

Requirements Engineering Inleiding

Requirements engineering

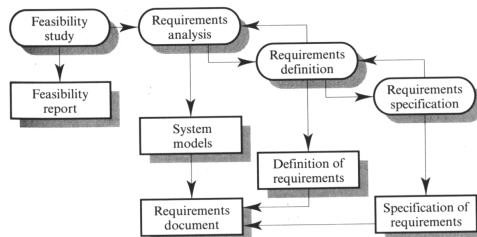
“Requirements engineering”

- het gebruik van een systematisch proces om de requirements en beperkingen van het te ontwikkelen systeem vast te stellen en te formuleren
- er wordt een onderscheid gemaakt tussen:
 - “**requirements definition**” = hoge niveau, abstracte beschrijving, in natuurlijke taal
 - vaak gebruikt voor een offerte, contract
 - lezers: gericht op management van klant en ontwikkelaar, gebruikers, systeem architecten,...
 - “**requirements specification**” = gestructureerd document met meer gedetailleerde (en meer technische) beschrijving
 - ook “functional specification” genoemd
 - lezers: ontwikkelaars, gebruikers, systeem architecten,...
 - “**software specification**” = nog meer detail dan de requirements specification
 - optioneel, indien nodig
 - basis voor design en implementatie
 - lezers: ontwikkelaars, systeem architecten

Requirements engineering

“Requirements engineering” proces

- alle activiteiten die leiden tot de productie van de requirements definition en requirements specification
 - “feasibility study” (haalbaarheidsstudie)
 - “requirements analysis” = het proces waarmee de requirements afgeleid worden
 - observatie van bestaande systemen, gesprekken met gebruikers,...
 - systeem modellen, prototypes,...
 - “requirements definition” = informatie van de analyse in een document neerschrijven
 - “requirements specification” = meer gedetailleerd document, eventueel terug naar vorige fasen



Requirements engineering

Requirements document

- = requirements definition + requirements specification
- gewenste kenmerken
 - mag enkel het externe gedrag van het systeem vastleggen
 - volledig en consistent
 - gemakkelijk te wijzigen
 - referentie voor het onderhoud van het systeem
 - structuur moet naast die van het design gelegd kunnen worden voor validatie
- generische structuur van een requirements document
 - inleiding: doel, situering in de organisatie / tegenover ander systemen,...
 - terminologie
 - systeem modellen: vb. data flow, object modellen,...
 - functionele requirements
 - niet-functionele requirements / beperkingen
 - verwachte evolutie van het systeem en zijn omgeving
 - requirements specification: meer detail over de functionele requirements

Validatie

- bewijzen dat het gewenste systeem gedefinieerd is (om fouten zo vlug mogelijk te ontdekken)
- te controleren aspecten van het requirements document
 - validiteit voor diverse gebruiker: kan het systeem doen wat verwacht wordt?
 - consistentie: geen conflicten tussen requirements
 - volledigheid
 - realisme: is het haalbaar? technisch, binnen de tijd,...
 - verstaanbaarheid
 - verifieerbaarheid
 - doorlopen van de requirements mogelijk via verwijzingen
 - aanpasbaarheid ("enduring <-> volatile" requirements), weinig referenties ...
- moeilijk uit te voeren
 - prototyping kan helpen
 - regelmatig "requirements reviews": manueel, meerdere lezers van het document

Requirements Analyse

Requirements analyse

Doelstelling "requirements analysis" / "elicitation"

- informatie over het applicatie domein en het systeem vergaren
- interactie met betrokken personen: gebruikers, opdrachtgever,...("stakeholder")
 - weten niet precies wat ze willen
 - formuleren dit niet nauwkeurig, in eigen terminologie
 - verschillende personen hebben verschillende belangen (wensen, in organisatie)
 - achteraf duiken bijkomende requirements en betrokken personen op
- activiteiten tijdens "requirements analysis"
 - verstaan van het applicatie domein
 - verzamelen van requirements
 - classificatie/structurering van requirements (ev. productie van systeem modellen)
 - behandelen van conflicten tussen requirements
 - prioriteiten stellen tussen requirements
 - validatie

