

Requirements Engineering voor informatiesystemen

Een sociotechnische benadering



René Joosten

**Bachelorscriptie
Informatiekunde, NIII
Nijmegen 2006**

Voorwoord

De zoektocht naar een goed scriptieonderwerp was niet makkelijk. Ik wilde iets doen met de vakken die ik het meest interessant en relevant vond in mijn studie. Uiteindelijk kwam ik uit bij twee vakken; Requirements Engineering en Sociotechniek. Deze vakken vinden hun oorsprong in twee totaal verschillende studierichtingen, wat het juist leuk maakt om deze theorieën/methodieken te proberen te combineren. Zeker omdat ik denk dat de theorieën achter Sociotechniek, die zich vooral op het sociale vlak afspelen, Requirements Engineering kan helpen. Dit laatste is namelijk vooral een technische aangelegenheid.

Natuurlijk heb ik deze scriptie niet kunnen maken zonder enige hulp. Vooral wil ik mijn scriptiebegeleider, Mario van Vliet, bedanken voor zijn begeleiding bij de voortgang van mijn scriptie. Daarnaast wil ik graag Mies, Anke en Henri bedanken, twee vriendinnen en een vriend die mij geholpen hebben met het taalgebruik in deze scriptie. Verder wil ik mijn ouders en andere vrienden bedanken voor het met mij meedenken over de voortgang van mijn scriptie.

“Concern for man and his fate must always form the chief interest of all technical endeavours. Never forget this in the midst of your diagrams and equations.”

- Albert Einstein

Inhoudsopgave

VOORWOORD	II
INHOUDSOPGAVE	III
1 INLEIDING	1
2 DEFINITIES	1
3 INFORMATIESYSTEMEN	2
3.1 ASPECTEN	2
3.2 TOT SLOT	2
4. SOFTWARE ENGINEERING	3
4.1 FASES	3
4.2 ENKELE OPMERKINGEN	4
5. REQUIREMENTS ENGINEERING	4
5.1 SOORTEN REQUIREMENTS	4
5.2 REQUIREMENTS EISEN	4
5.3 STROMINGEN	5
5.4 FASES	5
5.5 TOT SLOT	6
6. SOCIOTECHNISCHE SYSTEMEN	6
6.1 TWEE DEELSYSTEMEN	6
6.2 DE VIER VARIABLEN	7
7 SOCIOTECHNISCHE PRINCIPES	7
7.1 PRINCIPE 1	7
7.2 PRINCIPE 2	8
7.3 PRINCIPE 3	8
7.4 PRINCIPE 4	8
7.5 PRINCIPE 5	8
7.6 PRINCIPE 6	8
7.7 PRINCIPE 7	9
8 INFORMATIESYSTEMEN EN SOCIOTECHNISCHE SYSTEMEN	9
8.1 SYSTEEMKENMERKEN	9
8.2 DE VERGELIJKING	10
8.3 CONCLUSIE	10
9 SOCIOTECHNISCHE PRINCIPES TOEGEPAST OP DE REQUIREMENTS ENGINEERING VAN INFORMATIESYSTEMEN	10
9.1 PRINCIPE 1	10
9.2 PRINCIPE 2	11
9.3 PRINCIPE 3	11
9.4 PRINCIPE 4	11
10 HET C2000-SYSTEEM	11
11 DE PRINCIPES TOEGEPAST OP REQUIREMENTS ENGINEERING	12
12 CONCLUSIES	13
LITERATUUR	15

1 Inleiding

Sinds de opkomst van de eerste gecomputeriseerde informatiesystemen eind jaren '70, worden er steeds hogere eisen (requirements) gesteld op welke wijze deze systemen zich mogen gedragen.

Men kan dan denken aan eisen met betrekking tot:

- de hoeveelheid fouten die aanwezig mogen zijn in het systeem;
- de aansluiting van het systeem bij de organisatie;
- de acceptatie van het systeem door de gebruikers.

