

Software analysis

Introduction

The essential software requirement

Why do we need requirements?

Requirements problem		Consequence	Solution
1.	Geen communicatie	Misverstanden	Agenda punten opstellen
2.	Gebrek aan individuele motivatie	Oneerlijke verdeling van werk,	Groepsgevoel creëren
3.	Geen reflectie	Verder bouwen op fouten	Tijd maken om te leren uit fouten
4.	Geen groepsleider	Onduidelijke taakverdeling	Ieder een bepaalde functie geven

- ➔ De meest significante bijdrage aan het falen van projecten hebben betrekking tot de 'requirements'.
- ➔ De meest voorkomende oorzaak van het falen van een project: **changing requirements**.
- ➔ Het grootste probleem in software development processen: **requirements management**.
- ➔ Door minder dan 5% in het verwerken van requirements te investeren: **budget overruns**.

Who needs requirements?

Stakeholders		Requirements needs
1.	Project leider	Kostprijs (vb. max. €10.000), deadline (vb. voor een bepaalde datum), Geïnvesteerde tijd (vb. binnen de 200 manuren)
2.	Sales	Wat kan het programma, afleverdatum, prijs,
3.	Project team members	Development, deadline (hoeveel tijd heb ik), layout,
4.	Project testers	Zijn er nog features die nog niet geïmplementeerd zijn, voldoet het aan de requirements, is het gebruiksvriendelijk voor het aangewezen publiek

- ➔ Conclusie: requirements hebben een directe invloed op het succes of de vooruitgang van een project.

What are requirements?

- ➔ Requirements zijn specificaties van wat zou uitgevoerd moeten worden. Het zijn beschrijvingen van hoe het systeem zich zou moeten gedragen. Het kunnen beperkingen zijn op het ontwikkelingsproces van het systeem.
- ➔ Je kan het vergelijken met een contract tussen de stakeholder en de project manager.
- ➔ Volgens IEEE is een requirement:
Een vermogen dat nodig is voor een gebruiker om een probleem op te lossen of een doel te bereiken.

Levels and types of requirements

Er zijn drie soorten requirements:

→ **Functional requirement**

Definieert een functie die moet worden voorgesteld door het systeem.

→ **Quality requirement**

Definieert een kwalitatieve eigenschap dat door het systeem moet worden gecreëerd.

→ **Constraint**

Een requirement die de ruimte voor oplossingen beperkt om aan de noodzakelijke functionele en kwalitatieve eisen te voldoen.

Non-functional requirements

Term	Description
Business requirement	Een high-level business resultaat van de organisatie die een product produceert of van een klant die het koopt.
Business rule	Een beleid, richtlijn, standaard of regelgeving die sommige aspecten van een business bepaalt. Geen software requirement zelf, maar het ontstaan van verschillende soorten software vereist.
Constraint	Een beperking die wordt opgelegd aan de keuzes van de ontwikkelaar voor het ontwerp en de bouw van een product.
External interface requirement	Een beschrijving van een verbinding tussen een software systeem en een user, een ander software systeem of een hardware apparaat.
Feature	Een of meer logisch gerelateerde systeem mogelijkheden die waarde bieden aan een gebruiker en worden beschreven door een set van functionele eisen.
Functional requirement	Een beschrijving van een gedrag dat een systeem vertoont onder specifieke voorwaarden.
Non-functional requirement	Een beschrijving van een eigenschap of kenmerk dat een systeem moet vertonen of een beperking dat het moet respecteren.
Quality attribute	Een soort van niet-functionele requirement dat een service of gedragskenmerk van een product beschrijft.
System requirement	Een top-level requirement voor een product dat meerdere subsystemen bevat, die alle software kunnen zijn of software en hardware.
User requirement	Een doel of taak dat specifiek klassen van gebruikers in staat moeten zijn om te presteren met een system of een gewenst product.

Bad VS. Good requirements

Bad/good requirement?	Why?
1. We need to be able to respond to a code red incident anywhere on the planet within 24 hours.	Goed, de tijd is bepaald. (Als er ASAP stond, was het een slechte requirement)
2. The system shall validate and accept credit cards and cashier's checks. High priority.	Welke credit kaarten?
3. The system shall process all mouse clicks very fast to ensure user's do not have to wait	Wat is "heel snel"?
4. I want the system to automatically calculate sales taxes based on relevant sales tax laws.	Wat is relevant?
5. The website visitor won't need to click more than once to get to the order page from any other page on the site.	Er zal op iedere pagina een knop komen te staan.
6. The user must have Adobe Acrobat installed.	Welke versie?
7. Sales needs to be able to see which contracts will be expiring within the upcoming 90 days.	Goed, ze weten voor wie het bested is, ze weten wat nodig is en voor welke tijd.
8. The clerk enters basic loss information specific to the claim line. The system confirms that there are no existing, possibly competing claims and assigns a claim number. The clerk confirms they are finished; the system saves and triggers acknowledgement to be sent to the agent.	Een duidelijk stappenplan.

Requirements development & management (RE)

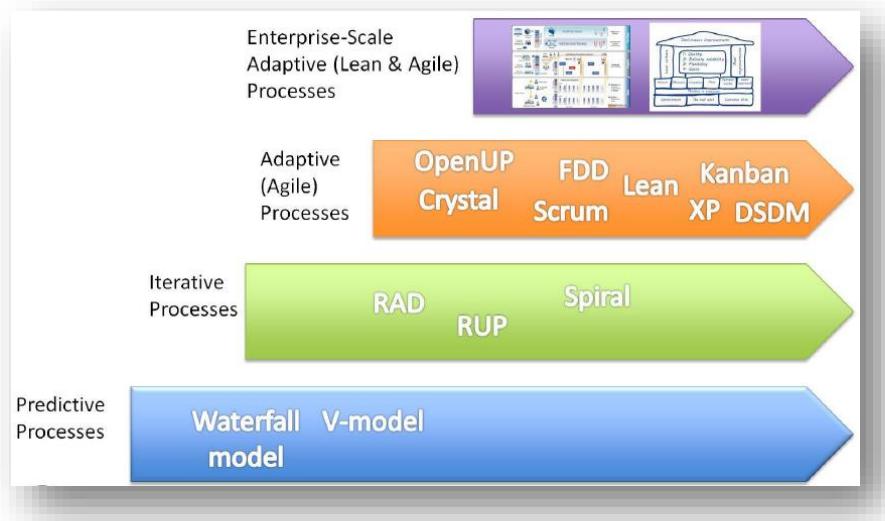
➔ Een systematische en gedisciplineerde aanpak van de specificatie en het beheer van requirements met de volgende in doelen :

- Het kennen van de relevante requirements, het bereiken van een consensus tussen de stakeholders over deze requirements, het documenteren van hen volgens de gegeven standaarden, en systematisch te beheren,
- Inzicht in en documenteren van de wensen en behoeften van de stakeholders,
- Specificeren en beheren van requirements om het risico van het leveren van een systeem dat niet aan de wensen en behoeften van de stakeholders voldoen, te minimaliseren.

➔ 4 hoofdactiviteiten:

- Elicitatie
- Documentatie
- Validatie en onderhandeling
- Beheer van requirements

Brief history of requirements



The role of the analyst, requirements engineer

- Informatie, aangeboden door een groot aantal mensen, analyseren
- Verantwoordelijk voor het opwekken van de werkelijke behoeften van stakeholders, niet alleen hun uitgesproken wensen
- Vergemakkelijkt de communicatie tussen organisatie units
- De business behoeften verbinden met informatietechnologie
- Kan als een 'vertaler' dienen tussen business units

IIBA underlying competencies

- Analytisch en probleemplossend denken
 - Creatief denken
 - Beslissingen nemen
 - Leren
 - Problemen oplossen
 - Systeem denkend
- Gedragskenmerken
 - Leer van goed en kwaad
 - Persoonlijke organisatie
 - Vertrouwensgevoel
- Kennis van zaken
 - Professionalisme
 - Ken je industrie
 - Begrijp de structuur van een organisatie
 - Oplossend denken
- Communicatievaardigheden
 - Mondelinge communicatie
 - Didactische vaardigheden
 - Schriftelijke communicatie
 - Luisteren

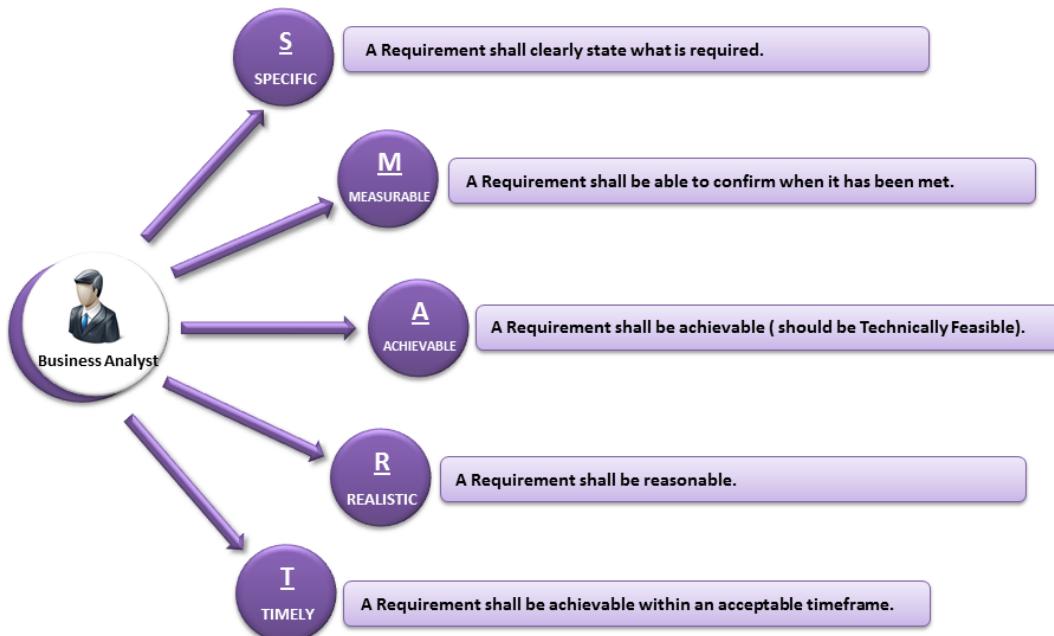
→ Interactievaardigheden

- Onderhandelen
- Leiderschap en beïnvloeden
- Teamwork

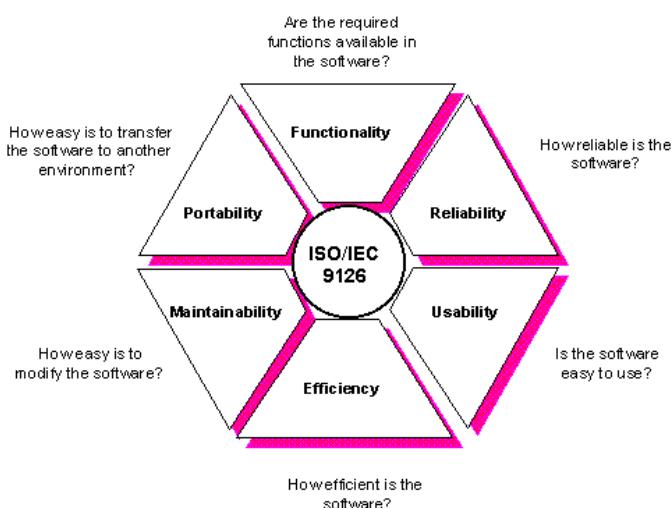
→ Softwaretoepassing

- Algemene applicaties
- Gespecialiseerde applicaties

SMART

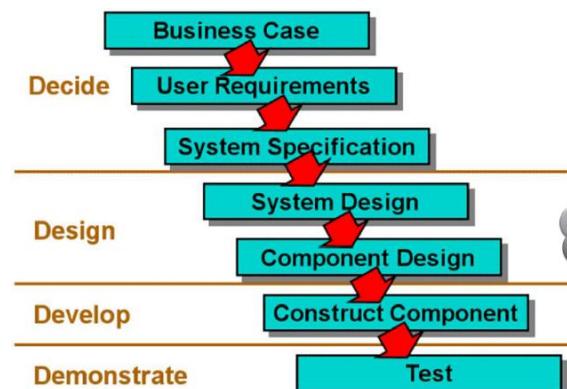


ISO 9126: The standard of reference

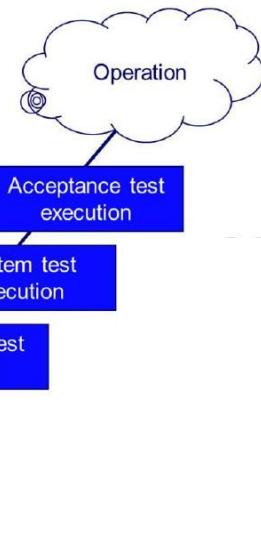
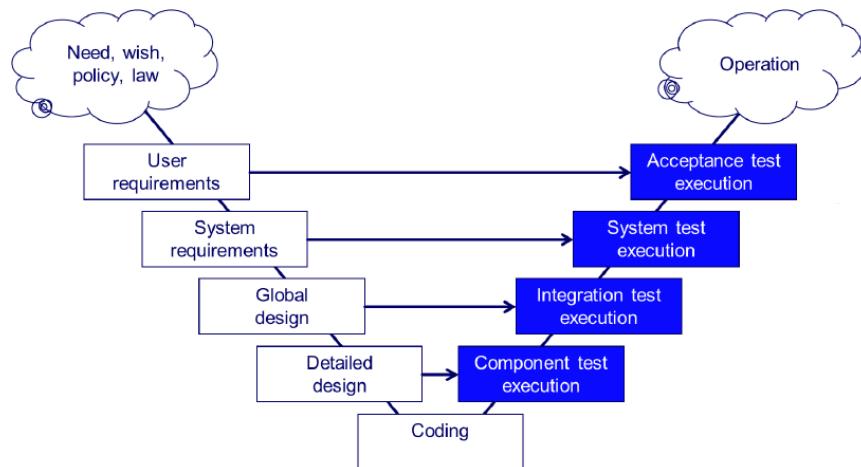
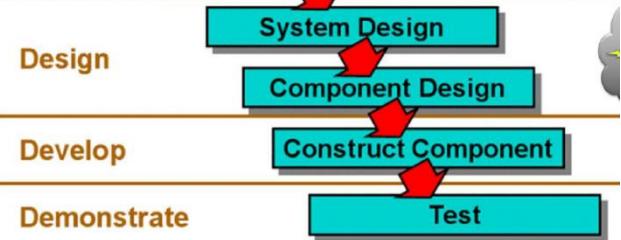


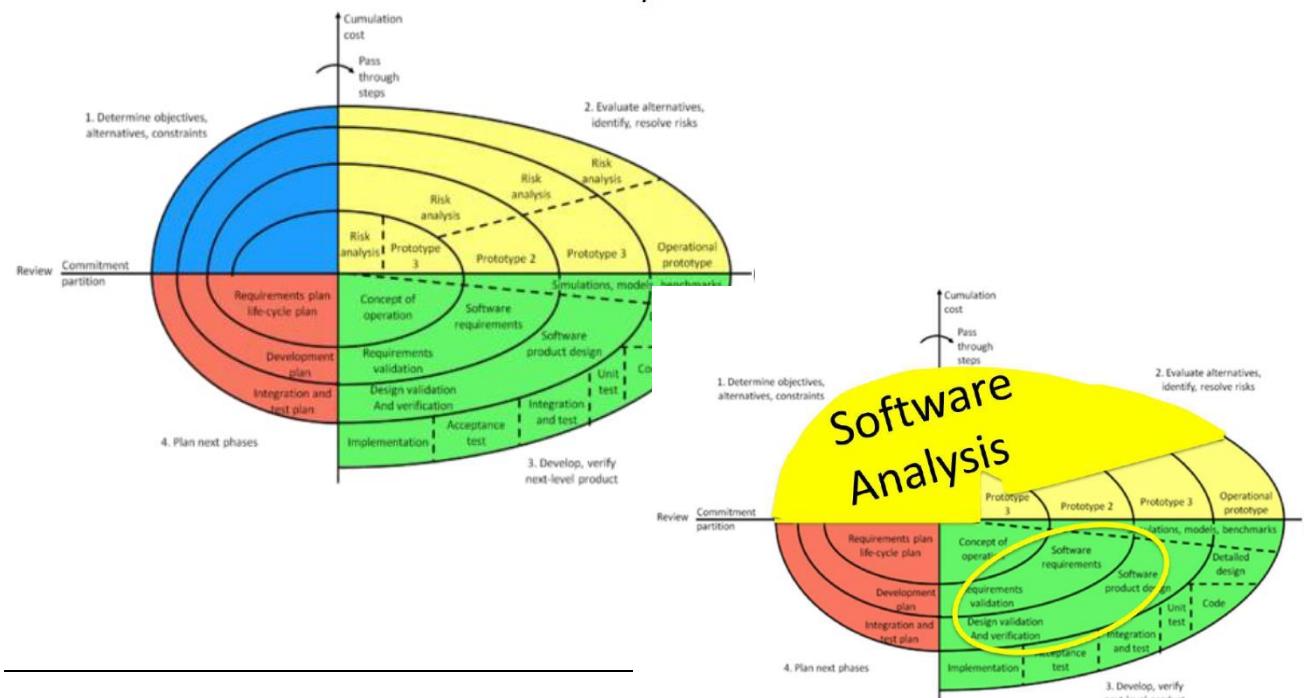
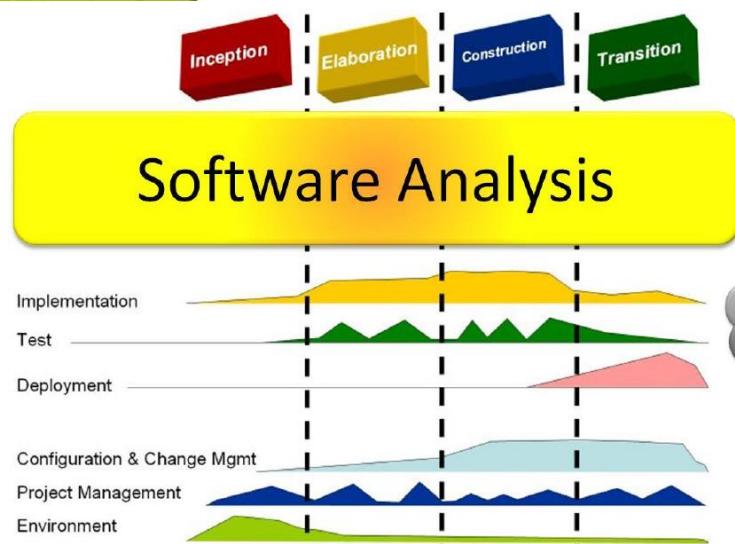
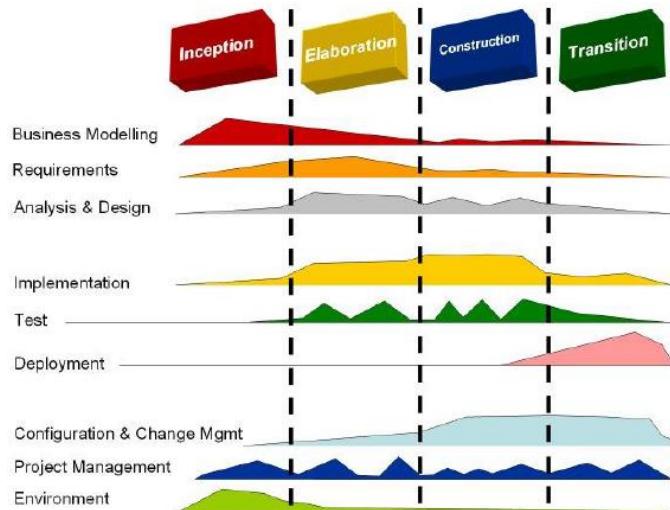
Het doel van deze standaard is om een kader voor de beoordeling van de kwaliteit van de software te bieden . ISO / IEC 9126 stelt geen eisen voor software , maar het definieert een kwaliteitsmodel dat van toepassing is op elke vorm van software.