Viewpoint-oriented analysis

- verschillende betrokken personen hebben verschillende "viewpoints" op het systeem: belangen, verwachtingen,...
- "viewpoints" worden gebruikt bij het structureren van de informatie bv. overlap ontdekken
- verschillende methoden voor analyse geven een andere betekenis aan het begrip "viewpoint"
 - bron of bestemming ("consumer") van data
 - representatie model = systeem model (data flow, state diagram,...)
- ➡
 - extern "viewpoint" = degene die diensten van het systeem verwacht
 - vb. VORD [Kotonya and Sommerville, 1992]
 - direct rekening houden met gebruikers
 - belangrijk voor interactieve systemen
 - externe kijk is goed voor structurering, ook voor niet-functionele requirements
 - conflicten tussen requirements zijn niet altijd eenvoudig vast te stellen / te behandelen

Method-based requirement analysis

- toepassen van een gestructureerde methode voor analyse (en ev. design)
- methode omvat meestal
 - een procesmodel dat de te volgen activiteiten beschrijft
 - notatie voor systeem modellen (object diagram, data-flow diagram,...)
 - design richtlijnen (vb. aantal deelobjecten van een object)
 - beschreven structuur / formulieren / templates voor rapporten en het requirements document
- alle methoden produceren veel informatie en documentatie => ondersteuning door CASE-tools is gewenst
- voorbeelden
 - object-georiënteerde methoden
 - VORD [Kotonya and Sommerville, 1992]

De context van het systeem

- tijdens de analyse wordt het systeem afgelijnd tegenover manuele systemen en andere software systemen
- tekenen van blok-diagrammen van alle sub-systemen

Sociale en organisatorische factoren

- beïnvloeden de requirements analyse (zelfs diverse viewpoints)
- weinig methoden houden hiermee rekening
- vb. systeem dat taken van bepaalde personen minder belangrijk maakt...hun medewerking is moeilijk maar noodzakelijk tijdens de analyse
- etnografie = een socioloog die enige tijd op de werkplek vertoeft om het werk van de mensen te analyseren (kennis vergaren over o.a. het manuele systeem)
- mensen kunnen moeilijk zelf hun werk beschrijven
- etnografie + prototyping is nuttig!

Requirements definition + specification

Requirements definition

Requirements specification

Non-functional requirements (als speciaal geval van requirements specs)

- hoe deze verschillende vereisten neerschrijven (notaties)
- hoe voldoen aan eisen van requirements formulering als
 - nauwkeurig
 - ondubbelzinnig
 - verstaanbaar
 - verifieerbaar
 - niet te duur om de specs te formuleren

Requirements definition Requirements specification

Requirements definition

- abstracte beschrijving van het systeem, bestaande uit
 - functional requirements = de functionaliteit van het systeem
 - non-functional requirements = beperkingen waarmee het systeem moet omgaan
- best afzonderlijk van de meer gedetailleerde requirements specifications (meer technisch), de specs in een afzonderlijk deel of appendix van het document opnemen.
- moet verstaanbaar zijn door gebruikers, klanten
- notatie moet gebaseerd zijn op natuurlijke taal, formuleren en eenvoudige diagrammen
- werkwijze: eerst ideeën neerschrijven, daarna deze groeperen en structureren

Requirements definition (vervolg)

- notatie
 - **natuurlijke taal**, maar dit heeft als nadelen
 - duidelijkheid? ondubbelzinnigheid?
 - vermengen van verschillende soorten requirements
 - » vb. funct. en niet-funct.
 - » vb. user interface requirements vermengen met de functie
 - verschillende requirements worden als 1 requirement samengevat
 - structureren en groeperen van requirements is niet eenvoudig
 - **natuurlijke taal in gestructureerd standaard formaat**
 - duidelijke nummering van paragrafen
 - gebruik van vette druk, schuine druk,...
 - minder kans dat iets vergeten wordt bij indeling in paragrafen per soort req.
 - steeds een motivatie voor de specificatie of keuze geven! (“rationale”) belangrijk bij wijzigingen.
 - eventueel een paragraaf voorzien met door de gebruiker uit te voeren acties
 - » vb. gebruiker selecteert een item, gebruiker beweegt de cursor,...