In de praktijk heeft men doorgaans nog niet volledig voldaan aan deze en andere eisen. De meeste organisaties ondervinden hier nog dagelijks problemen mee. Deze problemen zijn vaak te herleiden naar de mensen die met de systemen moeten werken. (het sociale vlak) In dit onderzoek stel ik de vraag hoe de sociotechniek, die zich onder andere richt op de interactie tussen de techniek en de mensen die deze techniek gebruiken, kan bijdragen aan het ontwerp van beter geformuleerde requirements voor informatiesystemen. Op deze manier hoop ik een aantal punten onder de aandacht te brengen tijdens de requirements engineering fase, die tot het ontwerp van een betere requirements van een systeem kunnen leiden. Dit leidt tot de volgende onderzoeksvraag:

Hoe kan de sociotechnische theorie toegepast worden op de requirements engineeringfase voor informatiesystemen om tot een beter ontworpen systeem te komen?

In de onderzoeksvraag staat een term die verduidelijkt moet worden. *Beter ontworpen informatiesystemen*. Hiermee bedoelen we informatiesystemen die beter aansluiten bij de eisen van de organisatie die het systeem gaat gebruiken.

Dit onderzoek zal worden uitgevoerd aan de hand van een literatuurstudie van sociotechnische systemen, en dan met name sociotechnische ontwerpprincipes, informatiesystemen en requirements engineering. Zo worden een aantal principes

geselecteerd waaraan requirements engineering zich moet houden volgens de sociotechniek.

Achtereenvolgens zullen in deze scriptie een aantal onderwerpen behandeld worden. Eerst zullen een aantal definities van de verschillende onderwerpen aan bod komen. Daarna zullen de essentie van informatiesystemen, software engineering en requirements engineering uitgelegd worden. Dan volgt een uiteenzetting van de sociotechniek, welke vergeleken zal worden met een informatiesysteem. Als laatste kijken we hoe sociotechniek requirements engineering kan optimaliseren.

Deze zullen vervolgens getoetst worden aan een casestudy van de Algemene Rekenkamer over het C2000 systeem.

2 Definities

In dit onderzoek krijgen we te maken met een aantal theorieën en methodieken. Nu zijn er veel verschillende stromingen binnen de verschillende onderwerpen. Er wordt voor elk onderwerp een zo algemeen mogelijke vorm gekozen. Wanneer dit niet mogelijk is, wordt er voor één specifieke stroming gekozen. In de rest van deze scriptie zullen deze definities en stromingen gehanteerd worden:

- *Sociotechniek (ST)*. • De sociotechniek is een framework voor onderzoek naar een optimale samenwerking te creëren tussen werk en organisaties. Een sociotechnisch systeem bestaat uit een sociaal en technisch systeem[1]. Dit zijn meestal uiterst dynamische systemen. Vanuit dit oogpunt kunnen een aantal principes van sociotechnische systemen worden opgesteld. In de ST zijn drie grote stromingen te onderkennen, de Scandinavische, Nederlandse en Amerikaanse. Ik heb specifiek gekozen voor de Amerikaanse, omdat deze het beste aansluit bij mijn onderzoeksvraag. Deze stelt een aantal principes op die zo algemeen mogelijk zijn geformuleerd. Dit betekent dat ze ook toepasbaar zijn op andere gebieden.
- *Informatiesystemen (IS)*. Een informatiesysteem kan men op heel

veel verschillende manieren interpreteren. In deze scriptie heb ik daar één mogelijk interpretatie van gekozen die in dit onderzoek nog verder wordt uitgewerkt, namelijk: *Informatiesystemen realiseren de informatievoorziening van organisaties, individuen en apparaten door middel van generatie, opslag, interpretatie, transformatie, transport en presentatie van gegevens, in de verschijningsvormen tekst, beeld of geluid.*[2]

- *Software Engineering (SE)*. Voor SE hanteren wij een heel algemene definitie [3]: *Software engineering is the systematic approach to the development, operation, maintenance and retirement of software*. In de laatste twintig jaar zijn er heel veel verschillende software engineeringmethodieken opgekomen die voldoen aan deze definitie. In de meeste van deze methodieken kunnen een aantal dezelfde fases herkend worden. Deze zijn verderop in het onderzoek uitgewerkt, en dan met name Requirements Engineering.
- *Requirements Engineering (RE)*. Zave [4] biedt een heldere definitie van requirements engineering: *“The branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families.”* Voor requirements engineering, evenals bij SE, zijn ook veel verschillen methodieken te onderkennen. Ik heb er hier voor gekozen om het zo algemeen mogelijk te houden, om het onderzoek breed toepasbaar te houden.