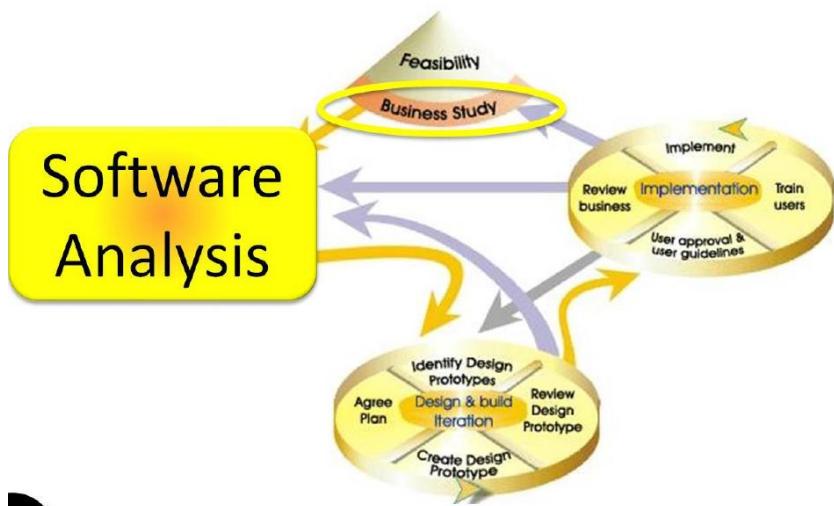
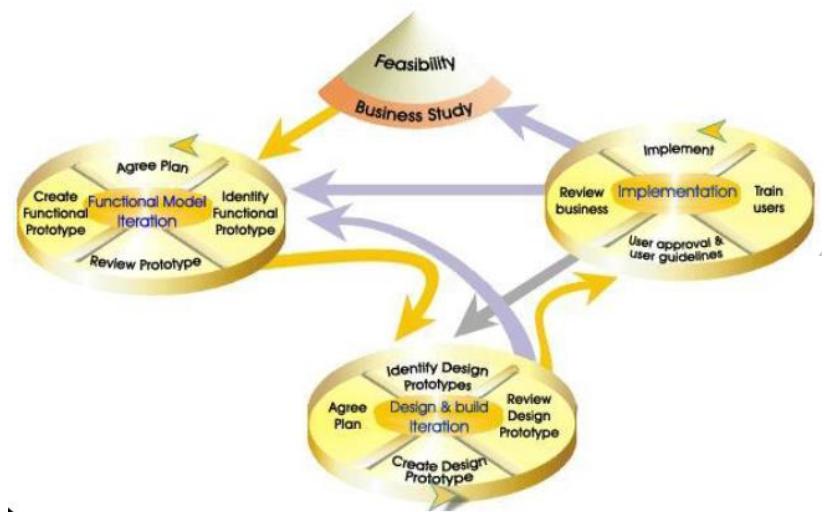
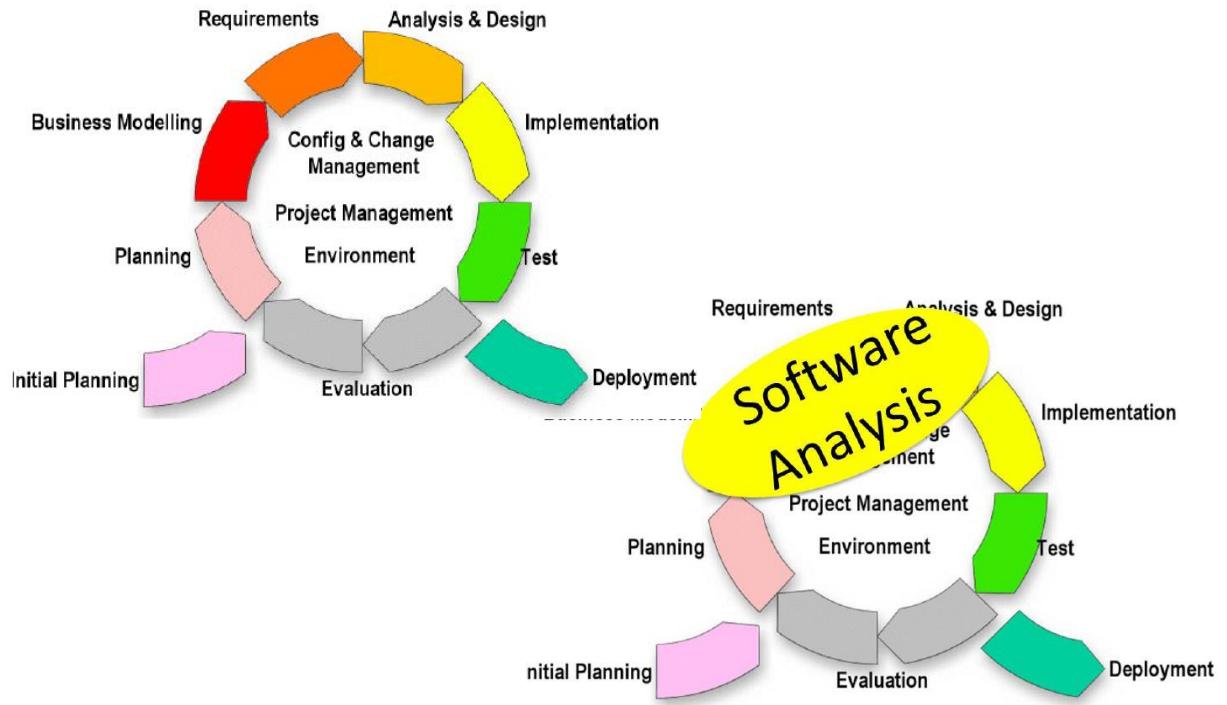
Requirements process

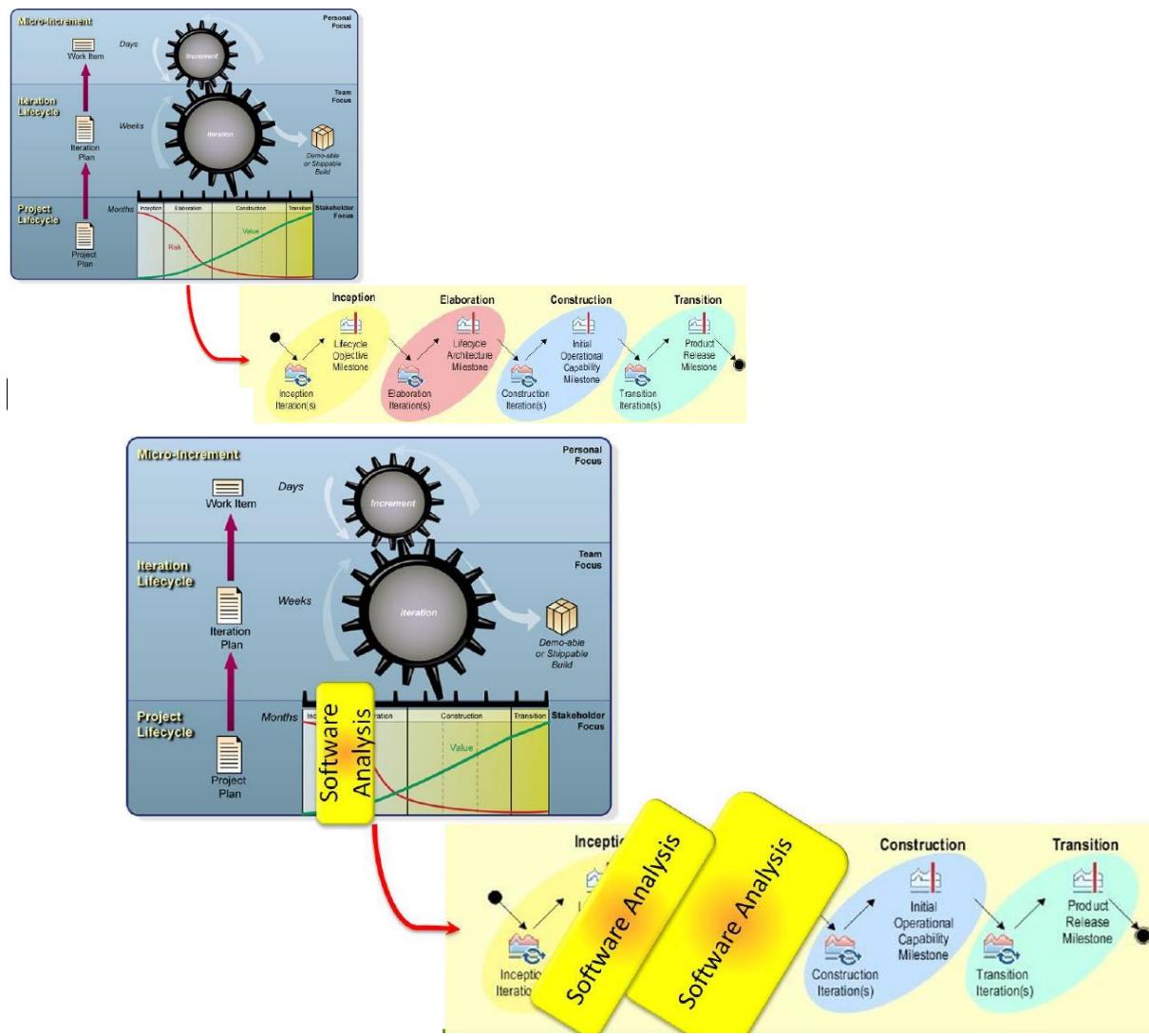
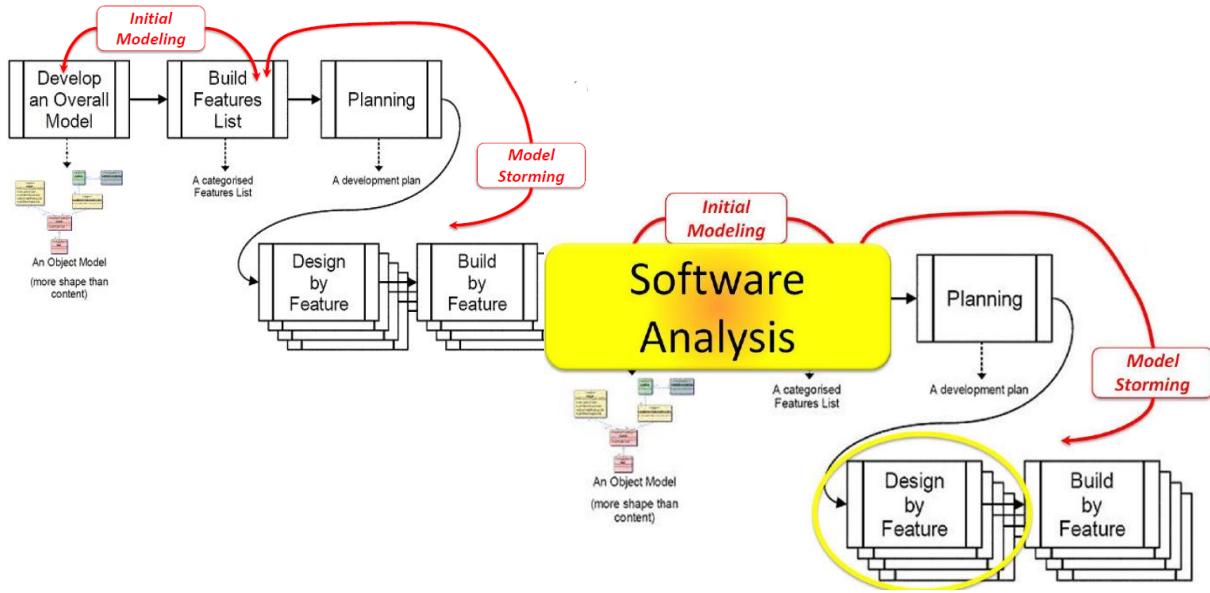


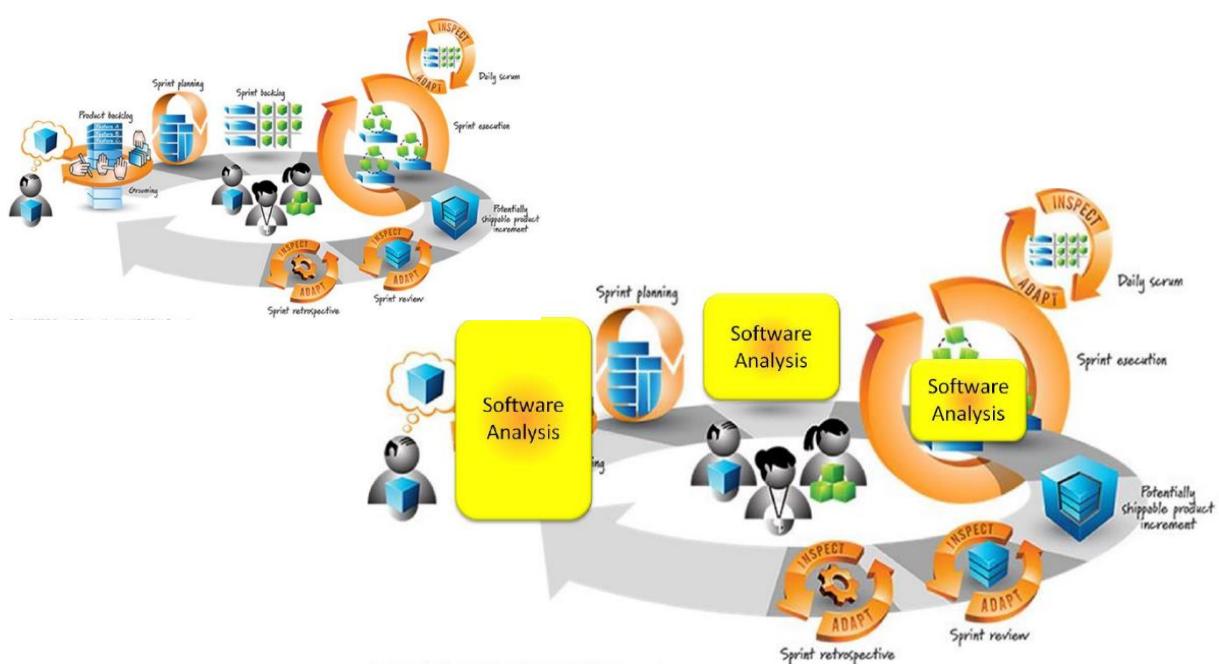
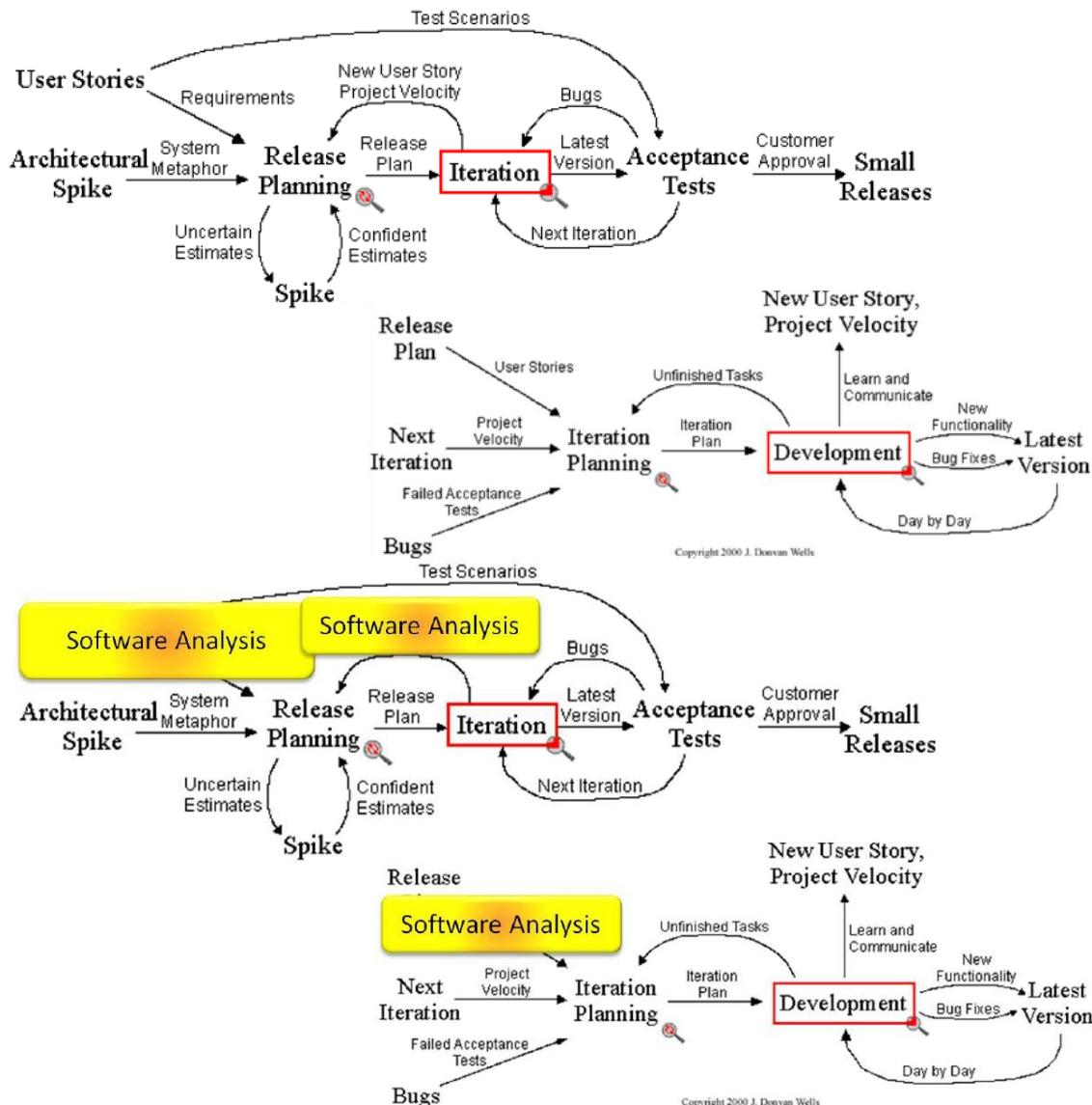
Software Analysis

