

# Requirements in een klein project - Voor een team van 1

Gastcollege  
Technische Universiteit  
Eindhoven

Harry Nieboer



# Insteek vandaag

- Requirements in de praktijk
  - Eerst Requirements in een klein project (één software engineer in project van één week)
  - Daarna Requirements in een groot project
    - Requirements met Endeavour
    - Aansluiting van Requirements op Architectuur, Bouw en Test
  - Vragen gewoon tussendoor
  - Op het einde tijd voor discussie

# Een Software Ontwikkelroces

Klinkt als:

- Grote rijen ordners met stoffige procedures
- Een wirwar van nutteloze formulieren
- Allemaal extra werk
- Iets voor grote bedrijven en mega projecten
  
- ... maar niet voor een klein project
- ... en zeker niet voor mij

# Een Software Ontwikkelproces



- Wordt meestal uitgelegd aan de hand van grote projecten
- Laten wij eens kijken naar een klein projekt
  - **1 Software engineer**
  - **1 Applicatie(tje)**
  - **1 Week**

# Bronvermelding:



## A Software Development Process for a Team of One

by [Philippe Kruchten](#)


Rational Fellow

*For some, the phrase "software engineering process" evokes an image of a huge set of dusty binders full of policies, directives, and forms, all saturated with administrative jargon. But, these are materials that would probably be used only by very large companies that deliver software at a snail's pace to government agencies and Fortune 500 companies -- software developed by armies of programmers aligned in giant cubicle farms and herded by "pointy-haired managers," like the one in the famous Dilbert cartoons by Scott Adams.*

*In reality, however, a software engineering process does not need to be such a monster. It can be as lightweight or heavyweight as the job at hand and the size of the development organization requires. Whether the project is a 200-developer mastodon or a short, solo gig, a good process can be tailored to fit the job.*



# Agenda

- Korte beschrijving van het probleem
- Wat gebeurde er van dag tot dag
  - Van het eerste contact
  - Tot en met de oplevering
- Tijdens de presentatie
  - Schrijf op wat je opvalt
  - Zowel positief als negatief
- Daarna korte bespreking
- En wat heeft dit te maken met  rocessen



- Opdrachtnemer **Nick**
- Software Engineer
- 12 jaar ervaring
- Werkt het liefst alleen
- Heeft zo z'n eigen werkwijze
- Heeft dagboek bijgehouden van projekt
  
- Opdrachtgever **Gary**
- Manager klein accountantsbedrijf




- Het bedrijf van Gary
- 10 accountants
- Werken voor klanten
- Moeten bijhouden wat ze voor welke klant doen ...
- ... en hoelang het duurt
- (100 Euro per uur)

## Probleem:

- Uren bijgehouden op post-it briefjes, in Excel, in planners
- Toegezonden in mail, op papier, op de voice-mail, op diskettes
- Heidens karwei om het allemaal bij elkaar te vegan



# Agenda

- Korte beschrijving van het probleem
- Harry: wat gebeurde er van dag tot dag
  - **Van het eerste contact**
  - **Tot en met de oplevering**
- Jullie: wat valt je op
  - **Schrijf op/onthoud wat je opvalt**
  - **Zowel positief als negatief**
- Daarna korte bespreking
- En wat heeft dit te maken met  rocessen

# Wat valt je op

## Hoe loopt het voor Nick (opdrachtnemer)

- Positief

- ...

- ...

- ...

- Negatief

- ...

- ...

- ...

## Hoe loopt het voor Gary (opdrachtgever)

- Positief

- ...

- ...

- ...

- Negatief

- ...

- ...

- ...





