

Software Engineering (100094)

Marko van Eekelen

marko@cs.ru.nl

kamer HG02.074

Overzicht

- Theorie: hoorcollege (2 icts)
- Practicum: GiPHouse (7 icts)

Beoordeling

- Practicum (groepswerk)

1. projectbeoordeling

- a. Klanttevredenheidsrapportage (via manager),
- b. **projectdocumentatie**,
- c. **testrapportage**,
- d. **projectpresentatie**,
- e. projectevaluatie (manager): product, proces, samenwerking

2. **werkstukken**beoordeling

3. globale indruk

- a. indruk van de docent
- b. indruk van de gip-directie (via de gip-directie)
- c. **groepsprojectevaluatie**
- d. **eigen projectevaluatie**

- Tentamen (schriftelijk, individueel)

Eindbeoordeling

Als beide onderdelen (practicum + tentamen) **voldoende** beoordeeld zijn,

dan is het eindresultaat het **maximum** van enerzijds het **naar studiepunten gewogen gemiddelde** en anderzijds het **ongewogen gemiddelde** van de twee;

zo niet dan geldt als eindresultaat de **laagste beoordeling van de twee**.

Bij de weging geldt:

7 ec voor het practicum en 2 ec voor het tentamen.

Boeken

- Verplicht:

*Software Engineering – A practitioner's Approach
European Adaptation, **sixth** edition*

Roger S. Pressman

ISBN 0-07-285318-2

- Ook interessant:

Software Engineering, sixth edition

Ian Sommerville

ISBN 0-201-39815-X

Hoorcollege

- Vorm:
 - Presentatie
 - Discussies
 - Werkstukjes
- Communicatie via blackboardemail
- Ook actuele informatie op

<http://www.cs.ru.nl/~marko/onderwijs/se>

Inhoud

1. 06 feb: Het systeemontwikkelp proces
2. 13 feb: Requirements-analyse
3. ... : Documentatie, Kwaliteit
4. ... : Architectuur, Object-oriëntatie
5. ... : Ontwerp
6. ... : Menselijke factoren
7. ... : Testen
8. ... : gastcollege?
9. ... : ...

College 1: Leerdoelen

- Hoofdstuk 1,2,3 van Pressman
- Reflectie betreffende definities van softwaresystemen
- Zicht op invloed van soort probleem op softwareontwikkeling
- Definitie Software Engineering
- Kennen van softwareprocesmodellen (CMM, Waterval, V-model, ...)

Wat is software?

- Software is :
 - Computerprogramma's, Executables
 - Assembler?
 - Byte Code?
 - Source code?
 - Bijbehorende documentatie?
 - Data ...?

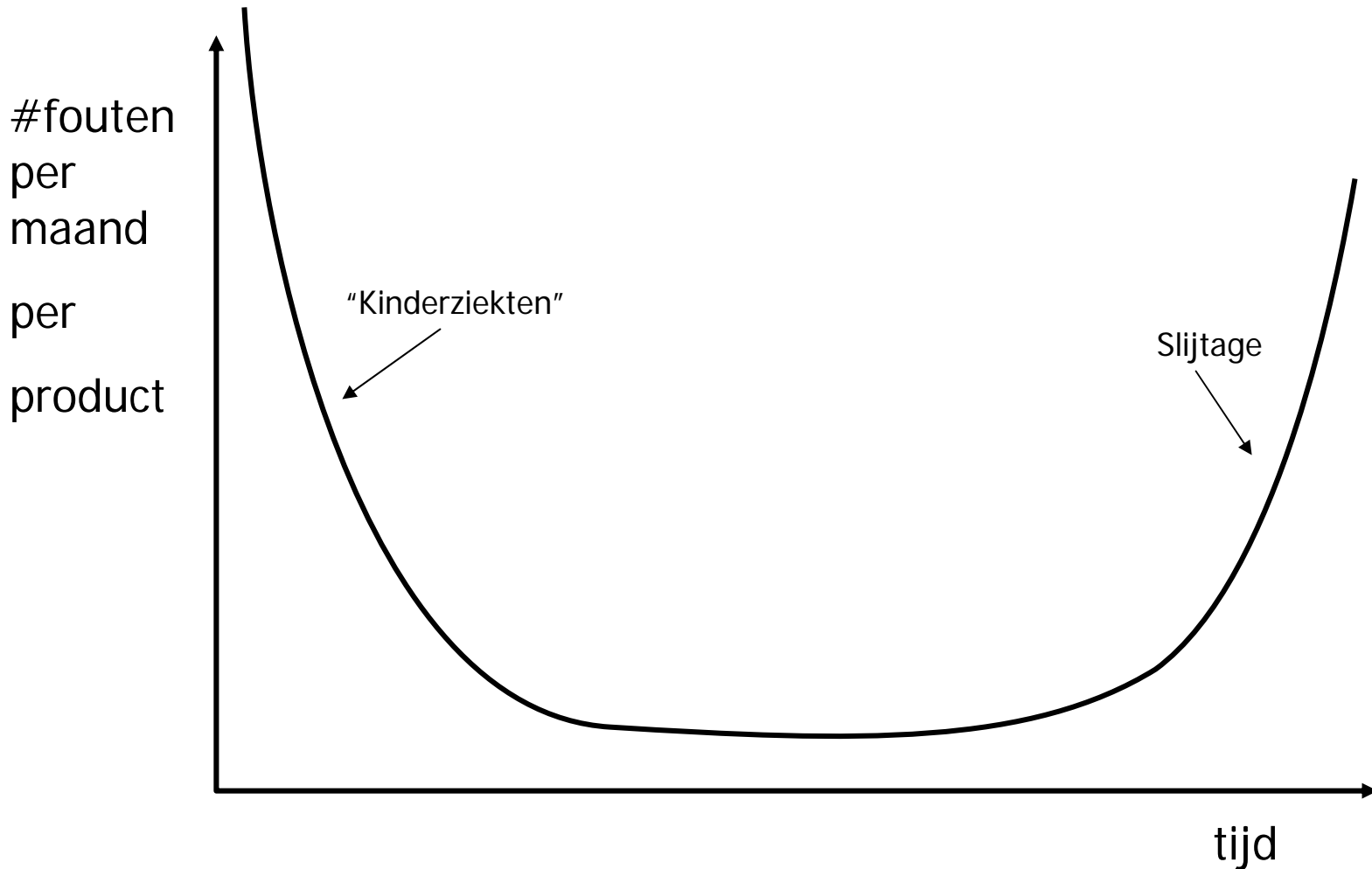
Wat is software?