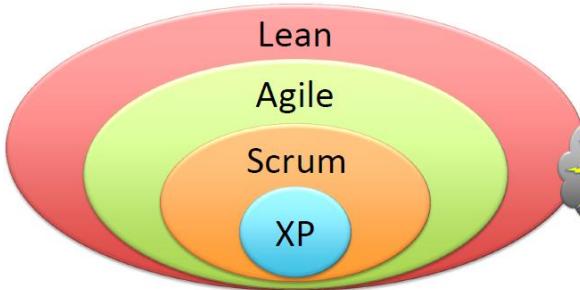
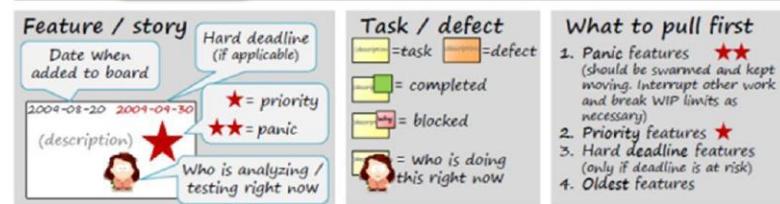
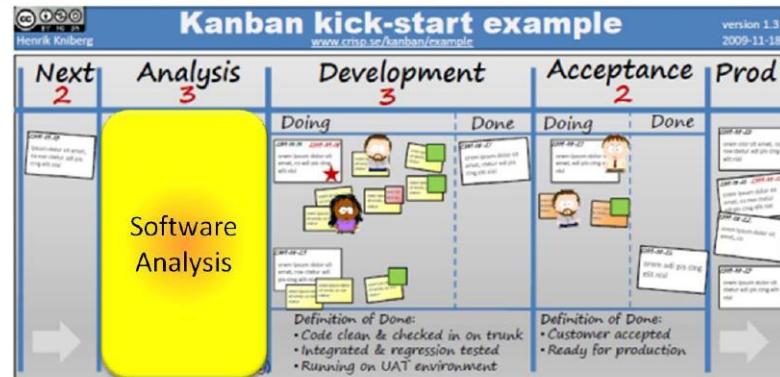
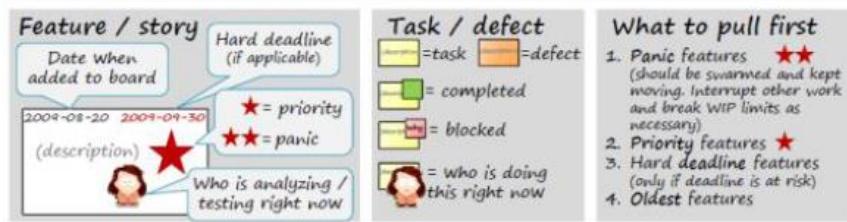
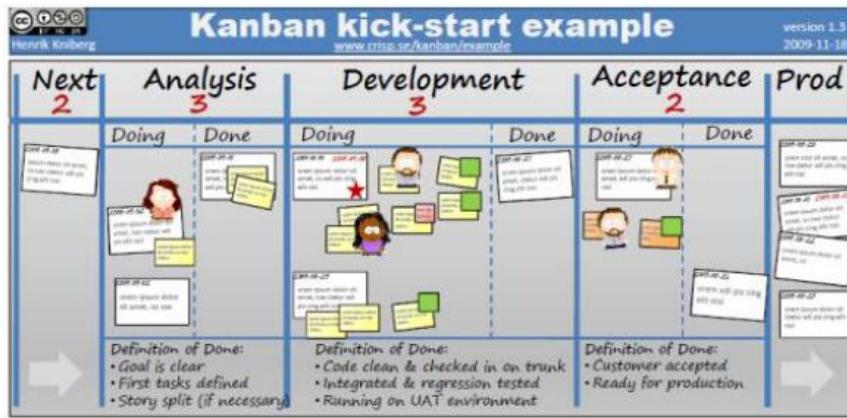












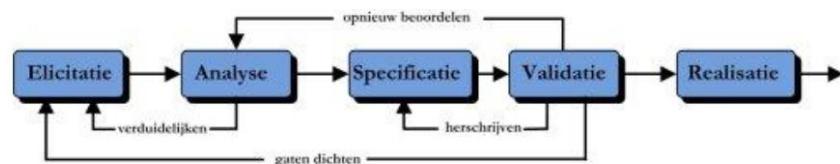
Commonly used requirements processes

Process 01

- ➔ Uniforme aanpak ontbreekt
- ➔ Requirements analyst:
 - Het verrichten van werkzaamheden op eigen oordeel
 - Persoonlijke ervaringen en voorkeuren spelen een grote rol
 - Veel verschillen binnen eenzelfde organisatie
 - Het wiel opnieuw uitvinden

Process 02

- ➔ Vroeger -> deel van het ontwikkelen van requirements
- ➔ Nu -> gebruikt als stappen in het requirementsproces
- ➔ Stappen zijn iteratief

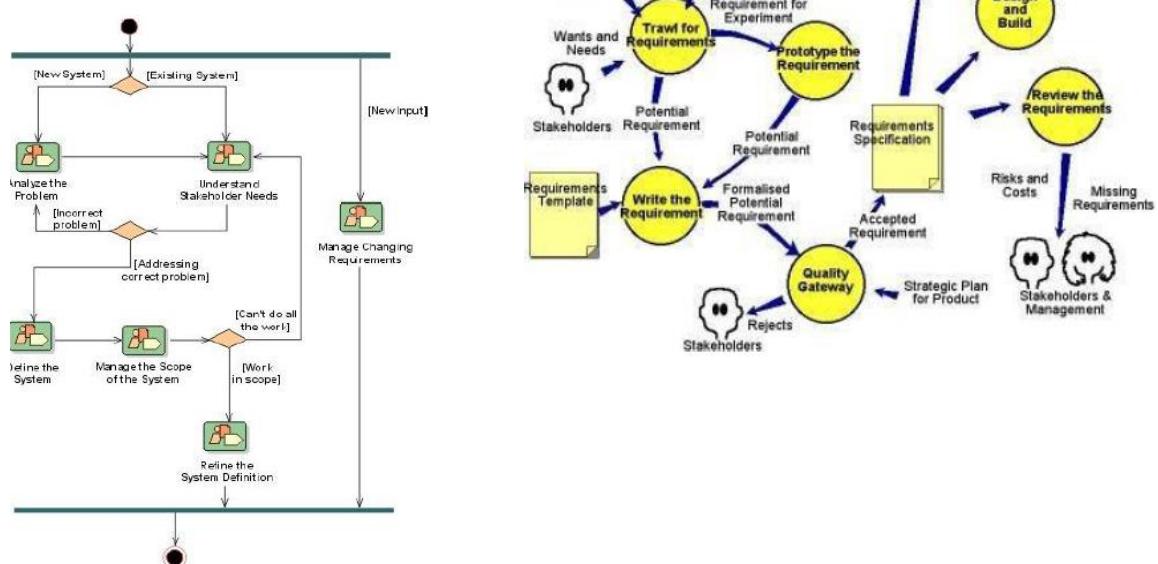


Process 03

- ➔ Volere¹ requirements process

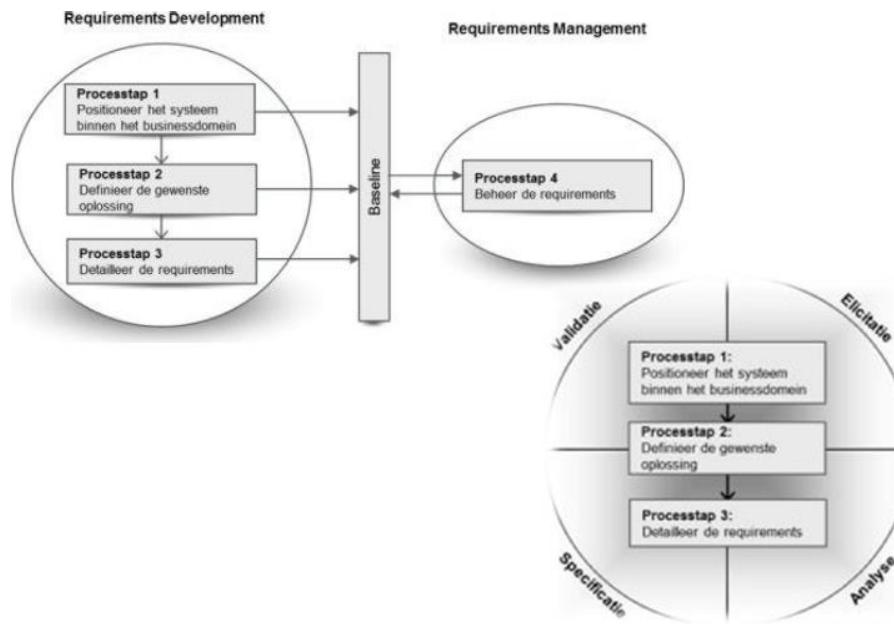
Process 04

- ➔ RE is een van de disciplines



¹ Volere is the umbrella that covers the collection of requirements templates, processes, books, consulting and training. Since its inception, Volere has been used by thousands of organizations around the world.

Process 05



The process used within the course

4 hoofdactiviteiten voor RE:

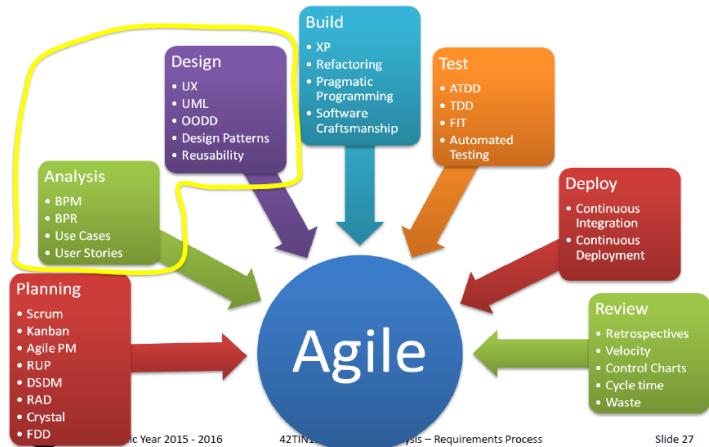
- ➔ Elicitatie²
- ➔ Documentatie
- ➔ Validatie & Onderhandelen
- ➔ Beheer van eisen

Recapitulation case “BA wanted”

“A small scale bank is looking for a new business analyst.

This bank has offices in several European countries and its head office is located in the Netherlands. The bank is working with an internet based Bank Information System. The new business analyst will be working in the head office. He/she will be the link between the business and the (IT-)projects and be responsible for the translation from change requests into business requirements and for new projects also for the elicitation of the business requirements.

The bank has adapted the SCRUM method and the business analyst will be participating in the scrum teams.”



² het begrijpen van het probleem waarvoor een (geautomatiseerde) oplossing wordt gezocht

	Requirements	Relevant
1	Successfully completed (business) information technology or economics studies	
2	Several years of experience as software developer and project leader /team leader in the area of ERP are desirable	
3	Good system development skills. You master at least one current programming language (ideally .net)	
4	Very good SQL knowledge and experience with MS SQL Servers	
5	Know-how in Web standard technologies such as HTML and XML/XSL/XSLT as well as knowledge in MS server technology IIS and Share Point	
6	Very good skills in the English language	
7	Analytical thinking skills and strength in conceptual areas	
8	Strong orientation to services and consulting	
9	High level of commitment	
10	Sure and convincing attitude	
11	Ability to work under high work pressure	
12	Convincing communication skills	
13	Being able to deal with cultural differences in an adequate way	
14	Testing experience on both system and acceptance level	
15	Domain knowledge of banking processes	

System and system context - General

System and system context

"If you can't describe what you are doing as a process, you don't know what you are doing."

Launching the requirements phase

→ De requirement elicitaie "Kick-off"

Consensus³ bereiken van de belangrijkste stakeholders

Om ervoor te zorgen dat je genoeg kennis hebt om requirements te eliciteren

Om ervoor te zorgen dat het project levensvatbaar is

Om het bereik vast te stellen van het werk wat nog gedaan moet worden

→ Stakeholders

Definieer de menselijke samenleving die een effect op het succes of op het project kan hebben. Een stakeholder is iemand die iets wint of verliest (een functionaliteit, inkomsten, status, ...).als gevolg van een project.

Zie stakeholder checklist.

→ Goals

Definieer het succes criteria voor het project.

Hoe weten we of het project een succes is of niet?

Worden gebruikt om het project te begeleiden en om het projectteam keuzes te helpen maken.

→ Scope

Definieert de grenzen van het onderzoek en van het op te bouwen product door het project

³ Dezelfde mening hebben

Referring to the IEEE 8030 - SRS

→ Het business probleem

Een korte beschrijving van de situatie die leidde tot de ontwikkelingsinspanning
Beschrijf het werk wat verbeterd kan worden

→ Doelen van het project – PAM

Wat zal het product (niet) doen?

What is the purpose?

What is the business advantage?

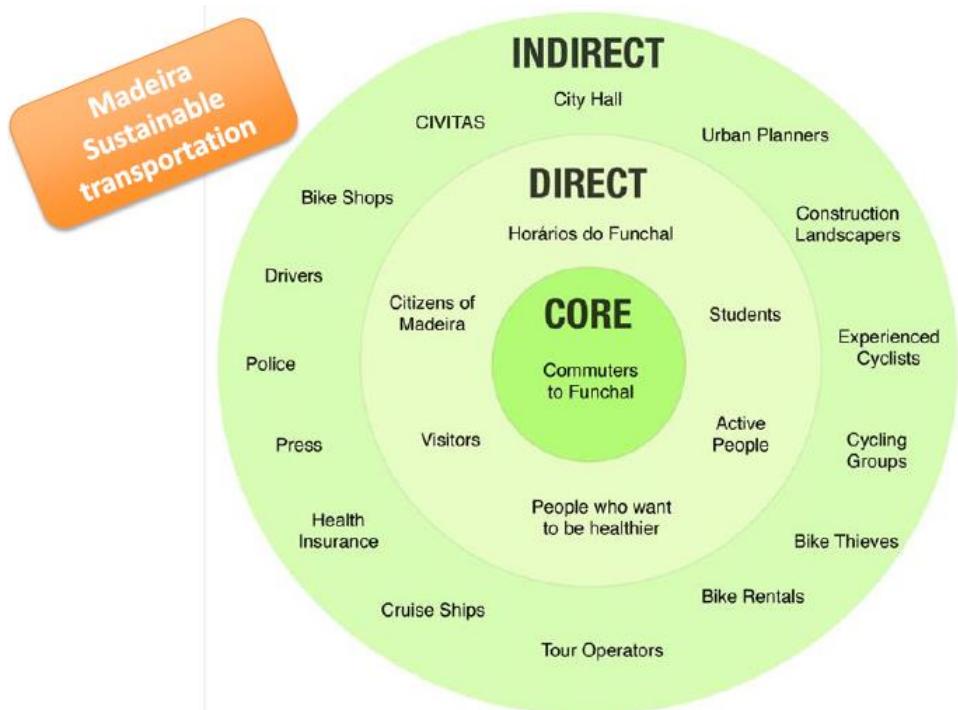
How will you measure the advantage?

Doelen die niet bekend worden gemaakt kunnen niet bereikt worden?

Exploring stakeholders

- Een persoon of organisatie die een (directe of indirecte) invloed op de eisen van het systeem hebben.
- Iedereen die interesse heeft in het project. De stakeholders kunnen het product opbouwen, gebruiken, zijn erdoor aangetast of hebben kennis om het op te bouwen.