3 Informatiesystemen

Alvorens we gaan kijken hoe een informatiesysteem ontwikkeld wordt, kijken we wat een informatiesysteem is. Er bestaan veel verschillende definities over. In het vorige hoofdstuk hebben we al gedefinieerd wat wij

onder een informatiesysteem verstaan. Opvallend is dat in deze definitie niet de woorden geautomatiseerd of gecomputeriseerd zitten. Dit is dan ook geen vereiste. We zullen echter wel in het vervolg aannemen dat het een geautomatiseerd systeem betreft. Dit omdat we requirements engineering in de context van software engineering zien.

3.1 Aspecten

Informatiesystemen staan in relatie tot een aantal aspecten om goed te kunnen functioneren, namelijk: organisaties, individuen en apparaten, welke allemaal met elkaar samenwerken. Deze aspecten, alsmede informatie, beslaan de vier aspecten waarin een informatiesysteem opereert en zullen hieronder verder uitgewerkt worden. [5]

Organisatorische aspecten

Een organisatie heeft een missie, visie en doelstellingen. Deze oefenen invloed uit op hoe informatiesystemen worden vormgegeven. Het geeft het systeem zijn “reden van bestaan” binnen de organisatie. Dit is binnen de informatiesystemen zichtbaar in een directe link naar de doelstellingen van een organisatie.

Menselijke aspecten

Deze richten zich op de individuele mens en haar relatie met het informatiesysteem. Denk bijvoorbeeld aan de menselijke capaciteiten, motivaties, drijfveren, wensen, manier van denken, communiceren, werken, leren, etc. Ze hebben allemaal invloed hebben op het systeem.

Informationele aspecten

Dit betreft alle informatie die uitgewisseld wordt door informatiesysteem. Dit kan in verschillende vormen gebeuren, in sensoren van een robotarm tot persoonsgegevens invoeren in een database.

Technologische aspecten

Dit is de technologie die het genereren, opslaan, interpreteren, transformeren, transporteren en presenteren van gegevens in verschillende verschijningsvormen ondersteunt met behulp van computers.

3.2 Tot slot

Deze vier aspecten van een informatiesysteem dienen al gedefinieerd te worden in het

Requirements Engineering deel, onderdeel van de ontwerpfase van het ontwikkeltraject, welke beschreven wordt in hoofdstuk 4. In de praktijk blijkt dat deze aspecten niet allemaal even goed vertegenwoordigd zijn. Dit betreft dan vooral de aspecten die moeilijk te herkennen en beschrijven zijn, namelijk de organisatorische en menselijke aspecten.

4. Software Engineering

Om goed te begrijpen wat requirements engineering inhoudt, kijken we allereerst naar software engineering in het geheel.

Er zijn zeer veel verschillende aanpakken om een informatiesysteem te ontwikkelen. Echter elk project bestaat min of meer uit dezelfde fases [6] (zie figuur 1). Van elke fase zullen de belangrijkste activiteiten beschreven worden. Ondanks dat er met fases wordt gewerkt, elk met een duidelijke einde, kunnen deze meerdere iteraties doormaken.

4.1 Fases

Tijdens het ontwerp wordt de basis gelegd voor het informatiesysteem. Hier vindt het leeuwendeel van het contact met de stakeholders plaats en wordt de informatie die hieruit volgt verder gemodelleerd. Met stakeholders worden personen of organisaties bedoeld die te maken krijgen met het te ontwikkelen systeem. Zij kunnen direct of indirect invloed op de requirements uitoefenen.

Definitie

Alvorens men een systeem gaat bouwen, moeten er keuzes worden gemaakt. Of het systeem de moeite waard is om te bouwen, wat het systeem moet bevatten en vooral ook wat het systeem niet moet bevatten.