Requirements specification

- meer detail bij de requirements definition: moet het systeem voldoende kunnen beschrijven om het design te maken
- vaak worden systeem modellen toegevoegd
- notaties:
 - natuurlijke taal, nadelen:
 - verschillende interpretatie door verschillende lezers
 - te flexibel, moeilijk uit te maken of twee formuleringen hetzelfde betekenen
 - groepering en structureren is moeilijk
 - gestructureerde natuurlijke taal, met standaard forms of templates
- ➡ • “design description languages”, gebaseerd op een programmeertaal
- ➡ • speciale talen voor formulering van specificaties, vb. RSL (weinig gebruikt)
- grafische notaties, vgl. systeem modellen
 - vb. SADT (voor specialisten, complexe notatie)
- ⇒ • formele mathematische specificaties (zie cursus Geavanceerde SE)

Requirements specification (vervolg)

- moeten “traceable” zijn, d.w.z. referenties bevatten naar gerelateerde specs
 - duidelijke nummering
 - refereren naar gerelateerde specs met een nummer
 - cross-reference matrix met relaties
 - soms door CASE tools ondersteund, vb. navigatie via links

Specificaties in gestructureerde natuurlijke taal

- beperkte terminologie
- bevatten soms controlestructuren zoals programmeertalen vb. herhaling
- gebruiken eventueel formulieren of templates met volgende secties: (vooral voor functionele specificaties)
 - naam en beschrijving van de functie
 - beschrijving van de invoer-waarden en hun bron
 - beschrijving van de uitvoer-waarden en hun bestemming
 - opsomming van andere functies of entiteiten die gebruikt worden
 - in een functionele benadering: pre- en post-condities voor de functies (wat is waar vooraf en nadien)
 - eventueel beschrijving van de neven-effecten (vb. op andere waarden dan de invoer)
- combinatie van verstaanbaarheid, uitdrukingskracht en uniformiteit
- ondersteuning door CASE-tools
- kan gecombineerd worden met formele mathematische specificaties

Specificaties in een PDL (Program Description Language)

- een taal afgeleid van programmeertalen als Ada of Pascal (leesbaar)
- om dubbelzinnigheid van specificaties te beperken
- kan op syntax en semantiek getest worden door tools
- geschikt in volgende situaties
 - opeenvolging of volgorde van acties is belangrijk voor een functie
 - specificatie van interfaces naar andere functies of sub-systemen vb. type van interface objecten
 - diensten geleverd door een sub-systeem
 - combineren met formulieren in gestructureerde natuurlijke taal
- nadelen
 - minder geschikt om informatie over het applicatie domein te noteren
 - lezer moet vertrouwd zijn met concepten uit programmeertalen
 - tendens om te vlug aan een design te denken, en de vrijheid van de designers al te vroeg te beperken
- zie Ada voorbeeld...

Niet-functionele vereisten (non-functional requirements)

- bepalen eigenschappen en beperkingen van het systeem
- soms horen niet-functionele vereisten bij het ontwikkelingsproces
 - te volgen kwaliteitsstandaard
 - te gebruiken CASE-tool of procesmodel
- verschillende types niet-functionele vereisten:
 - product requirements: af te leiden uit de eisen gesteld door de gebruiker
 - performance, reliability (failure rate), usability,...
 - organisational requirements: af te leiden uit vereisten betreffende de organisatie van de klant of het bedrijf dat de ontwikkeling doet
 - te gebruiken procesmodel, programmeertaal, design methodology...
 - external requirements:
 - inter-operabiliteit met andere software, ethische problemen, volgens de wet, privacy, veiligheid,...

Niet-functionele vereisten (vervolg)

- vaak conflicten met andere requirements
 - vb. snelheid <-> geheugengebruik
- problemen i.v.m. formuleren van niet-functionele vereisten
 - kwantitatief uitdrukken zodat ze verifieerbaar zijn
 - vb. niet "easy to use" maar "training time must not exceed two days"
 - toepassen voor snelheid, grootte, betrouwbaarheid, robuust...
 - moet afzonderlijk van functionele vereisten...
 - ...maar dan zie je soms moeilijk de relatie tussen de requirements
 - niet-functionele vereisten zijn eerder op systeem-niveau dan per functie
 - vb. hoe gemakkelijk het systeem moet aan te leren zijn