Stakeholder mapping



Stakeholderchecklist

- | | |
|---|--|
| <ul style="list-style-type: none">• Project manager• Business experts• Designers• Testers• Client / sponsor• Users / buyers• Usability experts• Operations• Maintenance | <ul style="list-style-type: none">• Security• Safety services• Support / helpdesk• Manufacturing• Marketing• Lawyers• Professional bodies• Special interest groups• Technology experts |
|---|--|

Stakeholder analysis

Type stakeholder	Voorbeelden	Type van invloed
Interne stakeholders	Werknemers	Primaire Stakeholders
	Managers	Primaire Stakeholders
	Aandeelhouders	Primaire Stakeholders
	Investeerders	Primaire Stakeholders
	Ondernemingsraad	Primaire Stakeholders
	Iemand die een project finanziert	Primaire Stakeholders
Externe stakeholders	Leveranciers	Primaire Stakeholders
	Klanten	Primaire Stakeholders
	Media	Secundaire Stakeholders
	Crediteuren	Primaire Stakeholders
	Concurrenten	Secundaire Stakeholders
	Een gebruiker van een system	Primaire Stakeholders
Interface stakeholders	Business experts (consultants)	Secundaire Stakeholders
	Hacker	Primaire Stakeholders
	Overheid (EU/Nationaal)	Secundaire Stakeholders
	Lokale overheid	Primaire Stakeholders
	Maatschappij	Secundaire Stakeholders
	Consumentenbond	Secundaire Stakeholders
Kamer van Koophandel	Kamer van Koophandel	Secundaire Stakeholders
	Opleidingen & Scholen	Secundaire Stakeholders

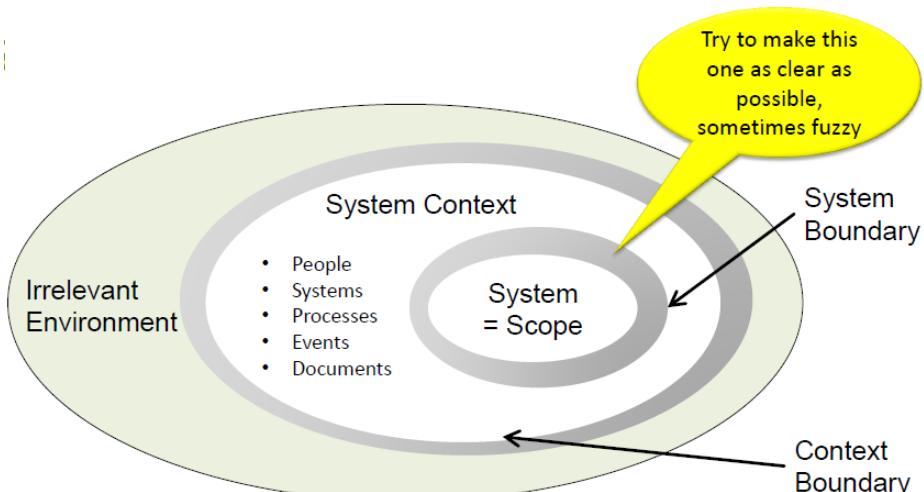
Users of the product

- Het nut om de gebruikers te identificeren: zodat je hun werk kunt begrijpen
- En het product dat je voor hen moet opbouwen
- Schrijf een sectie in je specificatie, voor de gebruikers, om alle gekende en potentiële gebruikers te beschrijven

Exploring the scope

- Definieer de grens van het onderzoek
- Welke informatie heb ik nodig van welke stakeholder?
- Een handige manier om de grens af te bakenen is door een contextdiagram te tekenen:
Mijn systeem en de grens van mijn systeem?
Context en de grens van mijn context?
Irrelevante omgeving?

System context and boundaries



! Let op de grijze zones, zowel system boundary als context boundary kunnen verschuiven over tijd (bv. veranderen van wetten, aspecten die relevant worden voor het geplande systeem, ...)

System context

- ➔ Bron van vereisten voor een systeem
- ➔ Bron = “aspecten die de definitie van requirements beïnvloedde of geïnitieerde”.
- ➔ Potentiële aspecten:
 - Personen (stakeholders)
 - Systemen (technische systemen, software, hardware)
 - Processen (technische, fysieke of bedrijfsprocessen)
 - Evenementen (technisch of fysiek)
 - Documenten (zoals wetten, normen, documentatie)

Context analysis: User groups

- ➔ De functionaliteit en bruikbaarheid van een product geschiedt door de context van het gebruik.
- ➔ De context wordt gekenmerkt door
 - De gebruiker van het product
 - De taken die ze uitvoeren
 - De werkomgeving
- ➔ Tool: checklist **MuSIC**
 - Must
 - Should
 - Important
 - Critical

System context and boundaries

- ➔ Hoe te documenteren?
 - Context diagrammen
=Data flow diagram level 0
 - Use case diagrammen
Actoren (personen of andere systemen) in de omgeving met hun relatie tot het systeem
 - Domain models

Quiz questions - Introduction

1.1 You have to recruit a requirement engineer. Which combination of skills is the best combination?

- A linguistic competent, analytical thinking, testing skills;
- B communication skills, moderation skills, ability to convince
- C domain knowledge, coding skills, testing skills;
- D project management skills, moderation skills, an ability to display empathy;

1.2 A person is about to be assigned to your project as a requirements engineer. What is the biggest risk?

- The requirement engineer:
- A doesn't have project management skills;
- B has no domain knowledge;
- C is introvert and shy;
- D is new in this organisation, so he doesn't have any knowledge about the organisation.

1.3 Which of the following statements best describes the term "stakeholder"?

- A everyone whose wishes have to be considered in the requirements specification;
- B all members of the project team;
- C a person or organization that has a (direct or indirect) influence on a system's requirements;
- D the total of all people named as a source for any requirements specification.

1.5 During an acceptance test a defect was detected, which could be attributed to the requirements having been incorrectly interpreted by the software developers. Which of the statements fits this circumstances? Pick the **two** you think are best

- A the correction will only generate minor costs, since only the requirements specification must be changed;
- B the defect should already have been recognized during the review of the requirements specification;
- C in the worst case, it could happen that the architecture has to be reworked which would generate substantial costs;
- D the defect should already have been recognized during the system test.

1.4 Which of the following statements typically characterizes the relationship between a requirements engineer and a stakeholder in the role of a tester?

- A The requirements engineer provides input for the work of the stakeholder;
- B The results of the requirements engineer are being managed by the stakeholder;
- C The stakeholder provides input for the work of the requirements engineer;
- D The stakeholder monitors the work of the requirements engineer;
- E The work of the requirements engineer is not related to the mentioned role of the stakeholder.

1.6 Which 3 of the following skills are important for the requirements engineer?

- A Communication skills
- B Analytical thinking
- C Conflict resolution
- D Testing skills

1.7 Which statements are TRUE/FALSE for Requirements

- True False
- There are three kinds of requirements: functional, quality and constraints.
- Quality requirements describe functionality.

1.8 Which **one** of the following is **not** one of the four major activities of requirements engineering?

- A Requirements management
- B Requirements elicitation
- C Requirements validation and negotiation
- D Requirements scoping

2.1 To determine scope and boundaries of a system context diagrams are often being used. Which **three** of the following attributes are compulsory in context diagrams?

- A scope;
- B content;
- C context;
- D interfaces (with its environment);
- E people.

2.2 Consider the following statement about scope and context. Which statements are TRUE/FALSE?

- True False
- by setting the scope we specify what “outside” and “inside” means – in relation to the system;
- requirements engineering cannot involve different scopes (e.g. enterprise, department, IT system, etc.);
- context describes the size of the system;
- scope describes the organisations, neighbouring systems, functionality (or similar) with a connection to the target system;
- requirements are always restricted by the scope.

2.3 At the beginning of a project, the boundary between a system and its context is often diffuse, the so-called ‘grey zone’. Indicate which of the following statements are true and which are false.

- True False
- a diffuse boundary is often not recognized for a long time because it is not depicted in the context diagram;
- a diffuse boundary between a system and the context indicates that the interfaces between the system and the environment have not yet been clarified;
- a diffuse boundary between a system and the context exists mainly at the beginning of a RE process and must be managed during the course of the RE process.

2.4 Indicate which of the following statements about the main purpose of a context diagram are true and which are false:

A context diagram is used

- True False
- to identify system boundaries;
- to test the requirements from the point of view of consistency and clarity;
- to identify all stakeholders of the system;
- to illustrate the sequencing of the exchange between the system and its context.

2.5 Indicate the items of information which are mandatory for them to be visible in a context diagram (multiple answers possible)

- A system name;
- B neighbouring technical systems;
- C system functions;
- D logical outputs;
- E system parameters;

2.6 Which statements are TRUE/FALSE for Requirements Engineering?

- **True False**
- A full understanding of system context is essential for successful requirements engineering.
- The system boundary is not likely to shift during the requirements engineering process.

System and system context – Context diagram

What

- ➔ Het context diagram toont welke gegevensstromen tussen de buitenwereld en de informatie systemen bestaan
- ➔ Dit context diagram beschrijft ook de systeem grenzen:
Wat het systeem moet worden en wat het niet mag hebben
- ➔ Context diagram = level 0 data flow diagram
- ➔ ≠ Flowchart

Types

- ➔ Physical DFD
Data flow diagrammen die het model van het huidige systeem representeren
Deze diagrammen worden gemaakt als de analyst het huidige systeem in detail bestudeert
- ➔ Logical DFD
Data flow diagrammen die het model van het voorgestelde systeem representeren

Notation



External Entity

*Representeert een entiteit die gegevens levert of ontvangt van een systeem
Voorbeelden: klant, sales afdeling, bediende, ...*



Data Flow

*Geeft het gegevensverkeer van input tot proces tot output.
Voorbeelden: klanten details, sale rapporten, ...*



Process

*Handelingen op invoergegevens naar de output gegevens.
Voorbeelden: bereken verkopen, ...*

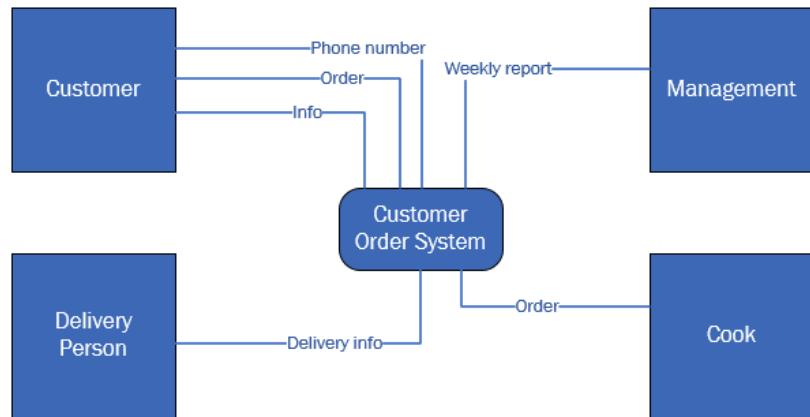


Data Store

*Indiceert het bestand of register waar de gegevens zijn geaccumuleerd.
Voorbeelden: bedienden register, verkoop transacties, klanten bestand, ...*

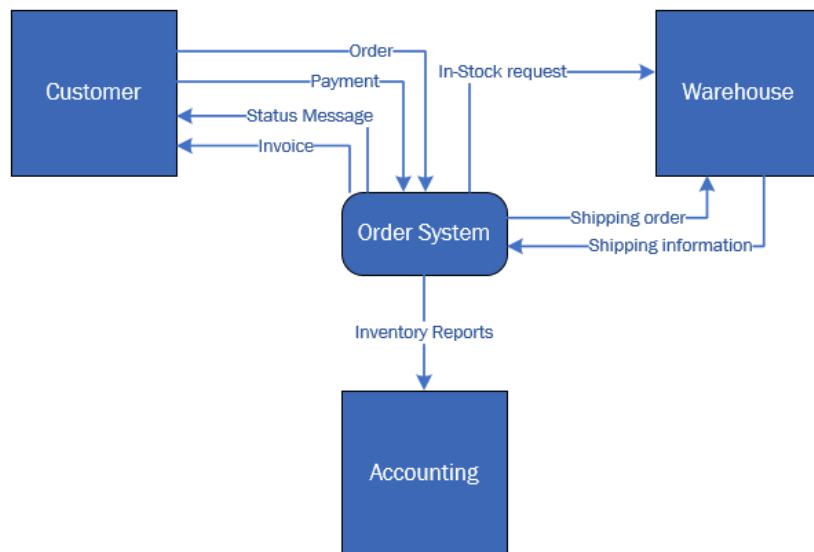
Perfect pizza

Perfect Pizza wants to install a system to record orders for pizza and chicken wings. When regular customers call Perfect Pizza on the phone, their phone number goes automatically into the Pizza system. The phone number invokes the name, address, and last order date comes automatically up on the screen. Once the order is taken, the total, including tax and delivery, is calculated. Then the order is given to the cook. A receipt is printed. Occasionally, special offer (coupons) is printed so the customer can get a discount. Drivers who make deliveries give customers a copy of the receipt and coupon (if any). Weekly totals are kept for comparison with last year's performance. Draw a CONTEXT DIAGRAM.



Precision Tools

Precision Tools sells a line of high-quality woodworking tools. When customers place orders on the company's Web site, the system checks to see if the items are in stock, issues a status message to the customer, and generates a shipping order to the warehouse, which fills the order. When the order is shipped, the customer is billed. The system also produces various reports. Create a CONTEXT DIAGRAM.



System & system context – Use case diagram

UML

- Unified modeling language
- Ontwikkeling, modelering taal in het gebied van software engineering, dat een standaard manier voorziet om het ontwerp te visualiseren
- Is geen methode
- UML modellen zijn grafische voorstellingen van bepaalde aspecten van het informatiesysteem

Business VERSUS system use case

Een business use case is een manier waarop een klant of enkele andere belanghebbende partijen gebruik kunnen maken van het bedrijfsleven om het resultaat te krijgen die ze willen of het nu gaat om een item te kopen, een nieuw rijbewijs te krijgen, om een factuur te betalen, of wat dan ook. Een belangrijk punt is dat een enkele uitvoering van een bedrijf use-case alle activiteiten moet omvatten die nodig zijn om te doen wat de klant (of andere actor) wil, en ook eventuele activiteiten die het bedrijf moet doen voor het proces compleet is vanuit haar standpunt.

Een systeem use case is een manier waarop een gebruiker van een computersysteem gebruik kan maken van het systeem om het resultaat te krijgen dat ze willen. Dit zal typisch iets zijn dat we gemakkelijk kunnen voorstellen als in één zitting op een enkele pc of een ander apparaat, zoals een geldautomaat of een mobiele telefoon, meestal met een enkele gebruikersinterface, of een klein aantal nauw verwante schermen zoals een wizard, en neemt misschien een paar minuten tot een half uur maximaal in beslag.

Aspect	Business Use-Case	System Use-Case
Who's the Primary actor?	Mainly a business actor e.g. customer; maybe other external party (regulator, shareholder) or an internal party (manager, etc)	Mainly a human user who initiates system behaviour; maybe another system, "scheduler" etc. But by definition a system actor
What's the use case for?	Something the actor wants to get done by using the business / organisation	Something the actor wants to get done by using the system / application
Who / what else may be involved?	May involve interaction with other external business parties as supporting actors	May involve interaction with other systems internal or external to the organisation.
What does it describe?	Describes an interaction involving the primary actor, the relevant parts of the business , and any supporting actor(s), in terms of their business behaviour	Describes an interaction involving the primary actor, the relevant parts of the system , and any supporting actor(s), in terms of their system behaviour . In the case of the primary actor, this means only their actions detectable by the system , such as making selections, supplying data etc.
How's it executed?	May involve many organisation units, systems (or not), technologies, manual / mental procedures etc.	Executed by automated steps in the system
Duration	Of Varying duration - May be very brief or very long-duration.	Typically quite short duration (Cockburn's "coffee-break" rule)

Business use case

Business use case

- ➔ De business use case beschrijft in een technologie vrije manier een bedrijfsproces als een opeenvolging van (inter)acties tussen een bedrijf actor en een werknemer als geheel om een doel van de business actor te vervullen.
- ➔ De business use case beschrijft een proces dat waarde levert aan de business actor, en wat het proces doet .

Business actor

- ➔ Een bedrijf actor staat voor een zakelijke rol die samenwerkt met het ondernemingsklimaat/proces.

Business events

- ➔ Een business event is iets dat buiten de werkingssfeer van SuD/werk gebeurt.
- ➔ Een business event vindt plaats buiten het bereik van het werk/organisatie.
- ➔ Het werk / organisatie leert dat een gebeurtenis heeft plaatsgevonden door de komst van een aantal gegevens (trigger).

Time-triggered business events

- ➔ Gebeurt wanneer een vooraf afgesproken tijdstip is bereikt
 - Een periodiek voorval
 - Een vast tijdsinterval
 - Een bepaalde hoeveelheid tijd die verstrekken is sinds een andere business event

Business use case - Format

Dia 12 – 14 pp WK04 – 11 – System and System Context SW-Analysis-1516 – 02 UseCase-Solution.pdf

Business use case - Example

Business Use Case Name

Check passagier in voor de vlucht.

Business Event description

De passagier besluit om in te checken.

Triggering business event

De passagier toont het ticket & paspoort aan een check-in agent in de luchthaven.

Preconditions

De passagier moet een geldige reservatie hebben.

Interested Stakeholders

Marketeer, bagage persoon, reservatie persoon, immigratie agent, ...

Active Stakeholders

Passagier, check-in agent

Normal business flow

→ Een passagier presenteert zijn ticket en paspoort aan een check-in agent in de luchthaven. De agent zoekt de reservatie informatie op op basis van het ticket, het paspoort en valideert de reservatie. De check-in agent valideert het paspoort en geeft het terug aan de passagier. De agent valideert of de procedure richtlijnen EU175 zijn gevold. De agent voegt het vliegnummer toe aan de reservatie. De agent duidt op de zitplaatsen. De agent stelt veiligheidsvragen. De passagier beantwoordt de vragen. De agent checkt de bagage in. De agent print en toont de boarding pass aan de passagier. De agent wenst de passagier een fijne reis toe.

Exception business flow

Als de passagier geen geldig ticket zou hebben, dan licht de agent toe dat het ticket ongeldig is. De passagier krijgt de mogelijkheid om een nieuwe vlucht te boeken.

Outcome

De passagier is geregistreerd en ingecheckt op de vlucht, de bagage is toegestemd op de vlucht, de zitplaats is aangewezen, en de passagier is in bezit van een boarding pass.