Ontwerp

In deze fase vindt het meeste contact plaats met de personen en organisatie of organisaties waarvoor het systeem wordt gebouwd. Dit gebeurt om een goed beeld te krijgen van het te

ontwikkelen systeem. Het is erg belangrijk dat deze fase zo goed als mogelijk wordt afgerond. Als er onderdelen niet compleet of incorrect zijn, heeft dit grote consequenties op de rest van het project. Op dit tijdstip worden de meeste eisen vastgelegd en is de fase waarin Requirements Engineering plaatsvindt.

De modellen, welke niets meer zijn dan manieren om eisen te communiceren hoe het systeem eruit moet komen te zien, worden hier vorm gegeven. Deze modellen bevinden zich op verscheidene abstractieniveaus. Dit om een zo goed en gedetailleerd mogelijk zicht te krijgen van systeem.

Vanuit de modellen die opgeleverd zijn in de ontwerpfase, kan er worden begonnen aan de bouw van het systeem. Gedurende de creatie van het systeem moet het uitvoerig getest worden. Dit testen gebeurt doorgaans eerst door testspecialisten en programmeurs, maar later ook door de toekomstige eindgebruikers van het product. Als op dit moment nog fouten geconstateerd worden, kan er nog iets aan gedaan worden. Echter grote wijzigingen zijn heel kostbaar om door te voeren, omdat het systeem dan grote veranderingen moet ondergaan.

Bouw

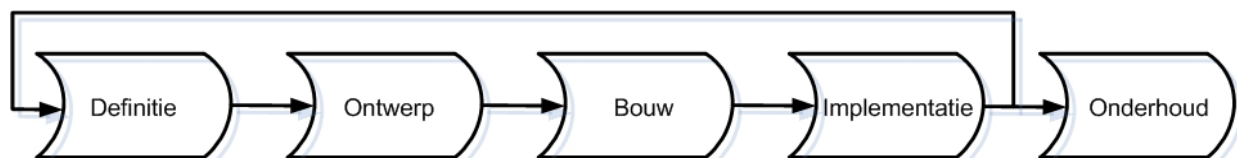
In de bouwfase gebeurt het meeste programmeerwerk. De programmeurs kunnen vanuit de modellen die de ontwerpfase heeft opgeleverd, het systeem gaan bouwen aan de hand van een vastgelegde structuur. Tijdens deze fase zal het systeem ook al uitgebreid getest worden.

Implementatie

In deze fase zal het ontworpen en gebouwde systeem ingepast worden in een bepaalde organisatie. Er zal onder andere moeten nagedacht worden over een invoeringsstrategie (in één keer of in stappen bijvoorbeeld).

Onderhoud

Als het systeem is opgeleverd, zullen er nog altijd enige onderhoudswerkzaamheden nodig zijn. Hierbij kan er gedacht worden aan fouten



Figuur 1

die nog optreden tijdens het gebruik van het systeem of kleine veranderingen in sommige van de functionaliteiten.

4.2 Enkele opmerkingen

Tijdens het gehele proces is er contact met de stakeholders. Maar dit is vooral het geval in het ontwerpfasen, en in het bijzonder bij Requirements Engineering. Het is daarom van groot belang vooral in deze fase een goed begrip van de wensen van de stakeholders te krijgen en hoe dit de omgeving gaat beïnvloeden. Met deze informatie zal de rest van het systeem worden gebouwd. Hoe verder we in het proces zitten, hoe moeilijker het zal worden om iets aan te passen. Ook zorgen deze aanpassingen in de bouw- en onderhoudsfases voor grote kostenposten. [7] Over het algemeen geldt, hoe vroeger de fouten ontdekt worden, hoe goedkoper het is.

5. Requirements Engineering

We hebben al kort bekeken wat requirements engineering betekent voor het ontwikkelproces van informatiesystemen. Dit zal verder toegelicht worden om tot een beter inzicht te komen hoe de RE in elkaar zit. We hebben al gezien dat het erg belangrijk is dat Requirements Engineering zo goed als mogelijk wordt afgerond. Als er onderdelen niet compleet of foutief zijn, heeft dit grote consequenties op de rest van het project.