Requirements in het Unified Process (UP)

Requirements in het Unified Process

Requirements = de vereisten waaraan de software moet voldoen

UP stelt “FURPS+” categorisatie voor

- FURPS: Functionality, Usability, Reliability, Performance, Supportability
- +: Implementation, Interface, Operations, Packaging, Legal

Merk op: zowel functionele als niet-functionele vereisten

Artefacten voor requirements analyse

- Overzicht probleemstelling
- Overzicht gebruikers(groep(en))
- Doelstellingen Systeem
- Systeem Functies (functionele req.)
- Systeem Attributen (niet-functionele req.)
- Use Cases

Requirements in het Unified Process

Systeem functies

- Functionele Vereisten
- Functionaliteit door opsomming Systeem Functies = Function Specification List
- Detectie Systeem Functies
 - Het systeem moet <X> doen
 - Vul X in op alle mogelijke manieren

Systeem attributen

- Niet functionele requirements
 - Bijv.: Usability, Reliability, Performance, Supportability, Interface,...
- Hebben een duidelijke relatie met de functionele requirements

Use Cases

Use case

- Definitie: Een Use Case is een document dat de opeenvolging van acties beschrijft van een actor gebruik makend van een systeem om een proces te doorlopen
- “Stories of using a system”
- Een actor is een entiteit die zich buiten het systeem bevindt en participeert in het proces (geeft input en ontvangt output)
- Onderdeel van analyse-fase, meer bepaald van de requirements bepaling
- Vooral voor invulling functionele requirements (De 'F' van FURPS+)
- Zeggen niets over de interne werking van het systeem (“Black box use cases”)

Types van UC

- Brief: slechts een paragraaf lang, beschrijft enkel het essentiële scenario
- Casual: beschrijft het essentiële scenario uitgebreider en de mogelijke variaties
- Fully Dressed: alle stappen en variaties tot in detail uitgewerkt
- Real Use Case (zie design): beschrijft de use case in termen van concrete input en output technologie.

Types actoren

- Primary : gebruikt het systeem om haar/zijn doel mee te bereiken
- Supporting : verleent een dienst aan het systeem
- Offstage : is geen primary of supporting actor, maar is geïnteresseerd in het systeemgedrag

Use cases vinden

- Actor-gebaseerd
 - Identificeer alle betrokken actoren (Merk op: een UC heeft altijd een primary actor)
 - Voor elke actor, identificeer het proces waarin ze participeren
- Event-gebaseerd
 - Identificeer alle externe acties waarop het systeem moet reageren
 - Verbindt deze acties met actoren en use cases

Brief use case template

Use Case : Titel van de Use Case
(genoemd naar het doel)
Actoren : Betrokken Actoren
Beschrijving : Korte beschrijving

Brief use case voorbeeld

Use Case : Koop goederen
Actoren : Kassier(ster), klant
Beschrijving : Een klant arriveert aan de kassa (POST) met de goederen die zij/hij wil aankopen. De kassier(ster) geeft alle goederen in aan de kassa en ontvangt het te betalen bedrag. De klant verlaat de winkel met zijn goederen.

Casual use case template

Use Case : Titel van de Use Case
(genoemd naar het doel)
Actoren : Betrokken Actoren
Beschrijving : Beschrijving use case en variaties

Casual use case voorbeeld

Use Case : Koop goederen
Actoren : Kassier(ster), klant
Beschrijving :

“hoofdscenario”:
Een klant arriveert aan de kassa (POST) met de goederen die zij/hij wil aankopen. De kassier(ster) geeft alle goederen in aan de kassa, ...

“alternatieve scenarios”:
Indien de klant met kredietkaart betaalt, en ...

Indien een van de goederen niet door de kassa wordt herkend, ...

Use cases

Fully dressed use case template

Use Case : Titel van de Use Case
 (genoemd naar het doel)
Actoren : Betrokken Actoren
Precondities en Postcondities
Beschrijving : Beschrijving scenario en
 variaties in twee kolommen

Actor actie 1		Systeem reactie 1
Actor actie 2		Systeem reactie 2
...		Systeem reactie 3
		...