- Software ...
 - wordt gebruikt
 - wordt onderhouden (veranderd)
- Software ...
 - representeert grote hoeveelheden bedrijfskennis
 - is bedrijfskapitaal
 - is gepatenteerd en gelicenseerd

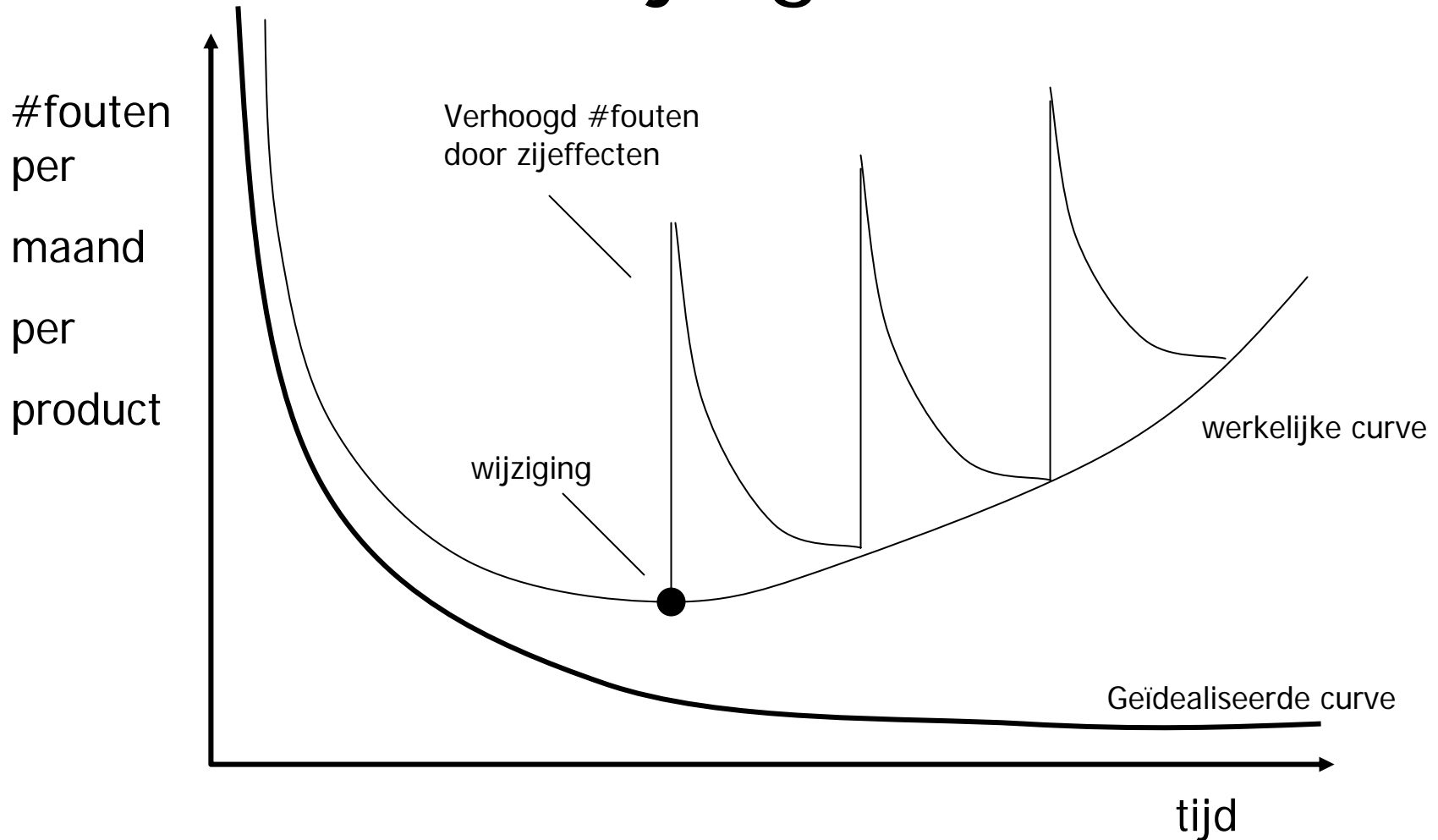
Waarin onderscheidt software zich van andere artefacten?