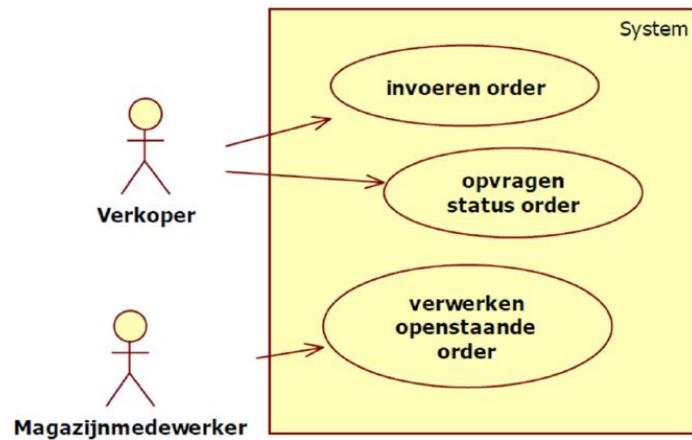
UML use case diagram



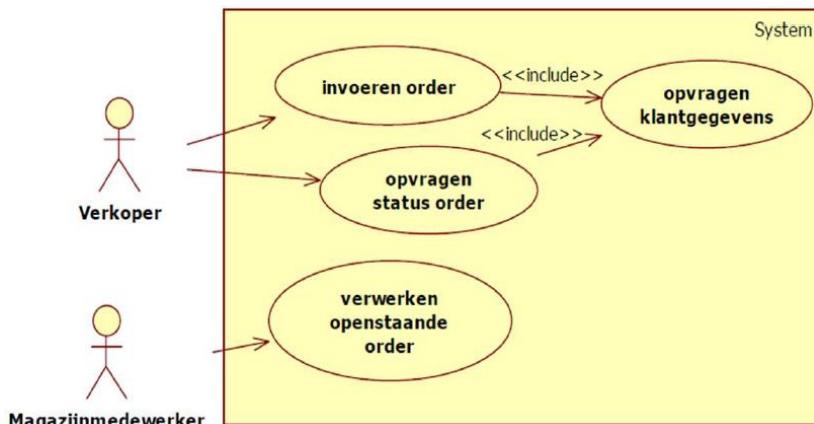
- Een figuur representeert een rol die wordt beoefend door een acotr
 - Een lijn connecteert de actor met een use case waarin die meespelen. (Gebruik cardinaliteit)
 - Een bijkomende verantwoordelijkheid wordt genoteerd door een →
Een supervisor heeft alle verantwoordelijkheden van een Salesperson, maar kan ook ...
 - De stippellijnen staan voor verschillende afhankelijkheden:
 - <<equivalent>> identieke activiteiten en een identieke flow – maar end users zien er verschillen in
 - <<extends>> al de activiteiten van de base use case worden ook in de extension use case uitgevoerd, de extension use case voegt nog enkele toegevoegde activiteiten toe of past de huidige activiteiten een beetje aan.
 - <<includes>> wanneer een case een subcase heeft, dan worden de activiteiten van de subcase meerdere keren herhaald bij het uitvoeren van de case.
 - <<requires>> Als de follower een leader vereist, dan moet de leader eerst worden afgehandeld voor dat de follower kan worden uitgevoerd.
 - <<resembles>> Twee use cases kunnen heel erg op elkaar lijken, maar toch verschillende activiteiten hebben.
-
- Een use case is verantwoord als het voor de gebruiker een waardevolle functionaliteit beschrijft. In een use case wordt beschreven wat een gebruiker met een systeem kan doen.
 - Waarom een use case model?
 - Om de scope van het systeem te bepalen
 - Als startpunt voor het modelleren
 - Voor de plannering
 - Als eenheid voor testen
 - Als eenheid voor gebruikershandleiding

Oefening

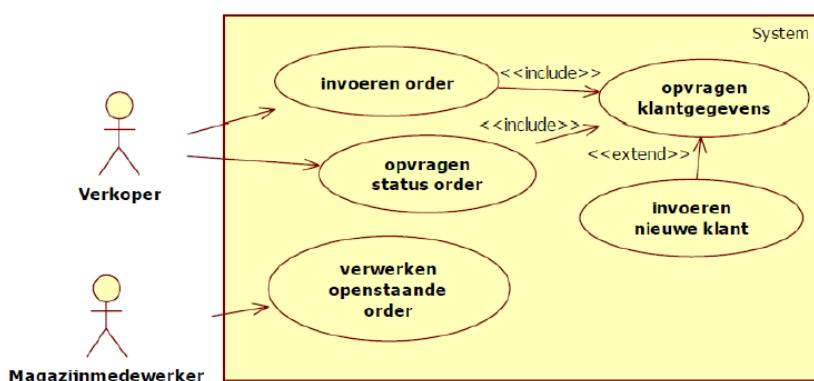
Een verkoper belt een klant met een aanbieding of de klant belt zelf voor een bestelling. De verkoper moet in het systeem de gewenste order kunnen vastleggen. In het magazijn moet een medewerker alle orders kunnen opvragen. Voor de goederen die op voorraad zijn maakt hij de bestelling in orde en laat een adressticker afdrukken. Soms belt een klant om te vragen naar de status van de order.



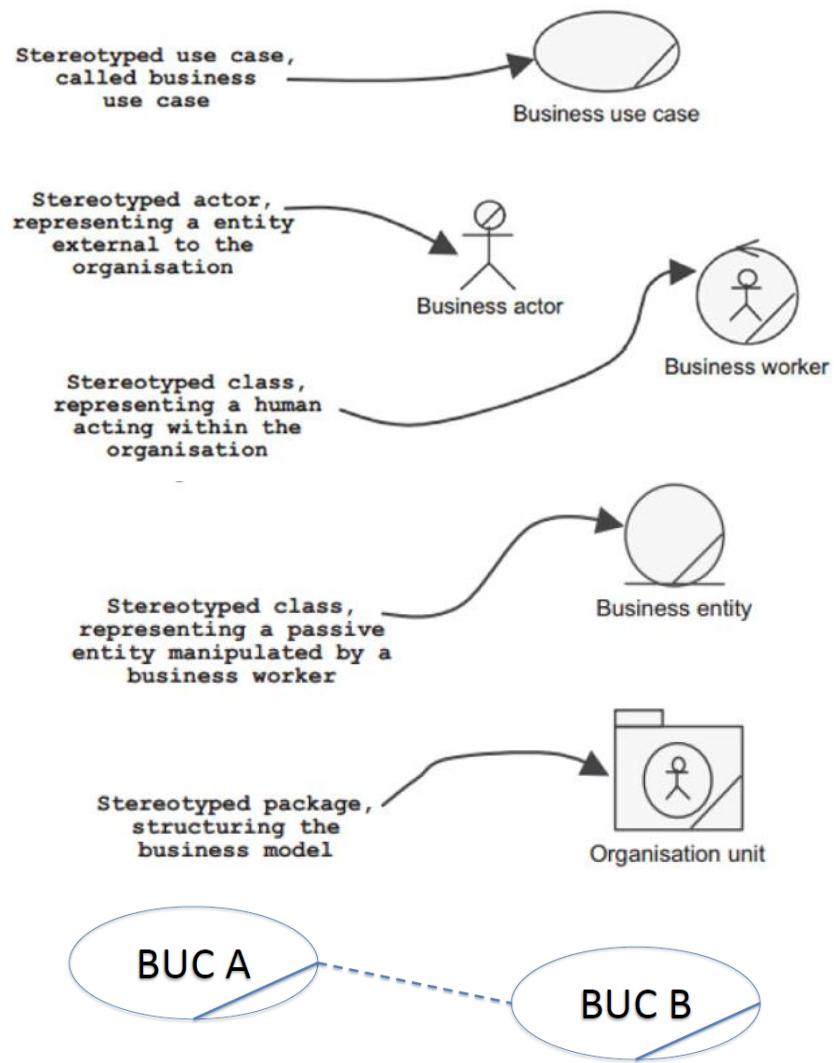
De actor voert de achternaam van de klant in. Het systeem toont alle klanten met die achternaam. De actor kiest de juiste klant. Het systeem laat alle details van de klant zien zodat de actor kan zien of hij de juiste klant voor zich heeft. De actor voert alle gewenste producten en de aantallen in en bevestigt het order.



Invoeren nieuwe klant: de actor voert de achternaam, voorletters, het adres en de geboortedatum van de klant in.



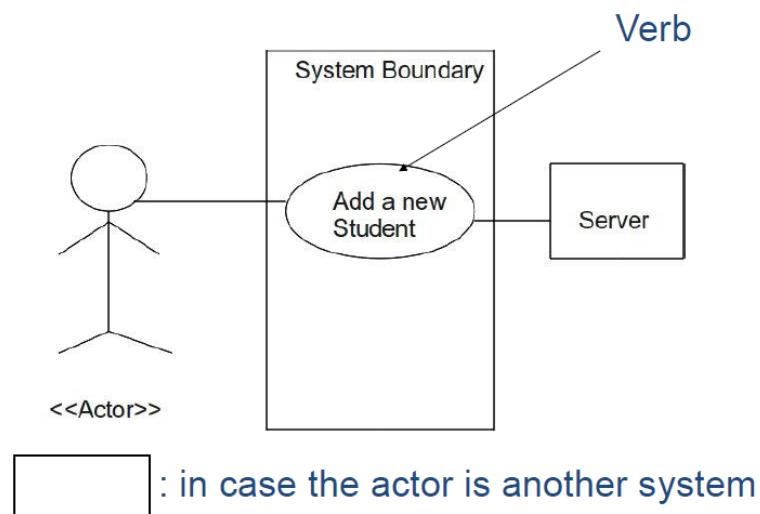
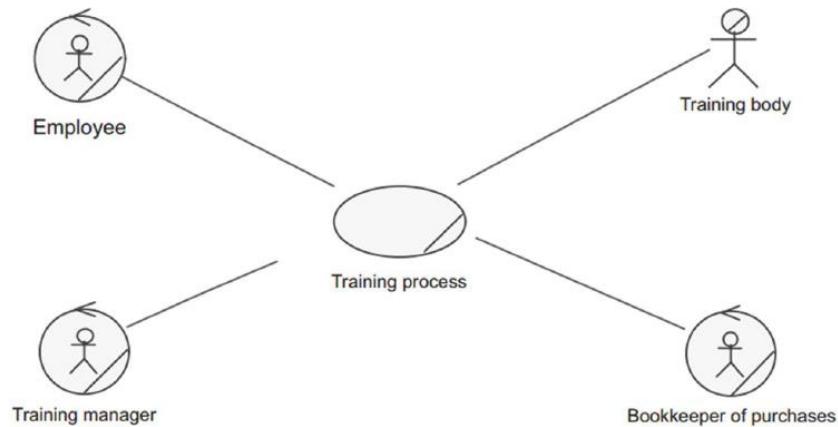
Business use case Model or Diagram

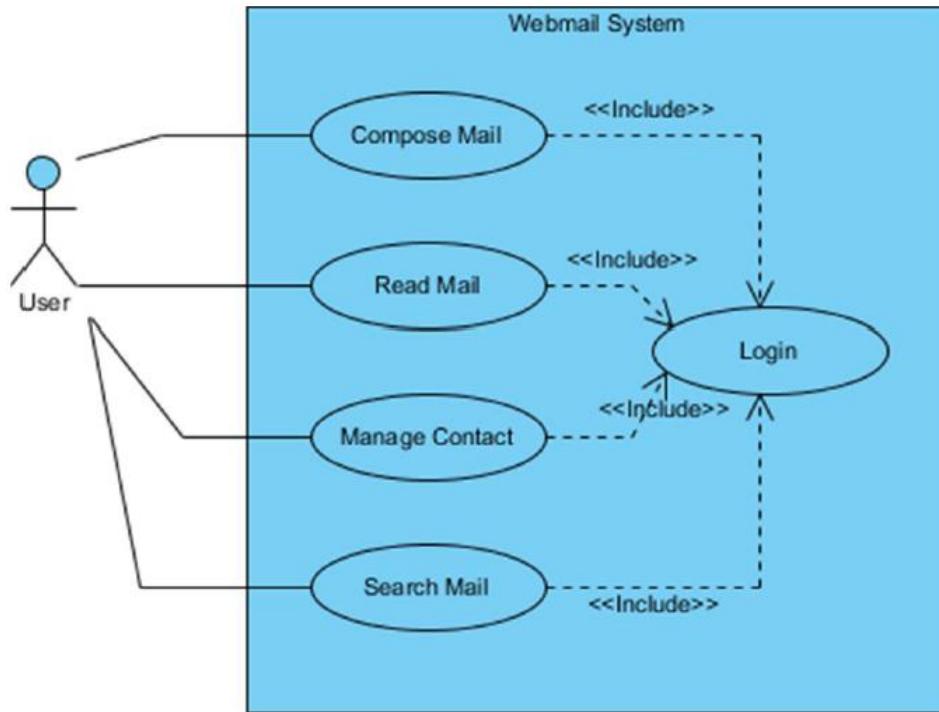


	Include	extend
Use case A	Can not without B	Can exist without B. Does not know that B exists
Use case B	Does not know which UC is calling ...	Knows to which UC it belongs

Example 1

1. The training process is initialized when the training manager receives a training request on behalf of an employee. This request is acknowledged by the person in charge who qualifies it and then forwards his or her agreement or disagreement to the person who is interested.
2. In the case of agreement, the person in charge looks in the catalogue of registered courses for a training course, which corresponds to the request. He or she informs the employee of the course content and suggests a list of subsequent sessions to him or her. When the employee has reached a decision, the training manager enrolls the entrant in the session with the relevant training body.
3. If something crops up, the employee must inform the training manager as soon as possible in order to cancel the enrolment or application.
4. At the end of the employee's training, he or she must submit an assessment to the training manager on the training course that he or she completed, as well as a document proving his or her attendance.
5. The training manager then checks the invoice that the training body has sent him or her before forwarding it to the bookkeeper of purchases.





Example 2

De klant haakt de slang af van een bepaalde brandstofpomp. Door het afhaken weet het systeem welke brandstofpomp moet functioneren. Het systeem activeert die bepaalde brandstofpomp, toont de prijs per liter van de brandstof, zet de tellers van getankte hoeveelheid en bedrag op 0 en start de compressor die deze brandstofslang onder druk zet. De klant drukt de handgreep in, de brandstof begint te lopen. Een debietmeter meldt aan het systeem elke centiliter die er getankt wordt. Het systeem toont de hoeveelheid en het bedrag. Een sensor aan de handgreep stelt eventueel vast dat de brandstofftank vol is en sluit een klepje in de handgreep: dit proces verloopt zonder tussenkomst van het informatiesysteem. De gebruiker kan ook zelf de handgreep lossen en zo stoppen met tanken. In beide gevallen geeft de debietmeter geen signalen meer aan het systeem. Als de klant de brandstofslang terug inhaakt wordt de brandstofpomp op niet actief gezet, en de gegevens van de tankbeurt komen op het scherm van de uitbater. De betaling wordt hier niet behandeld.

Actor(s)	De klant (persoon die tankt), uitbater van het tankstation
Precondition(s)	<ul style="list-style-type: none"> a) De brandstofslang is gesloten/ingehaakt b) De wagen staat in het bereik van de brandstofpomp met slang c) De motor van de wagen staat uit d) In de brandstofpomp is voldoende brandstof aanwezig e) De brandstoffank van de wagen is leeg of niet volledig vol f) De brandstofpomp is niet buiten dienst
Main scenario	<ol style="list-style-type: none"> 1) De klant⁴ haakt de brandstofslang af van een bepaalde brandstofpomp 2) Het systeem activeert die bepaalde brandstofpomp 3) Het systeem toont op die brandstofpomp de prijs per liter van de brandstof 4) Het systeem zet de tellers en het bedrag op 0 5) Het systeem start de compressor 6) De compressor zet de brandstofslang onder druk 7) De klant drukt de handgreet in 8) De brandstof begint te lopen 9) De debietmeter meldt aan het systeem elke centiliter die getankt wordt 10) Het systeem toont de getankte hoeveelheid en het bedrag. 11) De gebruiker lost de handgreet 12) De sensor stelt vast dat de brandstoffank vol is 13) De sensor sluit het klepje in de handgreet bij een volle tank (geen deel van het systeem) 14) De debietmeter geeft nu geen signalen meer aan het systeem 15) De klant haakt de brandstofslang terug in de brandstofpomp 16) De brandstofpomp wordt op niet actief gezet 17) De gegevens van tankbeurt komen op scherm van de uitbater.
Post condition(s)	Brandstoffank van betreffende wagen is gevuld

Concept	Frequency
Bedrag	2
Brandstof	2
Brandstofpomp	5
Brandstofslang	3
Brandstoffank	1
Centiliter	1
Compressor	2
Debietmeter	2
Druk	1
Gegevens	1
Handgreet	3
Hoeveelheid	1

Concept	Frequency
Klepje	1
Liter	1
Nul → 0	1
Prijs	1
Scherm	1
Sensor	2
Signalen	1
Systeem	7
Tank	1
Tankbeurt	1
Tellers	1

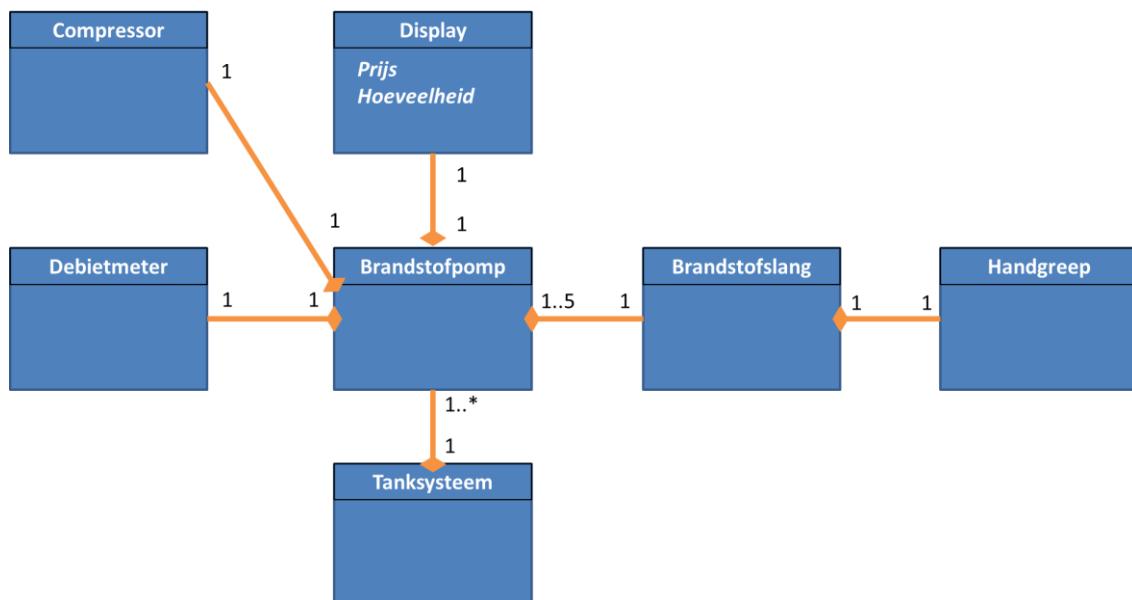
⁴ De klant, de gebruiker, de uitbater zijn actoren. Deze zelfstandige naamwoorden worden dus niet onderlijnd.