Fully dressed use case voorbeeld

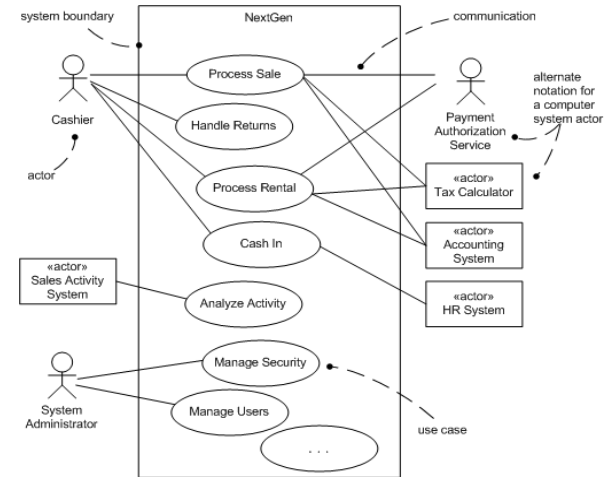
- Zie kopies
- Extra informatie, ook opgenomen in deze use case:
 - Doel
 - Overzicht (overgenomen uit brief UC)
 - Cross References (referenties naar Systeem functies)

Software Engineering 2008-2009

Slide 29

Use cases

Grafisch



Software Engineering 2008-2009

Slide 30

In de praktijk ...

Opstellen van use cases

- Richtlijnen:
 - Eerst de Brief Use Cases
 - Dan de Casual Use Cases hierop verder bouwen
 - Uiteindelijk de Fully Dressed Use Cases maken
 - Parallel met de drie vorige het diagram tekenen
 - Later zullen we de Real Use Cases behandelen
- Zijn bijna nooit perfect/volledig
- Bevatten geen technische informatie
- Op de eerste plaats geschreven documenten
- Meer informatie (voor de nieuwsgierigen)
 - <http://alistair.cockburn.us/usecases/usecases.html>
 - <http://alistair.cockburn.us/crystal/articles/ucai/usecasealternateintro.html>
 - <http://www.foruse.com/articles/eucobjects.htm>

Stelsel Sequentie Diagrammen

Software Engineering 2008-2009

Slide 31

Software Engineering 2008-2009

Slide 32

System Sequentie Diagrammen

System sequentie diagram (SSD)

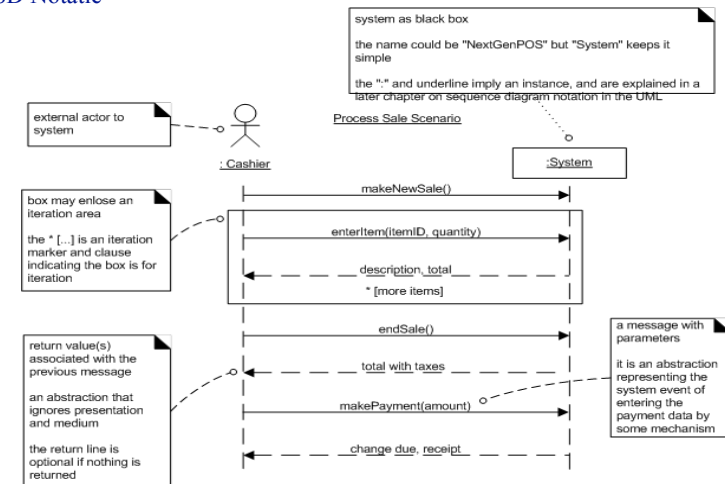
- Illustreert grafisch de interacties tussen actor en systeem dat gemaakt moet worden.
- Tijdens de interactie genereert een actor *systeem events meestal met systeem operaties als gevolg*.
- Wordt gemaakt voor belangrijke Use Cases
- Systeem wordt hier nog steeds als een zwarte doos beschouwd!
 - “system behavior” beschrijft **wat het systeem doet, niet hoe**.