- Software wordt ontwikkeld; niet gefabriceerd
- Software is niet tastbaar
- Software is complex
- Slijt niet, maar veroudert wel

Normale slijtage: bedrijfsmatige productkosten



Software "slijtage"



Software-toepassingen

- Bedrijfstoepassingen
- Real-time systemen
- Embedded systemen
- Web-applicaties
- Ontwikkelgereedschappen
- Wetenschappelijke software
- Systeemsoftware
- Kunstmatige intelligentie / Kennisgebaseerde systemen
- ...

Software-ontwikkeling

- Twee soorten problemen:
 - Tamme problemen
 - Gemene (wicked) problemen

Tamme problemen

- Goed gedefiniëerd
- Stabiel
- Duidelijk wanneer klaar
- Oplossingen zijn eenduidig goed of fout
- Na een foute oplossing kun je het opnieuw proberen
- Zijn onderdeel van een probleemklasse
- Pasklare oplossing

De realiteit: Wicked problems

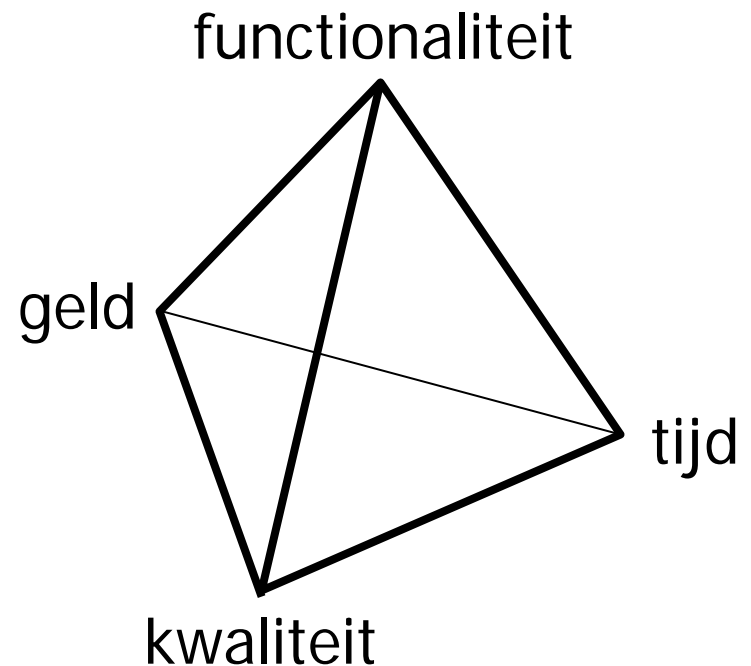
- Probleemdefinitie vaag; je snapt 't pas als je een oplossing hebt
- Veranderende omgeving (moving target)
- Onduidelijk wanneer klaar
- Goed of fout is relatief (klanttevredenheid)
- Uniek voor omgeving
- Omgeving verandert door project
- Geen pasklare oplossingen

Hoe ga je om met wicked problems?

- Het probleem temmen?
 - Vastpinnen, bevriezen
 - Simplificeren
 - Bestuderen
 - Tamme deelproblemen isoleren
- Doe wat je doen kunt, maar blijf je bewust van de 'wickedness'
- Zorg voor voortdurende afstemming

Software engineering

Hoe zet je de beschikbare middelen (**tijd** en **geld**) in om optimale **functionaliteit** en **kwaliteit** te behalen?



Software engineering

... is een **systematische**,
gedisciplineerde en **meetbare** aanpak
van de ontwikkeling, de uitvoering en het
beheer van software.

⇒ *Dit vraagt om een*
gedefiniëerd softwareproces

Wat is een softwareproces?

- Een georganiseerde verzameling activiteiten, methoden, tools gericht op het ontwikkelen, uitvoeren en onderhouden van software

Volwassen processen

- Key process area's
- CMM: Capability Maturity Model
Carnegie Mellon Software Engineering Institute (SEI)
<http://www.sei.cmu.edu/cmm/>

Key process area's

- Requirementsmanagement
- Projectplanning & -tracking
- Subcontract-management
- Configuratiemanagement
- Training
- Foutpreventie
- Innovatie
- Reviews
- Coördinatie tussen groepen
- ...