Concept	Freq.	Comments
Bedrag	2	Dit is het totaal bedrag. Dit wordt samengesteld door de hoeveelheid (de aantal getankte liter brandstof) maal de eenheidsprijs. Geen essentieel begrip
Brandstof	2	Een brandstof is een energiedrager die organisch is van aard. De vrijkomende energie kan gebruikt worden om warmte, licht, en kinetische energie te leveren. De brandstof behoort niet tot het systeem. Het totale systeem moet ook functioneren zonder brandstof. (cf. Voedsel microgolfoefening)
<u>Brandstofpomp</u>	5	Een apparaat dat wordt gebruikt om brandstof te verkopen. Het bestaat uit een display, een brandstofslang met handgreep, een compressor, een debietmeter. Dit is een essentieel begrip
<u>Brandstofslang</u>	3	De flexibele buis waardoor de brandstof van de brandstofpomp naar de brandstofftank wordt gepompt. Dit is een essentieel begrip
<u>Brandstofftank</u>	1	Reservoir voor de brandstof dat zich in de wagen bevindt. Dit is geen essentieel begrip
<u>Centiliter</u>	1	Meeteenheid van de getankte brandstof. Hiervoor kan ook liter gebruikt worden. Dit begrip is dus geen essentieel begrip
<u>Compressor</u>	2	Onderdeel van de brandstofpomp om de brandstof op een vooraf ingestelde druk te brengen om alzo de brandstof vanuit de brandstofpomp te transporteren naar de brandstofftank. Dit is een essentieel begrip
<u>Debietmeter</u>	2	Dit is een meetinstrument waarmee in de brandstofslang de doorstroming van de brandstof per tijdseenheid (het debiet) gemeten kan worden. Dit is een essentieel begrip
<u>Druk</u>	1	Is een grootheid. Dit is de kracht die uitgeoefend wordt op de brandstof per eenheid. Cf. compressor
<u>Gegevens</u>	1	Verzamelnaam voor o.a. de eenheidsprijs van de brandstof, de totale getankte hoeveelheid, het totaal bedrag, etc. Dit is geen essentieel begrip
<u>Handgreep</u>	3	Dit is een onderdeel van de brandstofslang, nodig om te kunnen tanken. Dit is een essentieel begrip
<u>Hoeveelheid</u>	1	De hoeveelheid getankte vloeistof. Dit is een essentieel begrip
<u>Klepje</u>	1	Zie ook sensor Geen essentieel begrip (zie tekst van de case)
<u>Liter</u>	1	Meeteenheid om de <u>hoeveelheid</u> brandstof weer te geven. Dit is geen essentieel begrip, cf. hoeveelheid
<u>Nul → 0</u>	1	Initiële waarde van de tellers bij het starten van een tankbeurt. Dit is geen essentieel begrip
<u>Prijs</u>	1	Dit is de <u>eenheidsprijs</u> (prijs per liter van de brandstof). Dit is een essentieel begrip
<u>Scherm</u>	1	De <u>display</u> waarop de gegevens zoals prijs per liter, getankte hoeveelheid, totaal bedrag, etc. worden getoond. Dit is onderdeel van de brandstofpomp. Dit is een essentieel begrip

Sensor	2	Onderdeel van de handgreep die vaststelt of de brandstofftank vol is en dan het klepje sluit. Zie ook klepje. Geen essentieel begrip (zie tekst van de case)
Signalen	1	Cf. debietmeter. Dit is geen essentieel begrip
Systeem	7	Het volledige systeem dat bestaat uit een of meerdere brandstofpompen. De term die zal gebruikt worden is <u>tanksysteem</u> . Dit is een essentieel begrip
Tank	1	Zie brandstofftank
Tankbeurt	1	De actie van het tanken. Dit is geen essentieel begrip
Tellers	1	Ijkwaardige meters die het volume of de hoeveelheid van de getankte brandstof meet en dit weergeeft. Dit zit vervat in het begrip "hoeveelheid". Dit is geen essentieel begrip

De domeinklassen zijn:

- Brandstofpomp
- Brandstofslang
- Compressor
- Debietmeter
- Handgreep
- Hoeveelheid
- Prijs
- Display
- Tanksysteem



Domain model

- Abstracte begrippen waarmee objecten uit het domein met eigenschappen en gedrag kunnen worden weergegeven
- In UML-diagrammen wordt de onderlinge samenhang tussen de begrippen weergegeven
- Vormen het fundament voor later te maken ontwerpklassen

How to find domainclasses

- Gebruik
 - Notities van vereisten
 - Use Case beschrijvingen
- Zoek
 - Zelfstandige naamwoorden
 - Werkwoorden
 - Bijvoeglijke naamwoorden
 - Attributen (a heeft een b en een c, a is opgebouwd uit een b en een c, a bestaat uit een b en een c)
- Conclusie
 - Geen garantie voor het vinden van alle domeinklassen

Example microwave

Actor(s)	Kok
Precondition(s)	Oven is inactief, stekker steekt in stopcontact. Deur is dicht. Voedsel binnen handbereik.
Main scenario	<ol style="list-style-type: none">1. Gebruiker opent deur.2. Lampje gaat branden.3. Gebruiker doet voedsel in oven.4. Gebruiker sluit deur.5. Lampje gaat uit.6. Gebruiker drukt 1x op knop voor werktijd van 60 sec. in te stellen.7. Lampje gaat aan.8. Microgolfbus gaat aan.9. Werktijd wordt zichtbaar op display.10. Werktijd wordt verminderd en in het venster zichtbaar gemaakt. Als werktijd om is:<ul style="list-style-type: none">11. Lampje gaat uit.12. Piepsignaal klinkt.13. Gebruiker opent deur en haalt voedsel eruit.14. Lampje gaat aan.15. Gebruiker sluit deur.16. Lampje gaat uit.
Post condition(s)	Voedsel is opgewarmd.

Begrip	#	Commentaar
Deur	5	Deur is een essentieel begrip
Display		Display is een essentieel begrip
Knop		Knop is een essentieel begrip
Lampje	6	Lampje is een essentieel begrip
Magnetronbuis	2	Magnetronbuis is een essentieel begrip
Oven		Oven is de kast die het geheel bij elkaar houdt, en dus essentieel
Piepsignaal		Piepsignaal is een essentieel begrip
Seconden		Eenheid waarin werktijd gemeten wordt
Venster		Is hetzelfde als display
Voedsel	2	Voedsel behoort niet tot het systeem. Oven moet ook functioneren zonder voedsel
Werkijd (= klok)	4	Essentieel voor de werking. => dus ook een klok nodig

→ Klasse is echter niet gelijk aan object

Een klasse is een beschrijving voor gelijksoortige objecten

Een object is een ding dat volgens de beschrijving gemaakt is

Één klasse kan veel objecten maken

Naam van domeinklasse	
Alarm	Voor het piepsignaal
Buis	Voor de magnetronbuis
Deur	
Display	
Knop	
Klok	Voor de werktijd
Lamp	
Oven	Voor het hele systeem

→ Associatie

Structurele relatie tussen 2 klassen

Wordt aangegeven door een doorgetrokken lijn tussen de 2 klassen

Conventie is dat namen van links naar rechts worden gelezen, maar kan ook van rechts naar links (zie pijltje)



→ Compositie

In deze relatie mag een onderdeel altijd maar tot één geheel behoren

Een deelobject kan niet zelfstandig verder bestaan als het object waar het deel van uitmaakt verdwijnt (niet gedeelde associatie)

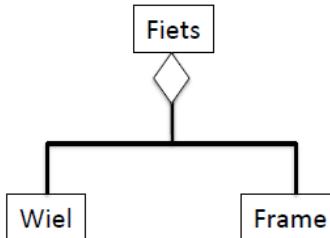
Gebruik van compositierelatie heeft de voorkeur boven de aggregatierelatie



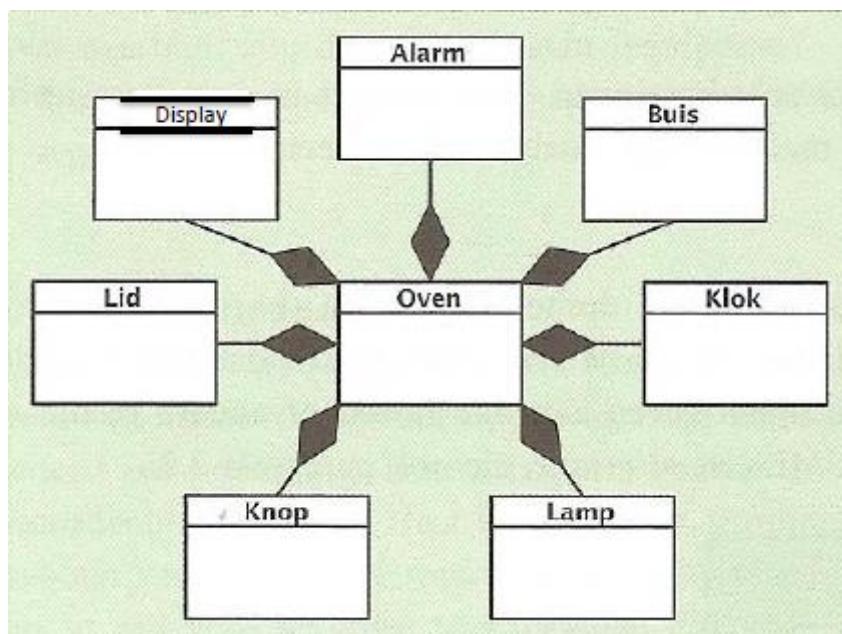
→ Aggregatie

Geeft aan dat één of meer klassen deel uitmaken van een andere klasse

Een gedeelde associatie (maakt ook deel uit van andere)

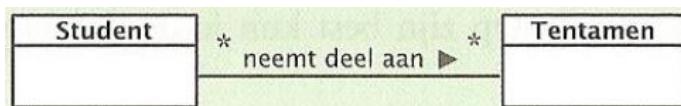


Fiets is voor een wielrenner niet één geheel.
Weersomstandigheden bvb bepalen de keuze van
de wielen die onder het frame gezet worden.



→ Multipliciteiten

De multipliciteit is het aantal instanties van een bepaalde klasse dat bij een associatie betrokken is in relatie tot precies één instantie van een andere klasse.



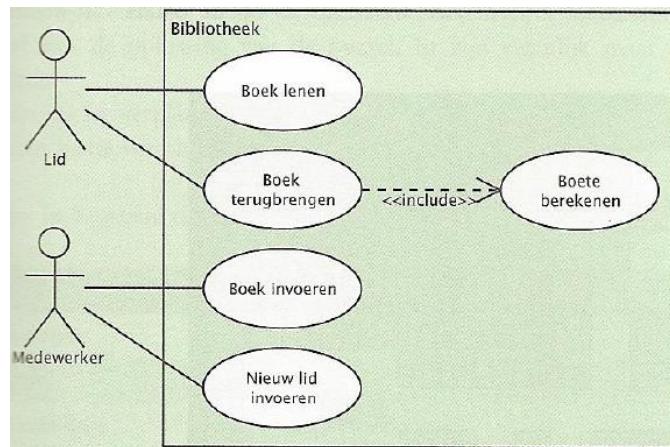
Elk student neemt deel aan een onbepaald aantal tentamens.

Aan elk tentamen wordt deelgenomen door een onbepaald aantal studenten.

Multipliciteit	Betekenis
1	Precies 1
2	Precies 2
1..5	Van 1 tot en met 5
6;7	6 of 7
1..*	Onbepaald aantal, maar ten minste 1
0..*	Onbepaald aantal, eventueel 0 (zelfde als *)
*	Onbepaald aantal, eventueel 0
0..1	0 of 1 (zelfde als 0,1)

Exercise: the library

"Van sommige boeken bezit de bibliotheek meer dan één exemplaar. Alleen mensen die lid zijn van de bibliotheek kunnen boeken lenen, en wel voor een periode van maximaal drie weken. Geleende boeken moet je binnen die periode terugbrengen, zo niet dan krijg je als lener een boete van 10 cent voor elke dag dat de uitleentermijn is verstreken. De bibliotheek moet een computersysteem krijgen dat alle uitleningen vastlegt. Elk lid heeft een pasje met daarop zijn/haar gegevens. Een scanner kan deze gegevens lezen. In elk boekexemplaar zit een streepjescode die uniek is"



Begrip	Commentaar
Bibliotheek	
Boek	
Boekexemplaar	Zelfde als exemplaar
Boete	
Cent	Munteenheid van boete
Computersysteem	
Dag	Tijdseenheid voor boeteberekening
Exemplaar	
Gegevens	Bedaald wordt gegevens die op pasje staan
Lener	Zelfde als lid
Lid	
Mens	Zelfde als lid
Periode	Zelfde als uitleentermijn
Scanner	
Uitleentermijn	
Uitlening	Het gaat hier om gegevens die bij de uitlening horen: lid boek, uitleentermijn en eventuele boete
Week	Eenheid voor uitleentermijn

→ Domeinbegrippen bepalen:

Bibliotheek (verwijzing naar de hele organisatie, niet nodig in het model)

Dag, cent, boete (hangen nauw met elkaar samen)

Lener, lid, mens (synoniemen)

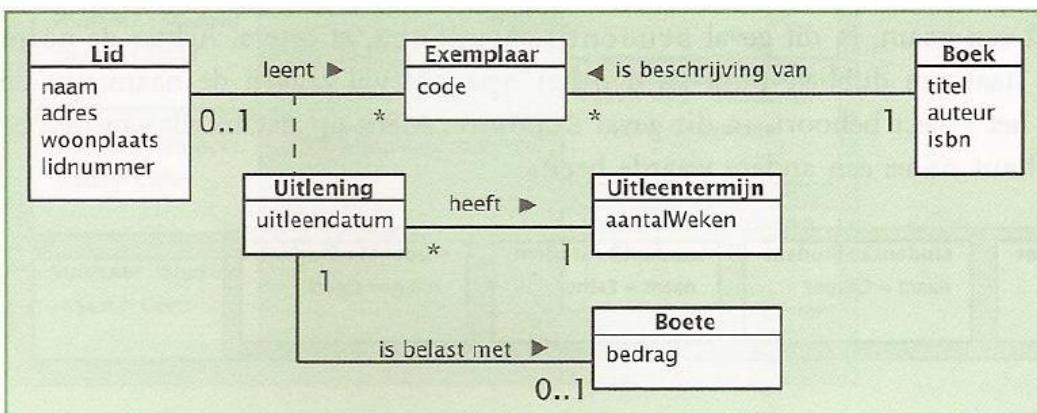
Computersysteem (systeem dat gemaakt moet worden)

Scanner (hardware en weinig waarschijnlijk dat het in de software een rol speelt)

Uitleentermijn, periode (zijn hetzelfde)

Week (eenheid voor de uitleentermijn)

Begrip	Commentaar
Boek	
Boete	Met als munteenheid cent; tijdseenheid dag
Exemplaar	
Gegevens (pasje)	Lidgegevens
Lid	
Uitleentermijn	In weken
Uitlening	



Requirements elicitation

What is requirements elicitation?

→ Elicitatie is:

Uitlokken of uitbrengen

Roepen of trekken

→ Requirements elicitation is een inspanning om informatie uit stakeholders en vakexperts te verkrijgen.

→ Elicitatie is geen stap of taak die je op een bepaald moment doet. Het is een set van technieken die je toepast, adequaat, tijdens de requirements fase.

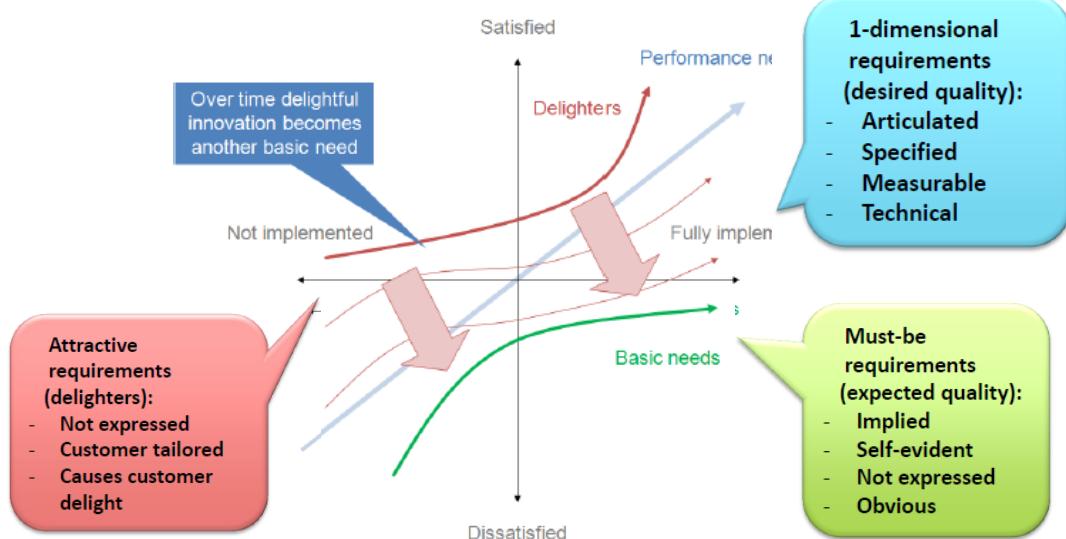
Why not requirements gathering?

Requirements gathering	Requirements elicitation
Like collecting sea shells	Like archeology
Take what you see	Planned, deliberate
More reactive, less proactive	More proactive, less reactive

Kano model

→ Ontwikkeld door prof. Noriaki Kano in de jaren 80

→ Om u te helpen proactief te ontdekken, te classificeren, en de integratie van de 3 soorten eisen die de klanttevredenheid beïnvloeden



Behoeften van de klant kunnen worden ingedeeld voor analyse door middel van een vragenlijst. Voor elke product functie worden een paar vragen geformuleerd die de klant kan beantwoorden op vijf verschillende manieren. De eerste vraag betreft de reactie van de klant als het product een bepaalde eigenschap heeft. De tweede betreft de reactie als het product de functie heeft.