~~Paar uur ?~~

**Paar dagen !**

Maandag een voorstel

Eerste idee van Nick:

- Iedereen krijgt klein timertje op het scherm
- Instellen welke activiteit nu wordt uitgevoerd
- Starten en stoppen
- Onderbreken en doorgaan
- Data wordt bewaard
- Totalen wekelijks in spreadsheet

## Duidelijkheid over:

- Wat wil Gary bereiken?  
Wat is de oplossing van Nick?
- Wat wordt wanneer gedaan?
- Wat kan er mis gaan?
- Wat levert het op?  
Voor Gary, voor Nick



## De 4 A4-tjes aanpak

(Best Practice van Nick)

**Visie**

**Planning**

**Risico's**

**Business  
Case**

## Visie

- **Wat is het probleem:**

Voor Gary's organisatie,

omdat het moeilijk is consistente gegevens te vergaren over bestede tijd

is het moeilijk voortgang te meten, klanten te factureren, schattingen voor nieuwe projecten af te geven

## Visie

- **Wat is de oplossing voor dit probleem:**

Personal Timer Tool (PTT),

- meet tijd
- verzamelt en bewaart gegevens
- maakt inzichtelijk waar tijd aan besteed wordt
- gemeten en voorspelde waarden
- helpt bij maken planningen