SSD en de “system boundary”

- Duidelijke grens tussen het systeem en de actoren
- Grens afgeleid uit de use cases
- Hier: systeem event is een extern event
- Een menselijke actor kan geen systeem events ontvangen
 - Stippellijn duidt “feedback” of “resultaat” aan

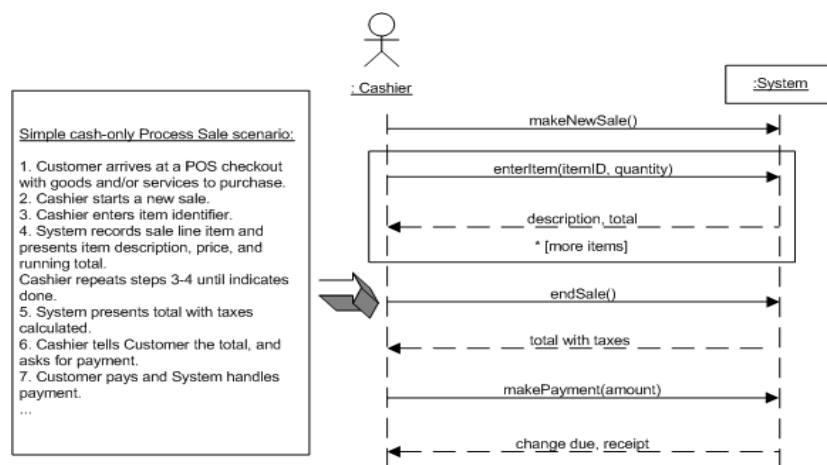
System Sequentie Diagrammen

SSD Notatie



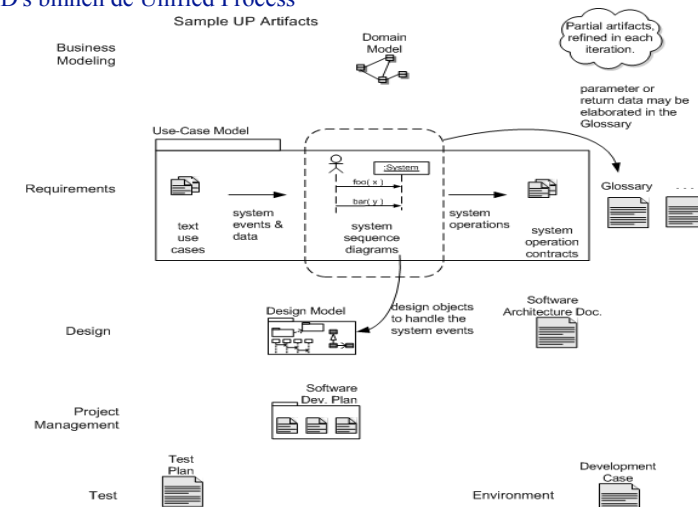
System Sequentie Diagrammen

SSD vanuit een use case



System Sequentie Diagrammen

SSD's binnen de Unified Process



Veel gemaakte fouten

- Geen afstemming artefacten
 - Bijv.: Vanuit de Use Cases moeten alle systeem functies gerefereerd worden
- Keuze actoren
 - Bijv.: Indien het systeem een AI component bevat is dat geen actor als het zich binnen de systeemgrenzen bevindt
- Granulariteit
 - Een Use Case is een redelijk grote begin-tot-eind beschrijving van een proces, typisch onderverdeeld in verschillende stappen en acties. Het is geen individuele stap of actie
 - Bijv.: “Een brood gaan kopen” versus “Het brood betalen”
- System events
 - Actors genereren “system events” die operaties van het systeem veroorzaken. Pijlen in een SSD wijzen dus steeds van de actor naar het systeem, *tenzij eventueel stippellijnen om aan te geven dat er een resultaat of feedback wordt terug gegeven!*
- Input/output technologie
 - Use cases en SSD's behoren tot de analyse-fase, we zijn dus slechts geïnteresseerd in **wat** het te ontwerpen systeem doet, niet **hoe**. Informatie over hoe een gebruiker input geeft, of hoe het systeem zijn output toont hoort hier dus niet thuis!
 - Bijv.: “De gebruiker kiest een item uit de listbox ...”

Opstarten van het project

- Groepjes vormen
 - Gebruik de hiervoor project-website.
- Brainstormen over de opgave
- Zoeken en noteren van systeem functies
- Zoeken en noteren van systeem attributen
- Maken van verschillende types use cases