 3):

- Lees hoofdstuk 7 van het boek.
- Maak requirementsdocumentatie zoals hiervoor beschreven (ongeveer 20 bladzijden).
 - Ook voor de klant leesbaar
 - Let hierbij ook op de traceerbaarheid.
- E-mail **streefdatum maandag 5 maart** met als subject “[SE] werkstukopgave2” naar marko@cs.ru.nl
- Per groep 1 email, vermeld groepsnaam!