## Visie

- **Wie zijn er bij betrokken:**

Accountants

Administratieve assistent

Project managers, planners

## Visie

- **Welke taken ondersteunen:**

Meet tijd voor activiteit

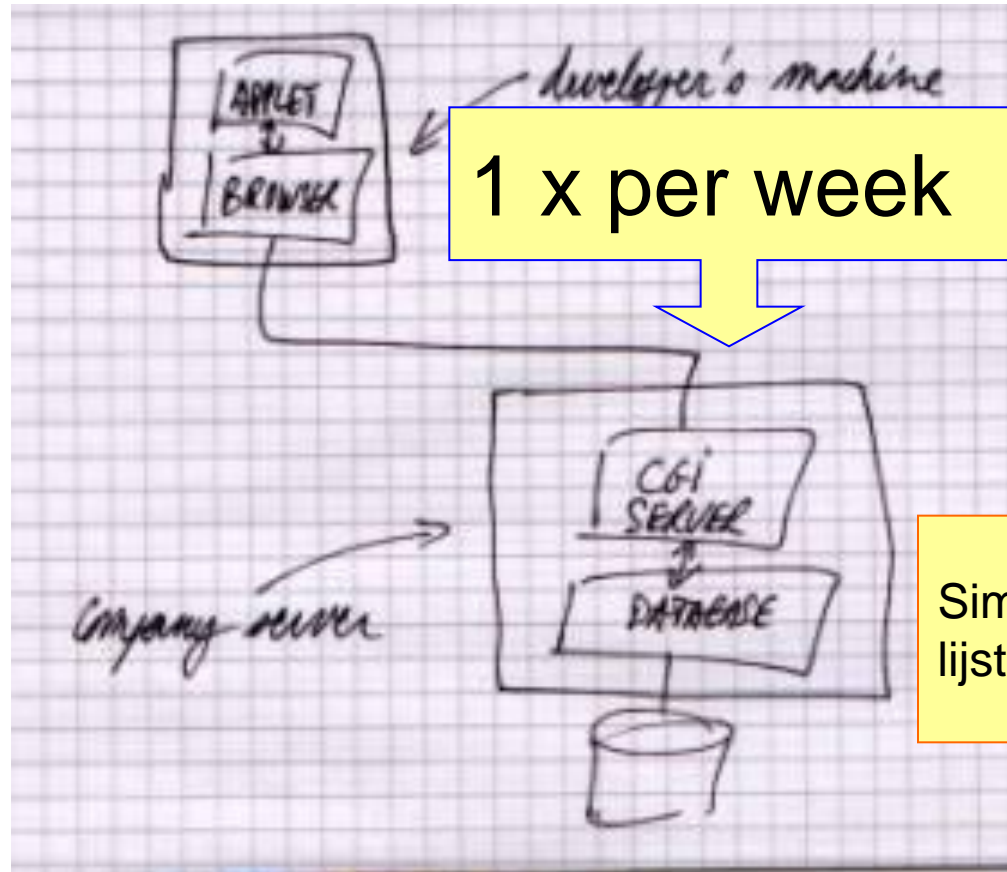
Genereer wekelijkse tijdstaat

Verzamel uren voor een projekt

Configureer tool en lijstmanager voor een projekt

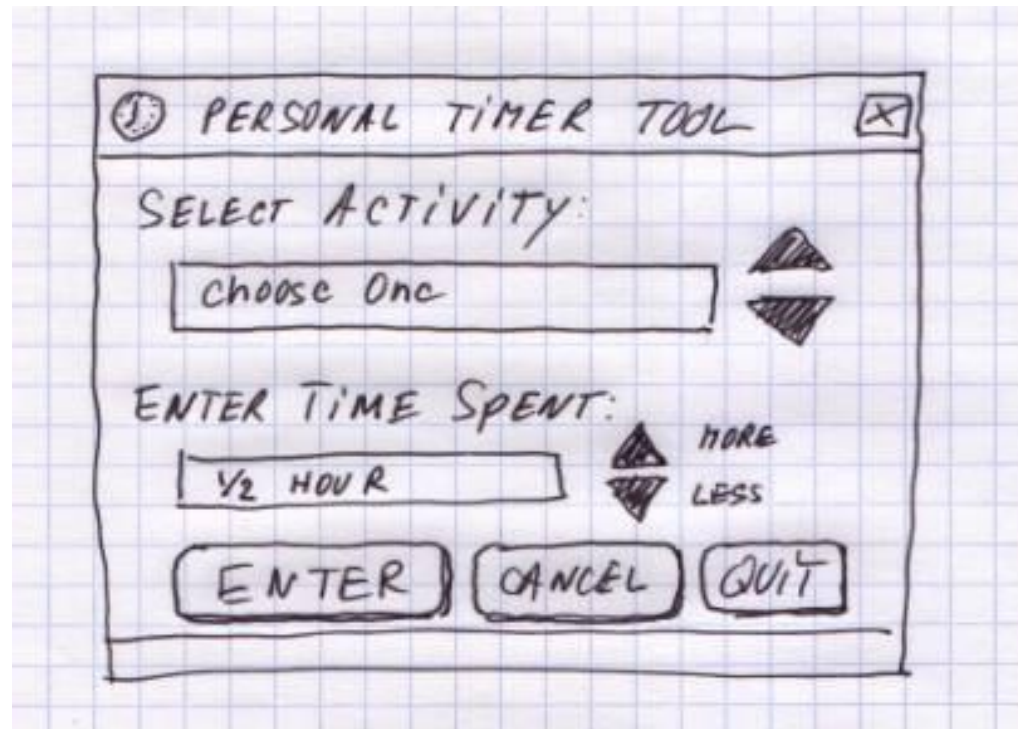
## Visie

- De oplossing in grote lijnen:



## Visie

- De oplossing in grote lijnen:



## De 4 A4-tjes aanpak

(Best Practice van Nick)

Visie

Planning

Risico's

Business  
Case

# De planning

maandag 22 november	dinsdag 23 november	woensdag 24 november	donderdag 25 november	vrijdag 26 november
Het voorstel	Prototype	Constructie (ontwerpen, bouwen, testen)	Constructie (ontwerpen, bouwen, testen)	
- Visie				
- Plan				
-Business Case	Risico's uitzoeken			
-Risico's				
Voorstel bespreken met Gary	Prototype bespreken met Gary		Eerste bêta presenteren	
Prototype	Ondersteunde taken uitwerken	Constructie (ontwerpen, bouwen, testen)	Afronden (verbeteringen?)	
	Testen bedenken			
			Opleveren	

## Stap 1: Het voorstel afstemmen

- Nu alles met Gary bespreken
- Controleren of beeld Nick = beeld Gary
- Afspreken dat Gary betaalt tot en met prototype
- Daarna besluiten of verder (GO/NOGO)

## Stap 2: De risico's uitzoeken

- Zeker weten dat ik alles heb
- Zeker weten dat ik niets onderschat
- Gaat het allemaal wel werken? Constructietesten!
- Als risico's uitgezocht, dan betere planning
- Prototype laten zien om feedback te krijgen
- Uitproberen door Gary en Eric (Accountant)

## Stap 3: Constructie en oplevering

- Ontwerpen, bouwen, testen
- Installatie eerste bêta op laptop Gary
- Daarna uitrol op alle werkplekken



## De 4 A4-tjes aanpak

(Best Practice van Nick)

Visie

Planning

Risico's

Business  
Case

# De Risico lijst

- Licentie voor ontwikkeltools is verlopen
- Database is te duur
- Werkt het mechanisme voor communicatie tussen computers wel bij Gary?
- Zitten alle machines bij Gary wel op het netwerk?