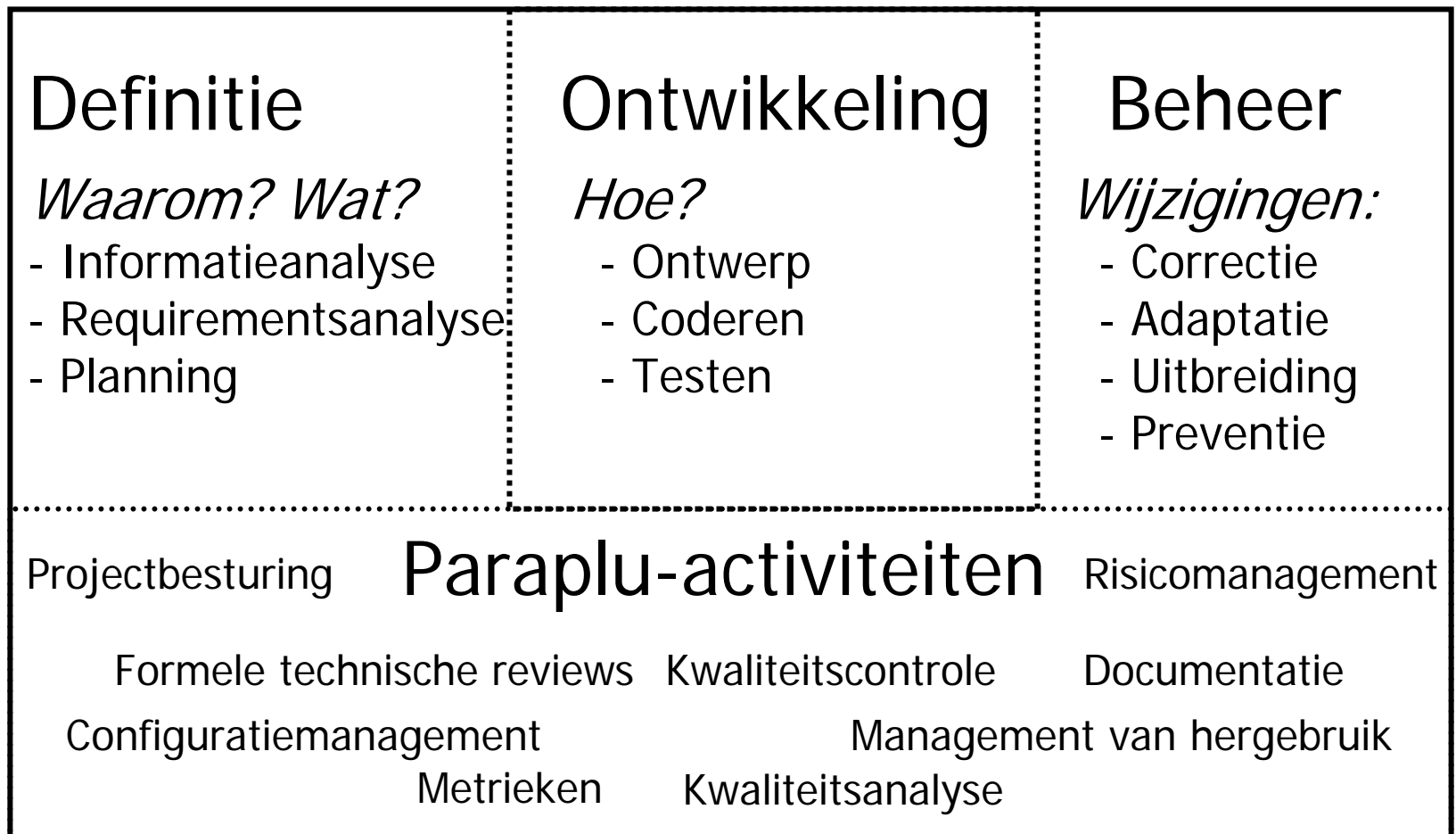
Key process area's

- Doelen
 - waarnaar wordt gestreefd in een KPA?
- Commitments
 - welke randvoorwaarden zijn er in de organisatie vervuld om de doelen te bereiken, en hoe leveren die een bewijs van de intentie om naar die doelen te streven?
- Voorzieningen
 - welke zijn getroffen om de organisatie in staat te stellen de doelen te bereiken?
- Activiteiten
 - welke zijn nodig om de KPA te laten functioneren?
- Monitoringsmethoden
 - hoe worden activiteiten bestuurd en beheerst?
- Verificatiemethoden
 - hoe wordt gecontroleerd of een KPA naar behoren wordt uitgevoerd?

Capability Maturity Model

Level	Focus	Key Process Area's
5 Optimizing	Continue proces-verbetering	procesmatig veranderingsmanagement, technologische innovatie, foutpreventie
4 Managed	Product- en proceskwaliteit	kwaliteitsmanagement, kwantitatief procesmanagement
3 Defined	Engineering-proces	peer reviews, coördinatie tussen groepen, software product engineering, geïntegreerd software management, trainingsprogramma, organisatieproces definitie/focus
2 Repeatable	Project-management	Software configuratie management, Software quality assurance, Software subcontract management, Software project tracking, Software project planning, Requirements management
1 Initial	Helden	