Functional form of the question	Dysfunctional form of the question
If you have to wait for less than 2 minutes before the customer service rep answers your call, how do you feel?	If you have to wait for more than 2 minutes before the customer service rep answers your call, how do you feel?
1. I like it that way 2. It must be that way 3. I am neutral 4. I can live with it that way 5. I dislike it that way	1. I like it that way 2. It must be that way 3. I am neutral 4. I can live with it that way 5. I dislike it that way

Door de antwoorden te evalueren door onderstaande tabel, kunnen de producten worden geclassificeerd worden in zes categorieën.

A	Attractive	O	One Dimensional
M	Must-be	Q	Questionable
R	Reverse	I	Indifferent

Customer Requirement		Dysfunctional (Negative) Question				
		1.Like	2. Must be	3. Neutral	4. Live with	5. Dislike
Functional (Positive) Question	1.Like	Q	A	A	A	O
	2. Must be	R	I	I	I	M
	3. Neutral	R	I	I	I	M
	4. Live with	R	I	I	I	M
	5. Dislike	R	R	R	R	Q

Elicitatie technieken

- ➔ Het vinden van bewuste, onbewuste en onderbewuste vereisten
- ➔ Keuze hangt af van:
 - Risicofactoren
 - Ervaring van de requirements engineer
 - Tijd en budget
 - Beschikbaarheid van de stakeholders
 - Onderzoekstechnieken
 - Interviews, vragenlijsten
 - Creativiteit technieken
 - Brainstormen, verandering van perspectief, analogieën, rollenspel
 - Observatie technieken
 - Document-centric technieken
 - Systeem archeologie, perspectief gebaseerd lezen, hergebruik van vereisten
 - Ondersteunen van technieken
 - Mind mapping, workshops, prototyping, use case modellering, audio/video

Interviews

- ➔ Interviewen van belanghebbenden
 - Niet de enige techniek
 - Gebruikers weten niet alle requirements
- ➔ Open vragen (Hoe, waarom, wanneer, waar, wat, wie)
 - Meer kans om informatie te krijgen
- ➔ Gesloten vragen (Doe, is, kan, zou kunnen, zal, zou, ...)
 - Deze krijgen ja/nee antwoorden
- ➔ Gebruik antwoorden van vragen om nieuwe vragen te stellen
- ➔ Type interviews:
 - Gestructureerd interview: vooraf gedefinieerde lijst van vragen
 - Ongestructureerd interview: bespreek onderwerpen met behulp van open-end vragen
- ➔ Drie belangrijke doelstellingen:
 - Neem informatie op om erna als input bij de requirements analyse te gebruiken,
 - Ontdek de informatie van de geïnterviewde nauwkeurig en efficiënt,
 - Verzeker dat het onderwerp is onderzocht, dat er wordt geluisterd en gewaardeerd
- ➔ Het proces bestaat uit vier belangrijke stappen:
 - Planning en voorbereiding,
 - Interview sessie,
 - Consolidatie van de informatie,
 - Opvolgen
- ➔ Opmerkingen
 - Stel doelen en doelstellingen voor het interview
 - Heb achtergrond kennis van de materie
 - Over het domein maar ook over de geïnterviewde

Survey/Questionnaires

→ Survey vragen zijn er in twee soorten:

Gesloten vragen met ingeblikte antwoorden die gemakkelijk te analyseren zijn

Open vragen waarin nader wordt ingegaan maar vereisen interpretatie

→ Voorbereiden van de enquête

Bepaal het doel van de enquête,

Kies het publiek en steekproefgrootte

Schrijf de enquêtevragen

Test te enquêten vragen

→ Overweging van gebruik

Voordelen van de twee soorten vragen

→ I have written a journal before. Yes No Sometimes Ja / nee vragen

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1. Reading English is more important than speaking English	1	2	3	4	5

Rating scales

Reading English is more important than speaking English ☺ ☻ ☻

→ Give a score from 1 to 10 in the space below (1 = not at all; 10 = extremely well). Numeric scales

How well do you know the students in Group B? _____

Grammar is important in language learning because it:

→

- Provides students with rules
- Helps them to become more fluent
- Gives them a language to talk about language
- Helps them to self-correct
- Enables them to become more accurate
- Other (please specify and explain your response) _____

Guided items

→

Please provide comments on teaching grammar by completing the sentences:

Structured items

The thing I like best about teaching grammar is _____

The thing I like least about teaching grammar is _____

My most difficult challenge in teaching grammar is _____

Annex E: ISO 9126 Quality Characteristics

Functionality A capability of the software product to provide function which meet stated and implied needs when the software is used under specified conditions.

- Suitability The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives.
- Accuracy The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.
- Interoperability The capability of the software product to interact with one or more specified systems.
- Security The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.
- Compliance The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality.

Reliability The capability of the software product to maintain a specified level of performance when used under specified conditions.

- Maturity The capability of the software product to avoid failure as a result of faults in the software.
- Fault-tolerance The capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.
- Recoverability The capability of the software product to re-establish a specified level of performance and recover the data directly affected in case of a failure.
- Compliance The capability of the software product to adhere to standards, conventions or regulations relating to reliability.

Usability The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.

- Understandability The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.
- Learnability The capability of the software product to enable the user to learn its application.
- Operability The capability of the software product to enable the user to operate and control it.
- Attractiveness The capability of the software product to be attractive to the user.
- Compliance The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability.

Efficiency The capability of the software product to provide appropriate performance, relative to the amount of resources used under stated conditions.

- Time behavior The capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions.
- Resource utilization The capability of the software product to use appropriate amounts and types of resources when the software performs its functions under stated conditions.
- Compliance The capability of the software product to adhere to standards or conventions relating to efficiency.

Maintainability The capability of the software product to be modified. Modifications may include corrections, improvements or adaptations of the software changes in environments, and in requirements and functional specifications.

- **Analysability** The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified.
- **Changeability** The capability of the software product to enable a specified modifications to be implemented.
- **Stability** The capability of the software product to avoid unexpected effects from modifications in the software.
- **Testability** The capability of the software product to enable modified software to be validated.
- **Compliance** The capability of the software product to adhere to standards or conventions relating to maintainability.

Portability The capability of the software product to be transferred from one environment to another.

- **Adaptability** The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered.
- **Installability** The capability of the software product to be installed in a specified environment.
- **Co-existence** The capability of the software product to co-exist with other independent software in a common environment sharing common resources.
- **Replaceability** The capability of the software product to be used in place of another specified software product for the same purpose in the same environment.
- **Compliance** The capability of the software product to adhere to standards or conventions relating to portability.

Documentation of requirements using NL

Natural language

- ➔ Iedere stakeholder verstaat het
- ➔ Het is universeel, wordt gebruikt om de gewenste omstandigheden te omschrijven
- ➔ Het is vaak dubbelzinnig, interpreteerbaar

Language effects

- ➔ Het gaat om de perceptie en de transformaties van de documentatie
- ➔ Wat is er gezegd VS. Wat wordt er bedoeld
- ➔ Het is de verantwoordelijkheid van de requirement engineer om te bepalen wat er bedoeld wordt
- ➔ Vijf meest relevante transformaties:
Normalisatie, universele kwantoren, onvolledige gespecificeerde proces woorden, zelfstandige naamwoorden zonder index-referentie, Onvolledig gespecifieerde voorwaarden

Exercise – new clock radio

Topic – New clock radio

The development of the innovative alarm clock and music system shall raise the profit of the company by 10% within 3 months after delivery to the wholesale dealer.

The clock shall produce no tic sound. Because people don't like to be woken up abruptly, the new clock radio shall wake up the sleeping people more smoothly than any other alarm clock.

The luminosity of the lighted digits shall be changeable continuously.

The clock radio shall be small and lightweight, because the users also want to use it on travel. Therefore the clock radio shall be usable with different power voltages. To achieve a minimum weight the alarm clock must not contain any iron parts.

The alarm clock shall be able to wake its environment with music or a wake-up call.

The clock shall be very easy to use. The snooze key shall be easily accessible. The clock shall withstand a heavy blow from a sleeper and a fall from 50 cm.

The clock shall be very low in energy consumption. The system may only be operated with batteries of type 3434.

Dealing with transformations

Vijf stappen om requirements te schrijven volgens de template:

→ Stel de mate van de verplichting vast

Vaststelling van aansprakelijkheid

Sterk aanbevolen (Je zou moeten...)

Toekomst

Zal, zou moeten en wil

→ Bepaal de kern van de requirement

De kern van de requirement is de vereiste functionaliteit

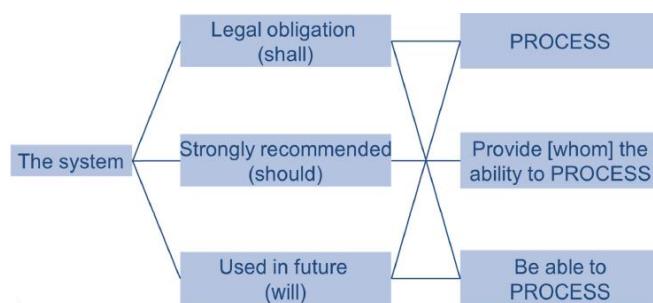
Processen zijn activiteiten of evenementen: beschreven a.d.h.v. werkwoorden

→ Categoriseer de systeemactie

De manier waarop het systeem werkt is nauw verbonden met de proces woorden

Drie soorten systeem activiteiten:

- Autonomoos systeem activiteit : het systeem voert het proces zelf uit
- Interactie van de gebruiker: het systeem biedt de gebruiker proces-functionaliteit
- Interface requirement: het systeem voert het proces uit afhankelijk van een derde factor.



De eerste drie stappen ...

Het geplande systeem moet een print optie bieden

- De optie is onmisbaar voor ons
- 'Print' is het proces woord
- Het systeem geeft de administrator de optie om ...

'Het systeem **zal** de **administrator** de keuze geven om te **printen**'

→ Voeg het object toe

Verdere elementen zijn nodig om het compleet te maken

Een aanvullend karakter van het proces woord

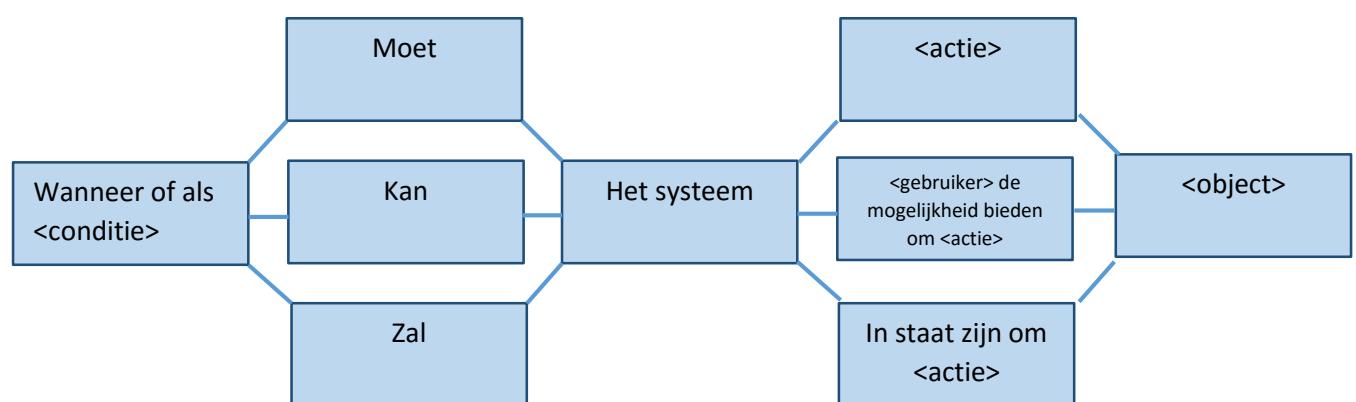
Wat, waar

'Het systeem **zal** de **administrator** de keuze geven om een error bericht te **printen** naar de netwerk printer'

→ Voeg de conditie toe

De functionaliteit is alleen gegeven onder een bepaalde logische of tijdelijke condities

'Als een error bericht gegenereerd is, **zal** het systeem de keuze geven aan de **administrator** om een error bericht naar de netwerk printer te **printen**'



Requirements document template

• Section 1: Introduction

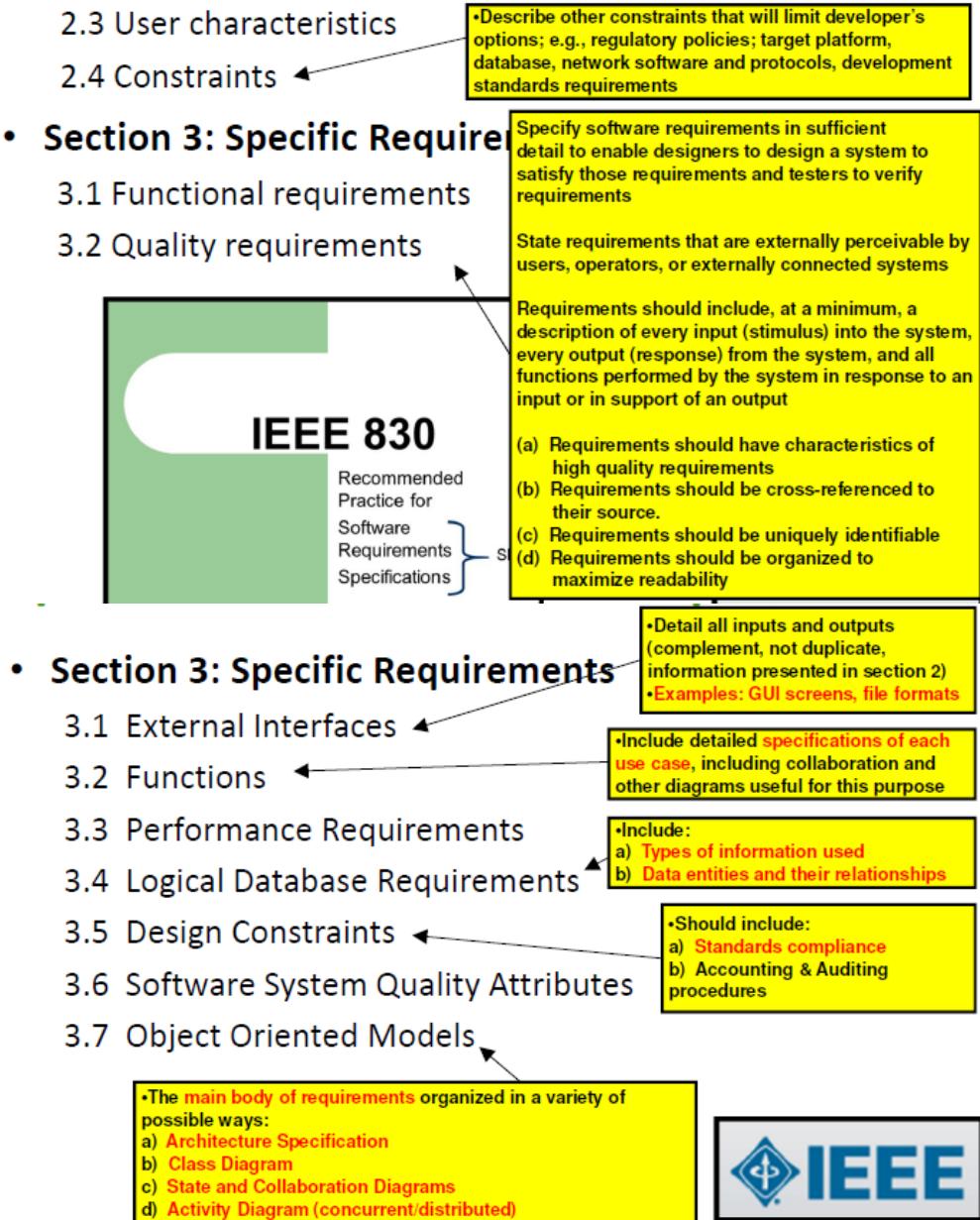
- 1.1 The purpose of the product
 - 1.2 Scope of the product (context diagram)
 - 1.3 Definitions, acronyms and abbreviations
 - 1.4 References
 - 1.5 Overview
- IEEE**
- Describe purpose of this SRS
 - Describe intended audience
 - Identify the software product
 - Enumerate what the system will and will not do
 - Describe user classes and benefits for each
 - Define the vocabulary of the SRS (may reference appendix)
 - List all referenced documents including sources (e.g., Use Case Model and Problem Statement; Experts in the field)
 - Describe the content of the rest of the SRS
 - Describe how the SRS is organized

• Section 2: Overall description

- 2.1 Product perspective
 - General overview, use cases, interfaces
 - 2.2 Product functions
 - A list of functionalities (incl. quality related) provided
 - Grouping should support a communication point
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies
 - 2.6 Apportioning of requirements
 - Requirements that may be delayed
- IEEE**

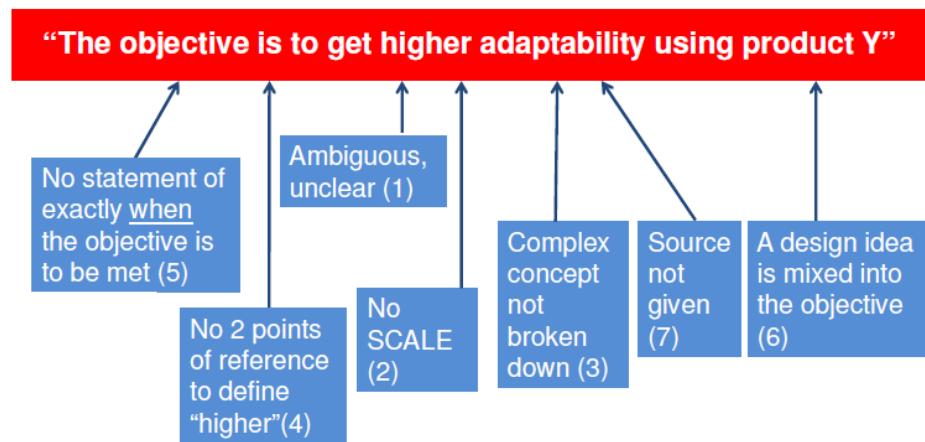
• Section 2: Overall description

- 2.1 Product perspective
 - General overview, use cases, interfaces
 - 2.2 Product functions
 - A list of functionalities (incl. quality related) provided
 - Grouping should support a communication point
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies
 - 2.6 Apportioning of requirements
 - Requirements that may be delayed
- IEEE**
- Summarize the major functional capabilities
 - Include the Use Case Diagram and supporting narrative (identify actors and use cases)
 - Include Data Flow Diagram if appropriate
 - Describe and justify technical skills and capabilities of each user class



Rules for quality requirements

1. Zij moeten ondubbelzinnig duidelijk zijn voor de beoogde lezer.
2. Zij zullen een schaal van de maatregel moeten specifiëren om het concept te definiëren.
3. Zij zullen complexe concepten opbreken in een reeks meetbare concepten.
4. Om "relatieve" termen als "hoger" te beschrijven, zullen zij ten minste twee referentiepunten op de gedefinieerde schaal specifiëren.
5. Zij moeten precies aan geven wanneer een kwaliteitsniveau beschikbaar is.
6. Zij zullen geen ontwerpideeën mengen in de specificatie van de doelstellingen .
7. De procesinput of "source" (contract, standaard marketingplan) van de requirement wordt gegeven.