## De 4 A4-tjes aanpak

(Best Practice van Nick)

Visie

Planning

Risico's

Business  
Case

# Wat kost het en wat levert het op

## Business Case voor Nick

- Kost vier dagen = 4000 Euro
- Kosten voor nieuwe licentie 1000 Euro

## Business Case voor Gary

- Bespaart half uur per week per medewerker
- Bespaart twee uur per week voor administratie
- Totaal 600 Euro per week, 2500 per maand
  
- Programma verdient zich in twee maanden terug

De 4 A4-tjes aanpak

(Best Practice van Nick)

Visie

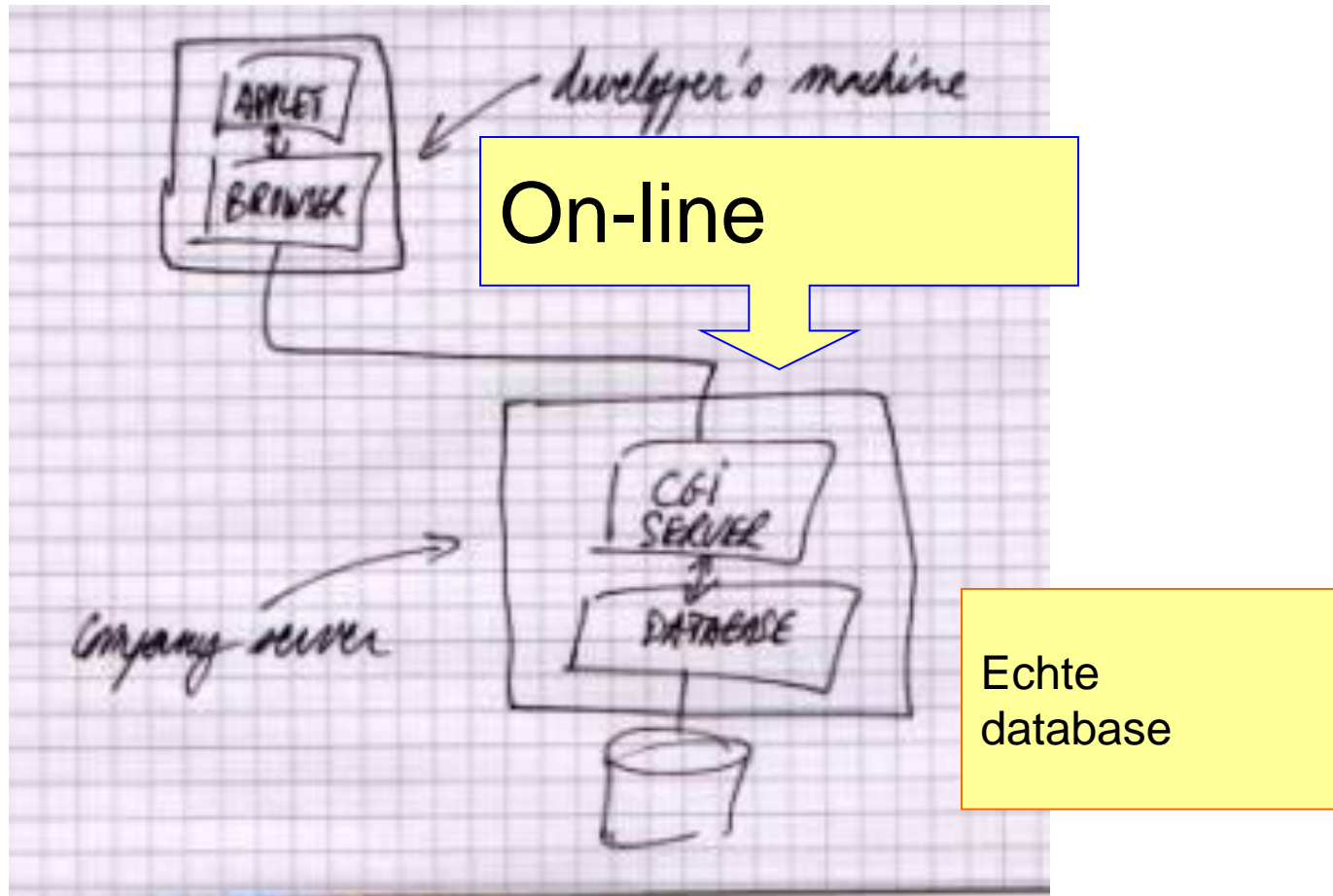
Planning

Risico's

Business  
Case

- Slecht nieuws
  - Gary wil alle data direct in 1 database
    - dus niet eenmaal in de week een overzicht
    - dus online eisen en database met beheerder
  - Medewerkers werken niet altijd op dezelfde machine
    - dus starten en stoppen op verschillende machines

# Maandagmiddag: bijgestelde voorstel van Gary



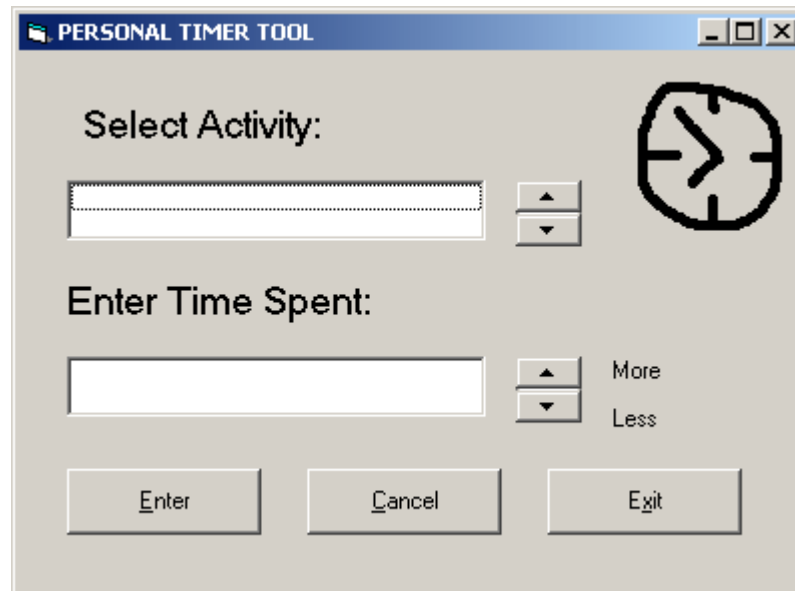
# Maandagmiddag: de bijgestelde planning

maandag 22 november	dinsdag 23 november	woensdag 24 november	donderdag 25 november	vrijdag 26 november
Het voorstel	Prototype	Constructie (ontwerpen, bouwen, testen) [Single person versie]	Constructie (ontwerpen, bouwen, testen) [Client-server versie]	Eerste bèta presenteren