Gefaseerde technologie



Softwareprocesmodellen

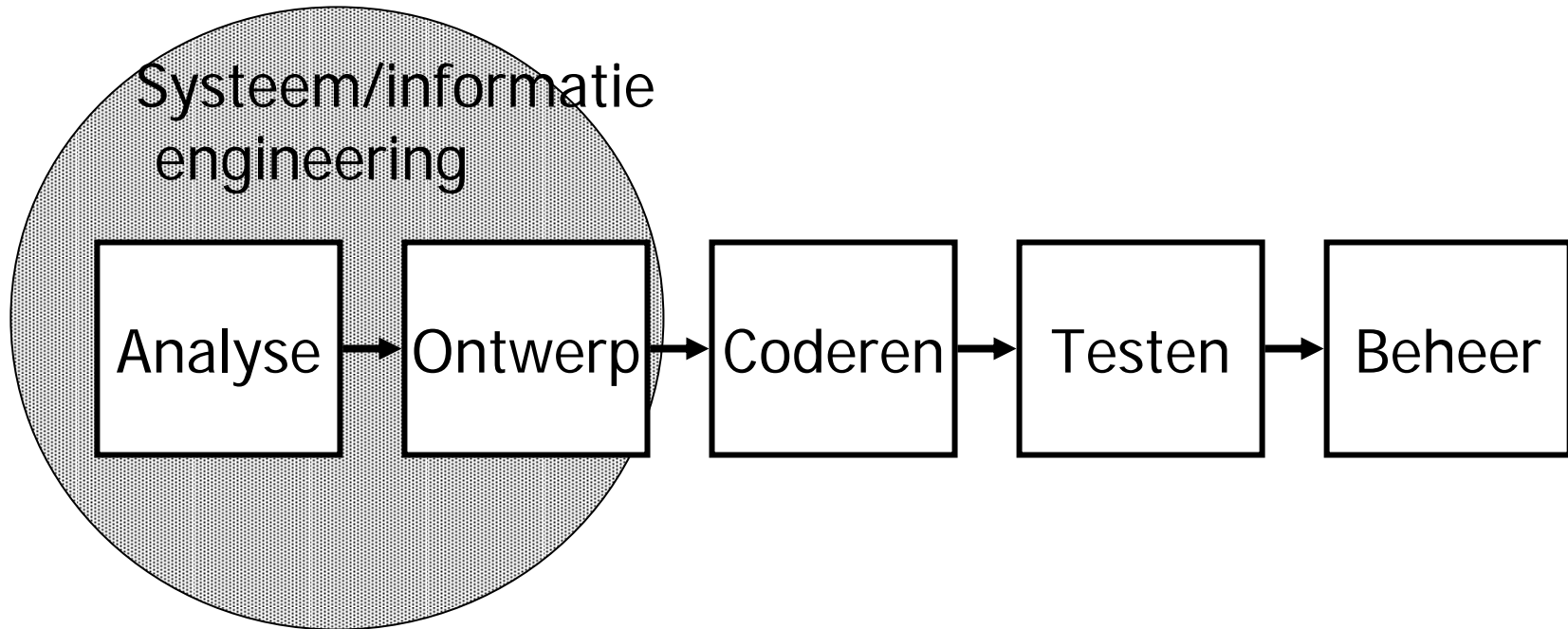
- Versimpelde representatie van een software-proces, vanuit een specifiek perspectief
 - Workflow - sequentie van activiteiten
 - Dataflow - stroom van informatie
 - Role/action - wie doet wat?
- Generieke procesmodellen
 - Waterval
 - Evolutionaire ontwikkeling
 - Formele transformaties
 - Integratie van herbruikbare componenten

Software procesmodellen

- **Lineaire ontwikkeling (klassiek, waterval)**
- **V-model**
- **Prototyping model**
- **Spiraalmodel**

- Rapid Application Development (RAD)
- Evolutionaire modellen
- Concurrente ontwikkeling
- Component-gebaseerde ontwikkeling
- Formele methoden
- 4GL
- Agile Development
- Open Source
- Extreme Programming
- ...

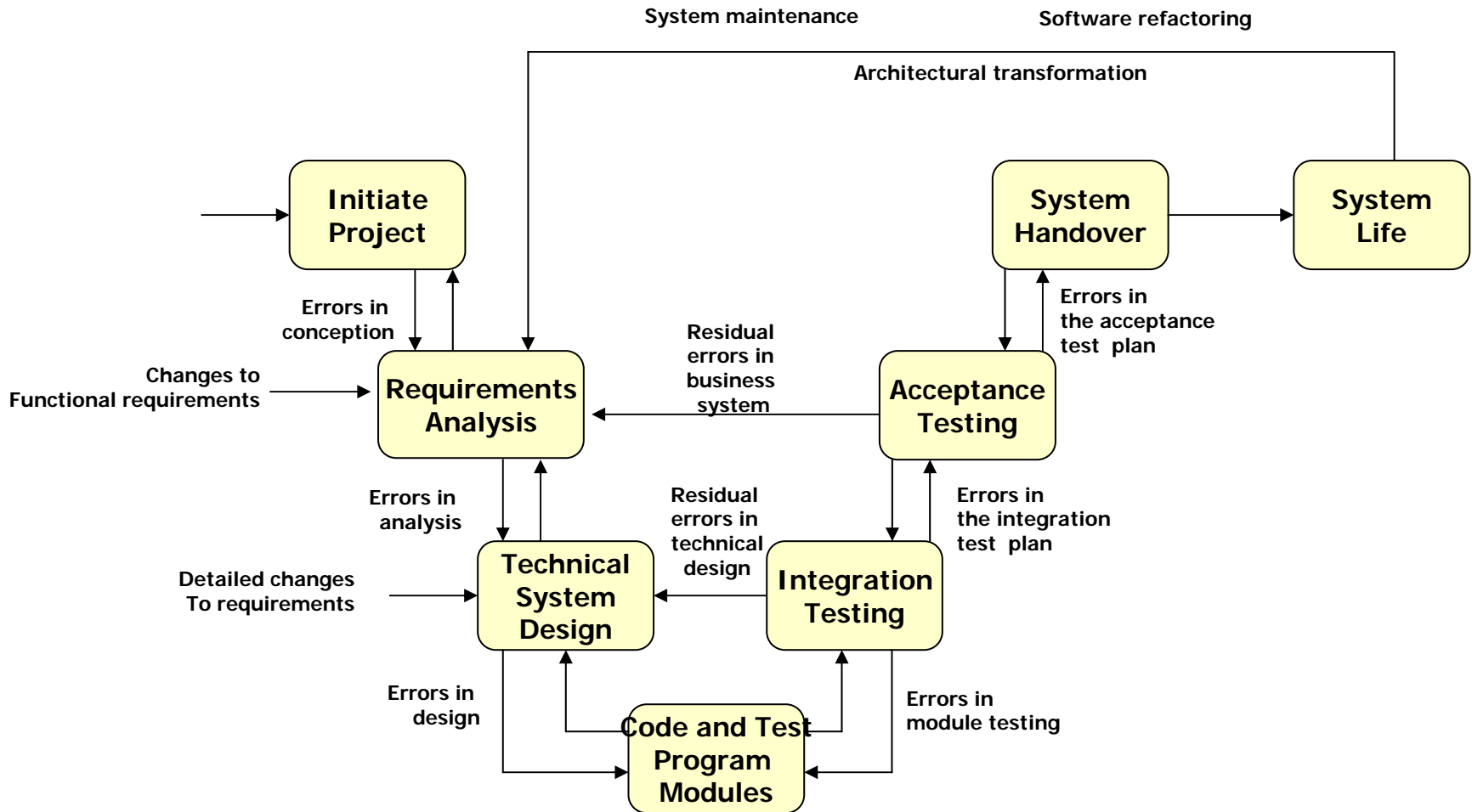
Lineaire ontwikkeling (waterval)