Model based requirements documentation

Model

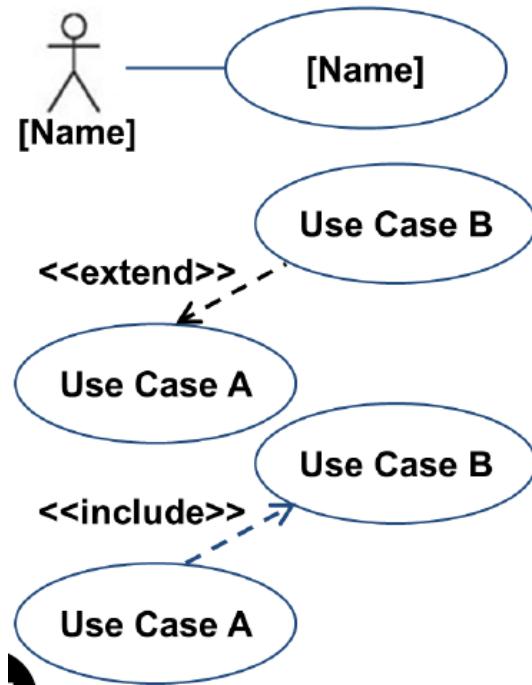
- Een model is een abstractie van de bestaande werkelijkheid of een plan
- 3 belangrijkste eigenschappen van modellen:
 - Vertegenwoordiging: werkelijkheid weergeven
 - Vermindering: verminder de weergegeven werkelijkheid
 - Pragmatisch: gebouwd voor een speciaal doel
- In RE worden er conceptuele modelleringstalen gebruikt
- Informatie is gemakkelijker en sneller te begrijpen via beelden
- Doelgerichte modelering van 1 perspectief

Most frequently used models

- Use Case Diagram (+description)
 - Representeert een users interactie met het systeem
- Entity relationship diagram
 - Beschrijft de data of informatie aspecten van een business domein
- Class diagram (OO)
 - Beschrijft de structuur van een systeem door het tonen van de systeemklassen, attributen, methoden en de relaties daartussen
- Data flow diagram (derived from context diagram)
 - Gedetailleerde functionaliteit
- Activity diagram
 - Gehele controle flow
- State chart
 - Toont het gedrag
- Sequence diagram
 - Toont hoe processen met elkaar omgaan en in welke volgorde

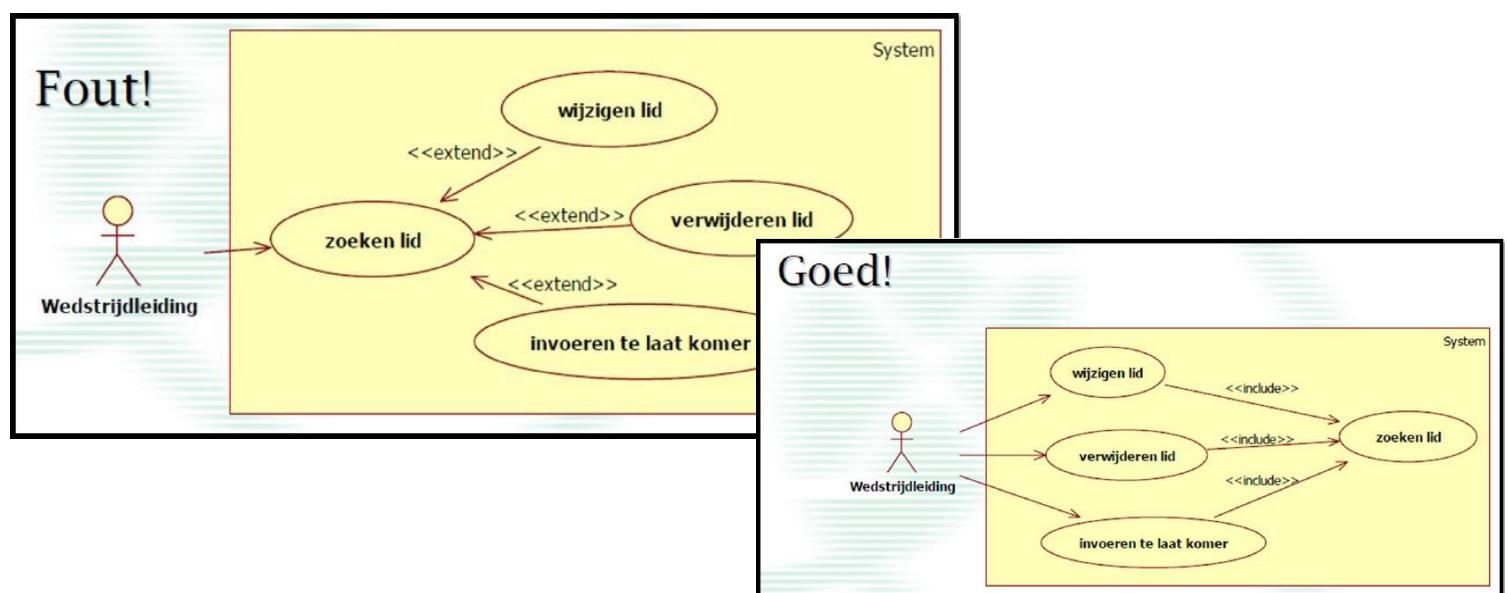
System use case diagrams

- Van een ander standpunt bekeken
- Toont relatie met de context
- Elementen:
 - Use cases, actoren, systeem grens, relaties tussen de elementen



System use case specifications

- Denk aan "What if scenario's"
- Een of meer use cases per event
- Men kan de requirements toevoegen aan iedere use case stap, om zo ontbrekende requirements te vinden.
- Veel templates beschikbaar
- Uitbereiding van de use case diagrammen door de belangrijke karakteristieke properties meer precies te beschrijven in natural language.
- Een opeenvolging van transacties in een dialoog tussen een gebruiker en het product met een bepaald resultaat.



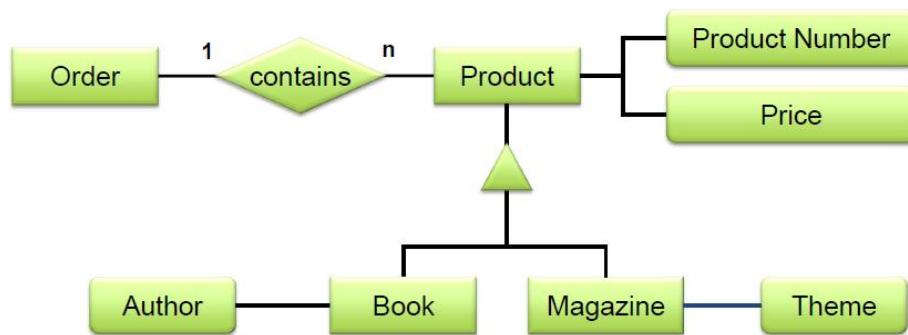
Data/structural requirements models

- De structuur van de data van het systeem documenteren, als het gebruik en de relaties binnen de systeemcontext
- Traditioneel: ERM – Entity relationship modellen
- Ook ondergebracht als Class models (UML)

Entity relationship model (ERM)

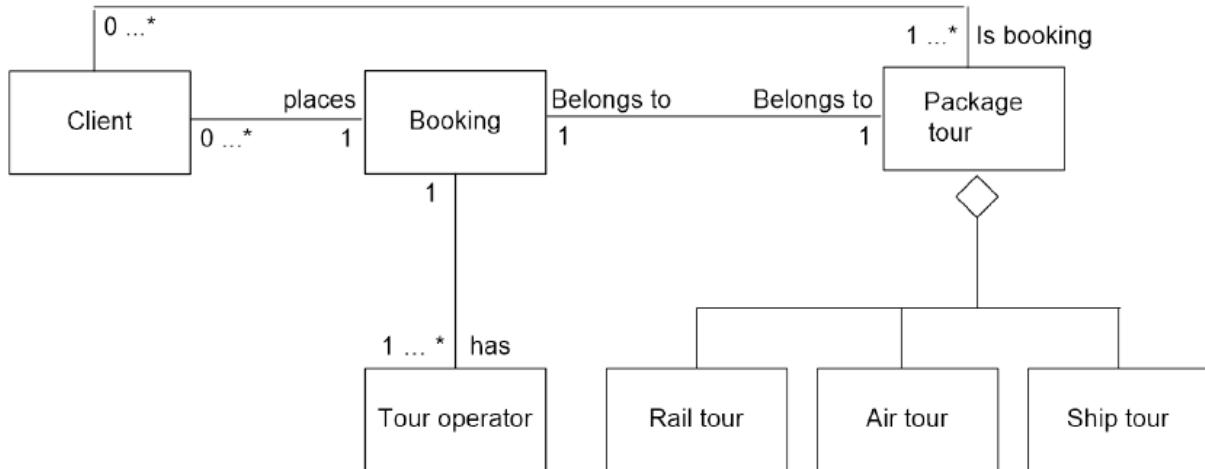
- Modelleert de structuur van data
- Niet voor functies, gedrag, controle
- Op grote schaal gebruikt voor relationele databases
- Verduidelijkt structuur, hiërarchie
- Begrijpelijk (communicatie tussen users)
- Typische modellering elementen:
Entiteiten, relatie typen, attributen

Symbol	Team	Voorbeelden
AUTO	Entiteit	Bediende, tafel, bestelling
KLEUR	Attribuut	Prijs, adres, grootte
△	Generalisatie	Voertuig {auto, vrachtwagen}
◇ BEVAT	Relatie	Een artikel behoort tot een bestelling
(min,max)	Cardinaliteit	(1,n), 1, (0,n)



Class model (UML)

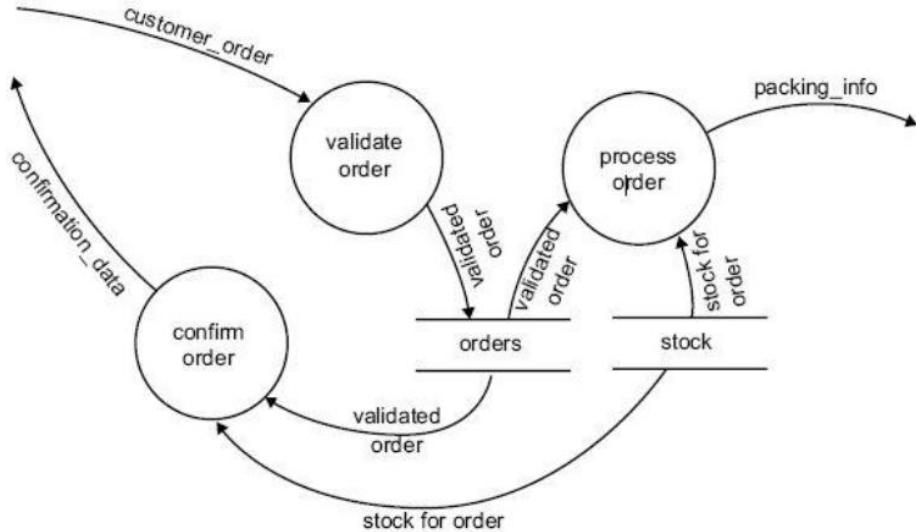
- Om statische structuren te modelleren
- Specifieke domein kennis
- Verschillende levels
 - Conceptueel, specificatie, implementatie
- Tools: UML modelling tool
- Veel gebruikte modellering elementen:
 - Klassen, associaties, aggregatie, compositie, generalisatie



Data Flow Diagram

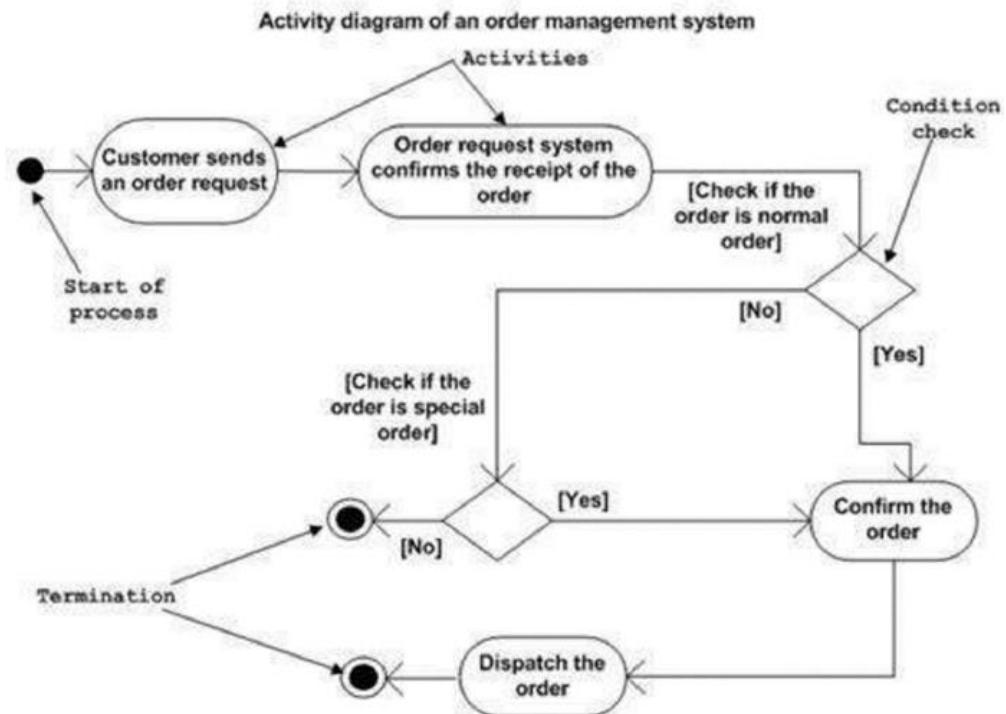
- Een data-flow diagram (DFD) is een grafische representatie van de “flow” van data d.m.v. een informatiesysteem
- Geeft geen informatie over de timing, sequenties of parallel gedrag
- Niveaus: hoog niveau diagram, laag niveau diagram
- Modellering elementen:

Processen, data flows, data stores, sources / sinks



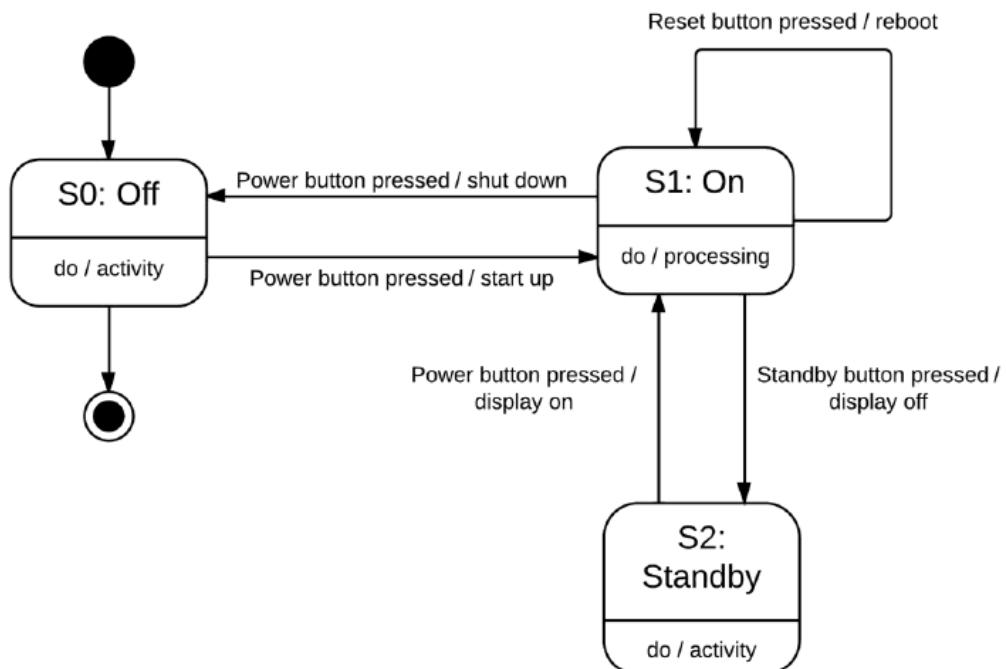
Activity diagram

- Focus op activiteiten van een process en de controle daartussen
- Beschrijft de volgorde van werkzaamheden
- Procedure als gevolg van individuele acties
- Threads kunnen parallel of voorwaardelijk zijn
- Vaak gebruikt om use cases te verduidelijken / scenario's detailleren
- Essentiële modellering elementen: acties, start- en eindnodes, controle flow, besluit knooppunten samenvoegen, hiërarchische elementen



- Gemakkelijk te leren, breed toepasbaar (use case details), kan complexe tijdgebonden procedures tonen
- Valkuil: het opgeven van te veel details, weinig praktische informatie in UML handleiding, ...

State Transition Diagram (STD)



- ➔ Beschrijft het gedrag van een systeem
- ➔ Alle mogelijke toestanden waar een systeem kan ingeraken
- ➔ Veranderingen staan als gevolg van een gebeurtenis
- ➔ Acties zijn korte effecten van transities
- ➔ Geschikt voor het modelleren van interfaces
- ➔ Geschikt voor simulatie en testen
- ➔ Kan moeilijk te lezen zijn voor buitenstaanders

Sequence diagrams

Tutorials pluralsight + youtube

Requirements validation and negotiation

Zie word doc. begrijpend lezen