- Visie				
- Plan				
-Business Case	Risico's uitzoeken			Afronden (verbeteringen?)
-Risico's				
Voorstel bespreken met Gary	Prototype bespreken met Gary			
Prototype	Ondersteunde taken uitwerken	Constructie (ontwerpen, bouwen, testen)	Constructie (ontwerpen, bouwen, testen)	Opleveren
	Testen bedenken			

# Maandagmiddag: de bijgewerkte Risico lijst

- Licentie voor ontwikkeltools is verlopen
- Database is te duur
- Werkt het mechanisme voor communicatie tussen computers wel bij Gary?
- Zitten alle machines bij Gary wel op het netwerk?
- Synchronisatie en updates in de database
- Consistentie van alle activiteiten, projecten en gebruikers over alle machines heen
- Toegang tot de database voor admin en gebruikers
- Zelfde gebruiker twee maal verbonden op verschillende machines
- Dialoog met gebruiker stopt en blokkeert anderen

# Maandag laat: het prototype werkt



# Dinsdag: prototype bespreken met Gary

- Prototype heeft zijn beperkingen:
  - **1 gebruiker (Gary)**
  - **1 activiteit (Nadenken)**
  - **start en stop**  
**(onderbreken en doorgaan toegevoegd tijdens demo)**
- Na vijf minuten een crash!
  - **Uitleggen dat robuustheid en compleetheid later komen**
- Look & Feel besproken
- Eric maakt zich zorgen over verlies van data