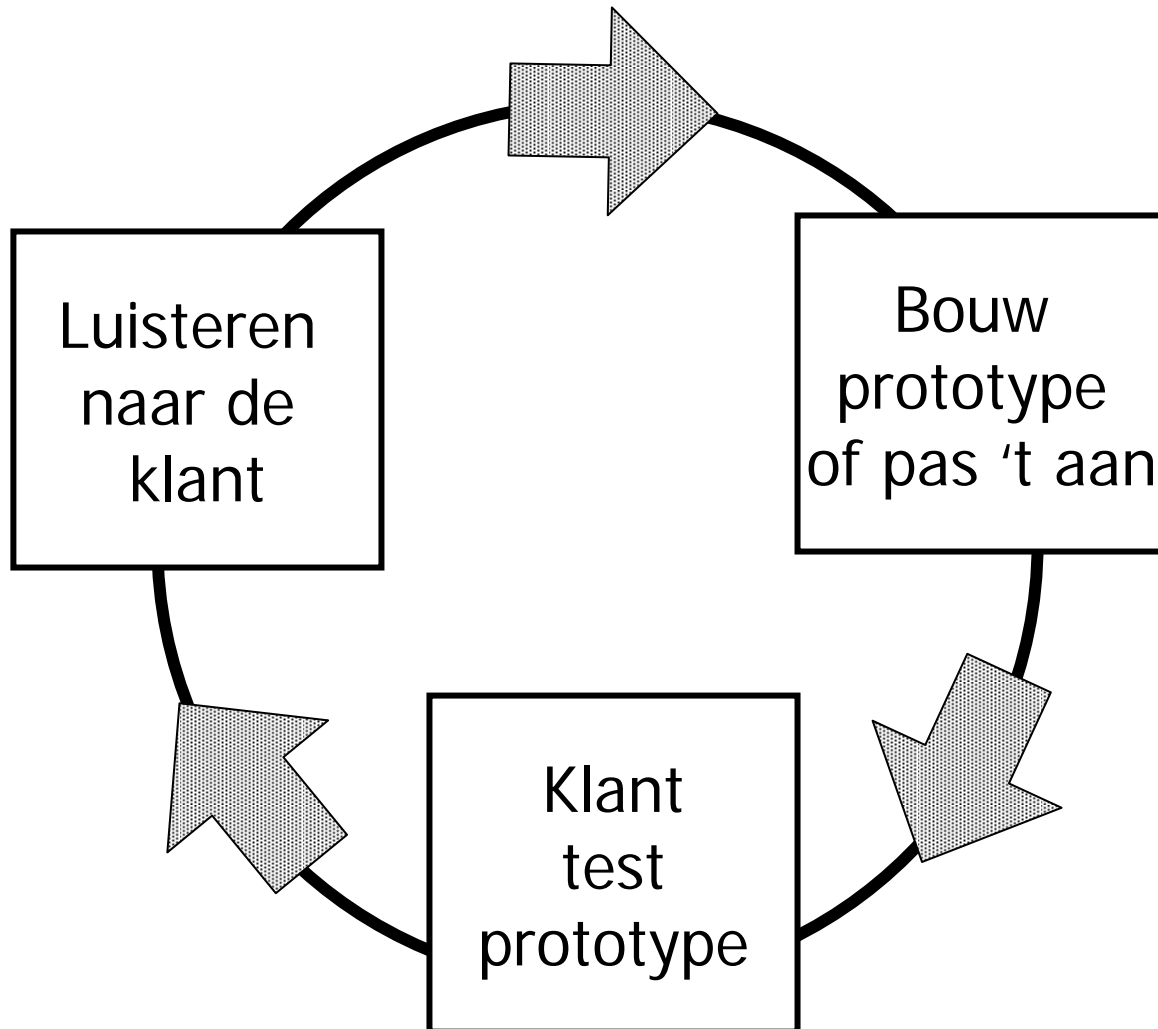
Fasen

- Analyse/Specificatie; achterhalen wat het systeem moet doen en wat de randvoorwaarden zijn
- Ontwerp; uitdetaillering van specificatie + vormgeving
- Coderen; programmeren, hoe moet de computer dit doen
- Testen/Valideren; controleren of de software werkt, en voldoet aan de verwachtingen van de klant
- Beheer/Onderhoud; wijzigen van de software ten behoeve van veranderende eisen uit een veranderende omgeving