**Maar een Go!**

## Dinsdag: ups en downs

- Risico eraf, risico erbij
- Interfacen met database lukt niet
- Systeem blijft crashen, te oude versie van database software
- Opgelost met tech support en download nieuwe versie (heeft veel tijd gekost)



## Woensdag: meer voortgang

- Probleem dataverlies opgelost
- Kopie van sources (huidige configuratie) apart gezet (je weet maar nooit)
- Lijst met testen gemaakt
- Dialoog toegevoegd om gegevens in Excel te bekijken

Eric belt met extra requirements:

- Persoon kan aan meer activiteiten
- Tegelijk werken, meerdere timers nodig

**Toevoegen aan risicolijst**

## Donderdag: bijna klaar

- Testen ok
- Netwerkprobleem opgelost
- Ipv “activiteit” nu “activiteit op projekt”
- Kleine handleiding
- Meer testen. Concurrency probleem op database. Oei!
- Concurrency probleem ‘s avonds laat opgelost



## Vrijdag ochtend: bêta en testen

- 's ochtends naar Gary's bedrijf
- Installatie op verschillende machines
- Installatie van de database
  
- Korte instructie aan medewerkers die gaan testen
  
- Bug list en wish list  
twee grote en twaalf kleintjes (soms smaak)




## Vrijdag middag – bugfixen op opleveren

- Kritieke problemen allemaal opgelost
- Drie kleine problemen laten zitten
- Vind zelf nog vier nieuwe problemen, direct opgelost
- Final release, releasenotes, installatieprogrammaatje

Om 22:00 uur is alles af.  
CD'tje branden en Champagne!  
Maandag installeren.

# Agenda

- Korte beschrijving van het probleem
  - Wat gebeurde er van dag tot dag
    - Van het eerste contact
    - Tot en met de oplevering
  - Tijdens de presentatie
    - Schrijf op wat je opvalt
    - Zowel positief als negatief
- Daarna korte bespreking
- En wat heeft dit te maken met  rocessen

# Wat valt je op

## Hoe loopt het voor Nick (opdrachtnemer)

- Positief
  - ...
  - ...
  - ...
- Negatief
  - ...
  - ...
  - ...

## Hoe loopt het voor Gary (opdrachtgever)

- Positief
  - ...
  - ...
  - ...
- Negatief
  - ...
  - ...
  - ...



# Wat viel mij op?

- Plan met een iteratief proces
- Business Case
- Visie afgestemd
- Requirements opgeschreven
- Nick is zich bewust van Risico's
- Ontwerp en architectuur voor belangrijke dingen

Klant tevreden  
Nick tevreden

Maar:

- Wel lange dagen!
- Wat als Nick ziek wordt?

- Vroeg testen, vaak testen.
- Bug list en Change list geprioriteerd (niet alles)
- Configuratie bewaard om op terug te vallen
- Gebruikers documentatie en release notes

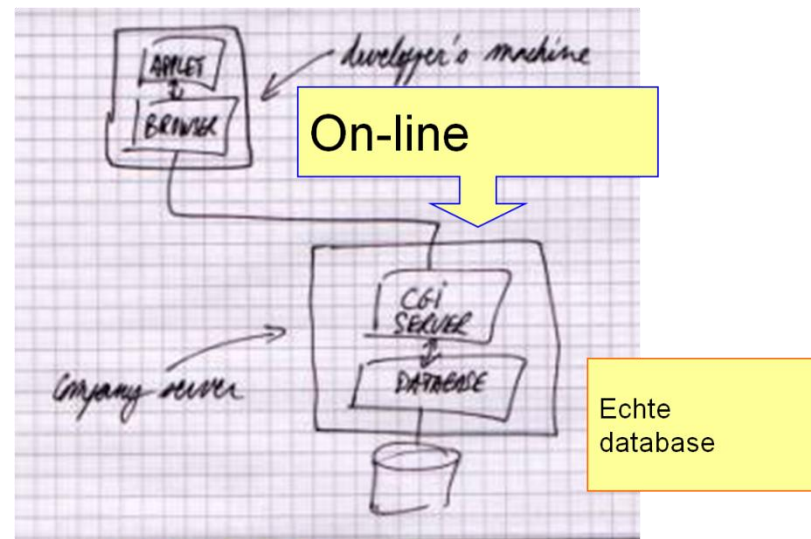
# Requirements in project van Nick

- Visie
  - **Probleembeschrijving**
  - **Oplossingbeschrijving**
  - **Stakeholders**
  - **Ondersteunde taken**
- Later ook:
  - **Alternatieven en foutafhandeling bij ondersteunde taken**



# Architectuur in project van Nick

- Architectuur
  - Eenvoudig, Client/Server
  - Schets



# Test in project van Nick

- Vroeg testen, Vaak testen
  - **Door Nick zelf**



# Agenda

- Korte beschrijving van het probleem
  - Wat gebeurde er van dag tot dag
    - Van het eerste contact
    - Tot en met de oplevering
  - Tijdens de presentatie
    - Schrijf op wat je opvalt
    - Zowel positief als negatief
  - Daarna korte bespreking
- En wat heeft dit te maken met  rocessen

# En wat heeft dit te maken met rocessen?

- Proces van Nick is lichtgewicht
  - Weinig documentatie
  - Weinig ceremonie en procedures
  - Documentatie bruikbaar voor later
    - Uitbreidingen, lessons learned, planning
- **Proces van Nick bevat**

**ALLE**

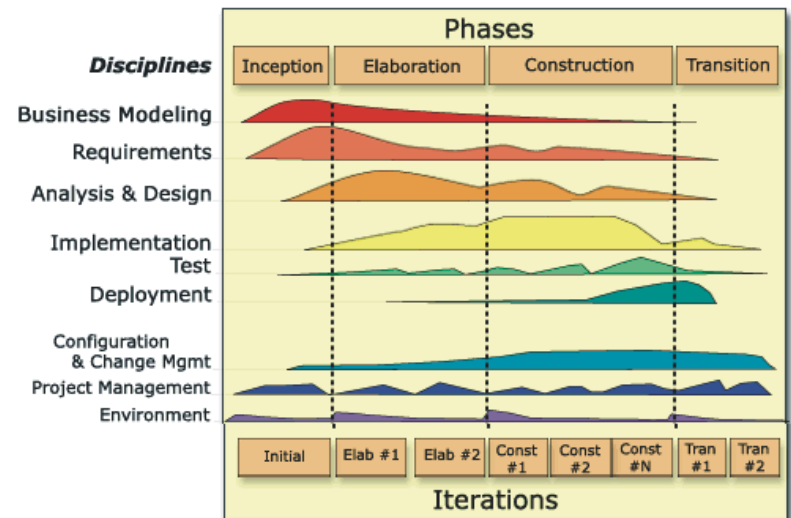
**belangrijke stappen van RUP**

**belangrijke stappen van Endeavour aanpak Infosupport**

# Best Practices



- Develop Iteratively
- Manage Requirements
- Use Component Architectures
- Model Visually (UML)
- Continuously Verify Quality
- Manage Change



# Wat viel mij op?

- Plan met een iteratief proces

Develop iteratively

- Business Case
- Visie afgestemd
- Requirements opgeschreven

Manage Requirements

- Nick is zich bewust van Risico's
- Ontwerp en architectuur voor belangrijke dingen