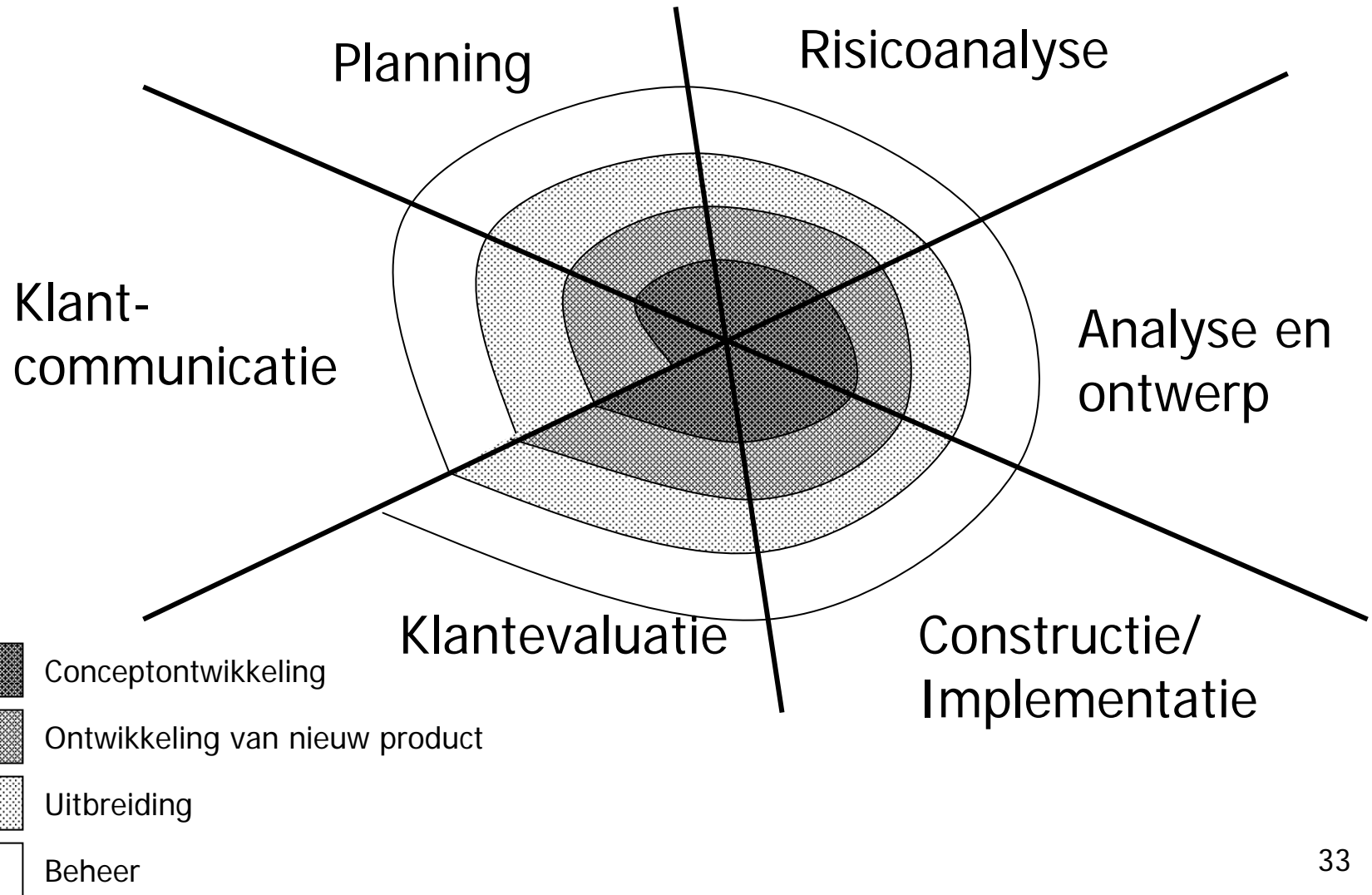
System's Lifecycle V-Model



Prototyping model



Spiraalmodel



Rationele benadering

Eerst ...

- **Waarom?** Als je die vraag niet kunt beantwoorden begin er dan niet aan.
- **Wie** worden er blij van? ... weten die dat ook? Wie begint er om zich heen te slaan als het project stagneert? (*stakeholders*)

En dan pas ...

- **wat** moet het systeem allemaal doen?
hoe pakken we dat aan? etc.