Use architecture

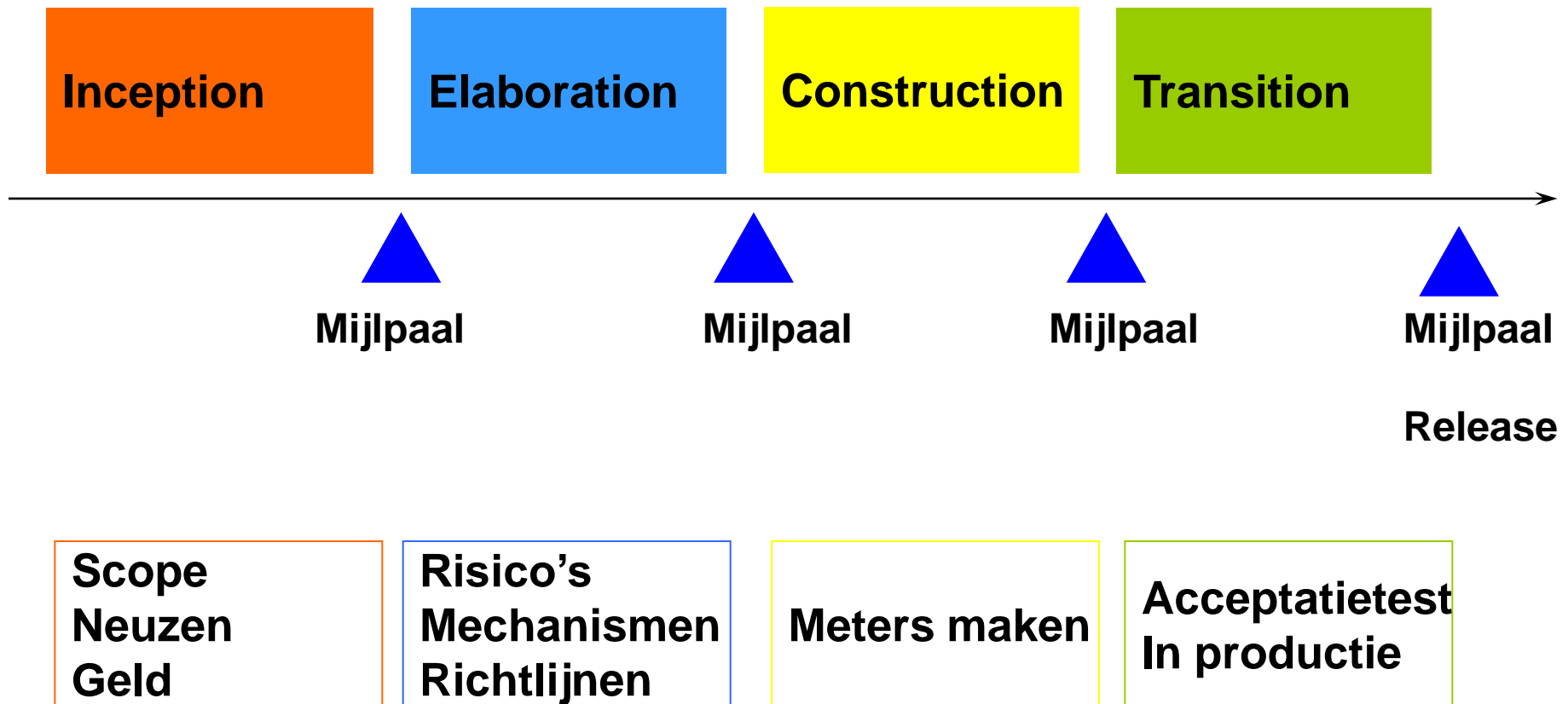
Model visually

- Vroeg testen, vaak testen.
- Bug list en Change list geprioriteerd (niet alles)
- Configuratie bewaard om op terug te vallen
- Gebruikers documentatie en release notes

Continuously verify quality

Manage change

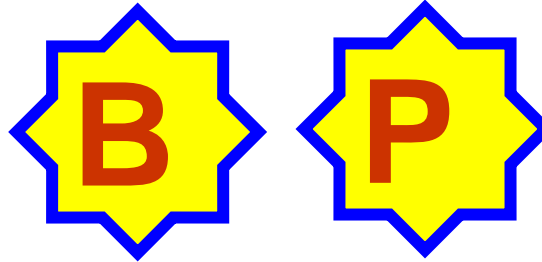
# Ook de RUP Fasering...






# Mijn conclusie?

- Best practices kan je altijd toepassen
  - Grote projecten
  - Kleine projecten
  - Wel even tunen op het project



# Agenda

- Korte beschrijving van het probleem
- Wat gebeurde er van dag tot dag
  - Van het eerste contact
  - Tot en met de oplevering
- Tijdens de presentatie
  - Schrijf op wat je opvalt
  - Zowel positief als negatief
- Daarna korte bespreking
- En wat heeft dit te maken met  rocessen

# Vragen en Antwoorden