Rationele benadering ⁽²⁾

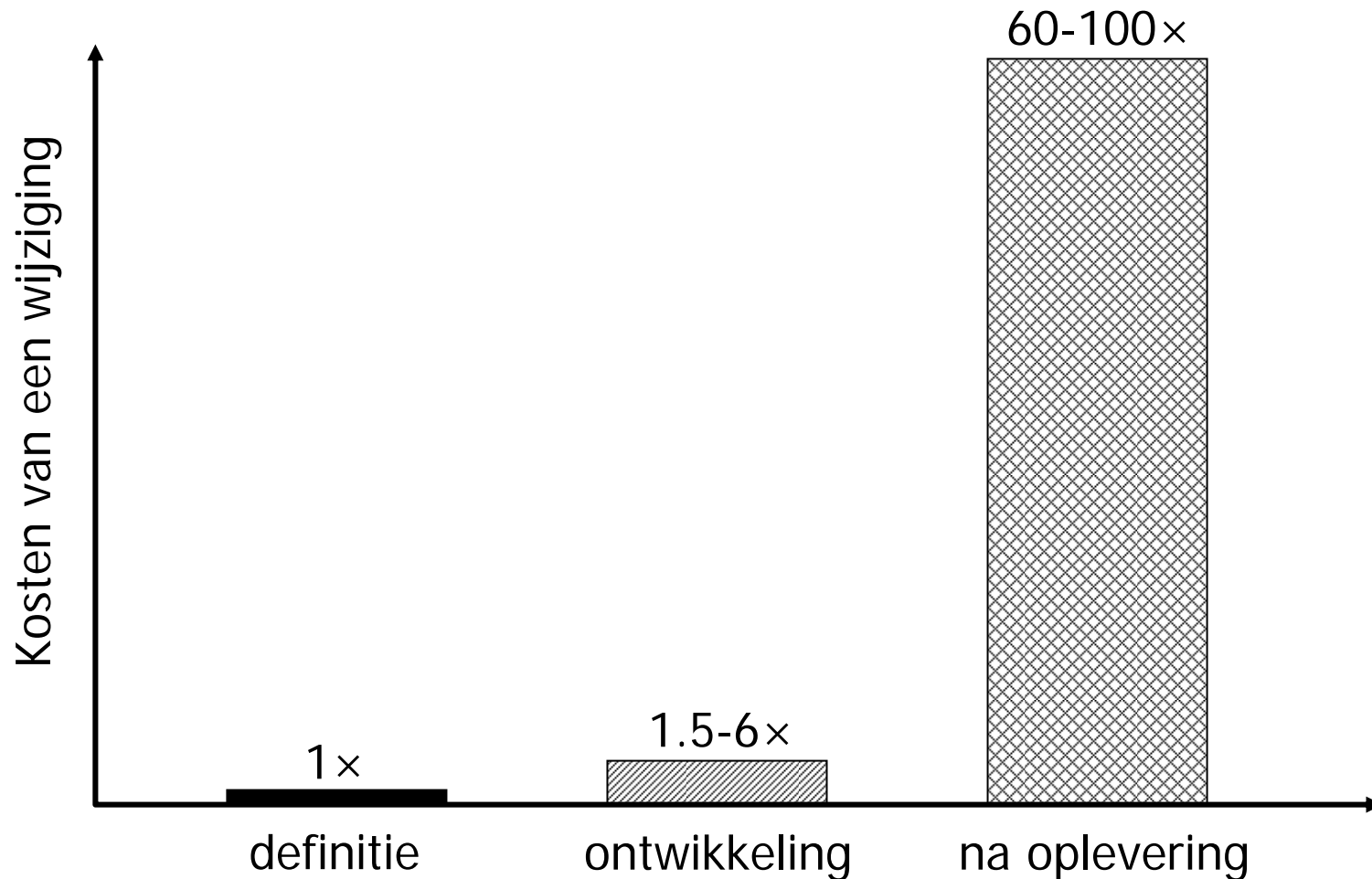
Zorg dat je weet (of beter nog: documenteert)
waarom je iets doet

- Doelen
- Alternatieven
- Voor- en nadelen
- Aannamen
- Risico's
- Wat is essentieel?
- Don't cares

Kosten van software

- 15% ontwikkelkosten
- 10% testen
- 75% onderhoud
- De verdeling van de kosten over de verschillende ontwikkelingsfasen is sterk afhankelijk van het model dat wordt gebruikt

Kosten van wijzigingen



Mythen

- Onze werkwijze staat uitgebreid beschreven, dus hebben we een goed gedefiniëerd proces.
- Als we in tijdsnood komen, trekken we even een blik programmeurs open.
- Een algemene beschrijving van de projectdoelen is voldoende om met de constructie te beginnen, de details komen later wel.
- Om te kunnen weten of 't goed is moeten we eerst zorgen dat 't draait.
- Requirements veranderen voortdurend, maar gelukkig is software flexibel.

Voor de volgende keer:

- Lees hoofdstuk 1,2,3 van het boek
- Bereid je eerste gesprek met je GiPHouse-klant voor.
 - Wat moet jij van de klant weten?
 - Wat moet de klant van jou weten?
 - Maak een lijst met vragen van (1 A4-tje) die je kunt stellen aan je opdrachtgever
 - Denk ook na over, en schrijf iets op over
 - waarom je een vraag stelt, en
 - of je geen belangrijke zaken over het hoofd ziet.
- E-mail **uiterlijk maandag 12 februari** met als subject “[SE] werkstukopgave1” naar marko@cs.ru.nl
- Per groep 1 email, vermeld groepsnaam!