5.1 Soorten requirements

De definitie van RE uit hoofdstuk 2 geeft gedeeltelijk aan wat een requirement is; een beschrijving van een aspect van het systeem dat een zekere taak kan vervullen of beperkingen oplegt. Requirements zijn aan de hand van deze definitie onder te verdelen in drie verschillende categorieën. [8]

Functional Requirements

Dit zijn de eisen die direct terug te vertalen zijn naar een functie of optie van het systeem. Men kan denken aan een eis zoals; "Het systeem zal elke nacht een overzicht uitdraaien van de inventaris".

Non-Functional Requirements

Deze zijn niet specifiek bedoeld voor een bepaalde functionaliteit van een systeem. Zij

plaatsen restricties op het te maken product en het software engineering proces zelf. Ook definiëren ze externe eisen waar het product aan moet voldoen. Voorbeelden hiervan zijn veiligheid, snelheid of betrouwbaarheid.

Domain constraints

Deze zijn bedoeld om inzicht te krijgen in een bepaald domein. Zaken zoals beperkingen van hardware, maar ook beperkingen in de omgeving van het systeem. Denk hier bijvoorbeeld aan de capaciteiten van de gebruikers.

Er is een groeiende aanhang onder software engineers om ook contextuele aspecten in de requirements op te nemen en te onderscheiden. Hierbij kan er gedacht worden aan menselijke en organisationele aspecten. [9].

5.2 Requirements eisen

De verzameling van requirements met hun onderlinge samenhang moet leesbaar zijn voor elke stakeholder. Hierbij kan gedacht worden aan eindgebruikers, systeemontwikkelaars, managers en klanten van organisaties. Deze verzameling wordt het *requirements document genoemd*.

Requirements zélf moeten aan een aantal eisen voldoen [10],

- Bij het onderhouden van het software systeem kunnen grote problemen ontstaan als niet alle informatie expliciet is gedocumenteerd. Dit zorgt voor situaties waar men onderdelen van het systeem niet meer kan *begrijpen*.
- Gespecificeerde requirements mogen *niet ambigu* zijn. Ze mogen maar op één manier kunnen worden opgevat. Omdat ze in natuurlijke taal worden genoteerd en deze van nature ambigu is, is dit geen makkelijke taak. Dit impliceert ook dat de gebruikte terminologie dezelfde moet zijn als die toegepast in het domein.
- Gespecificeerde requirements moeten *te controleren* zijn. Dit betekent dat tijdens het software engineering proces, er de mogelijkheid moet bestaan om na te gaan of de requirements worden nagekomen.

- Onderling moeten de requirements *consistent* zijn. Ze mogen niet conflicteren met elkaar.
- Voor elke requirement moet het duidelijk zijn, tijdens het gehele software engineering proces, hoe hij vertaald wordt naar het systeem. Een *goede ordening* maakt het mogelijk voor andere documenten om te refereren naar specifieke requirements.

Het requirements document zelf moet ook voldoen aan een aantal voorwaarden: [10]

- Requirements kunnen *veranderen*. Daarom moet het requirements document ook kunnen veranderen. Het moet zo georganiseerd zijn dat veranderingen in het document makkelijk zijn door te voeren.
- Een requirements document moet *compleet* zijn. Alle significante functional en nonfunctional requirements en de beschrijving van het domein moeten gedocumenteerd zijn. Als dit echter niet mogelijk is, omdat de specificatie alleen kan plaatsvinden op een later tijdstip, moet het document tenminste dit later tijdstip weergeven.
- Ook moeten het alle requirements apart én bij elkaar *SMART* zijn. Dit wil zeggen; Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden.

5.3 Stromingen

Het RE-proces bestaat uit twee aanpakken, elk met hun eigen voor- en nadelen. Deze zijn goal based en scenario-based Requirements Engineering. De meeste methoden bestaan uit

een mix van de twee. [11]

Goal based requirements engineering

Goals beschrijven de uiteindelijke doelen waaraan een systeem moet voldoen. Het is belangrijk dat algemene doelen vroeg in het RE-proces worden erkent. Het zoeken naar goals gaat echter het hele proces door, specifieke goals worden gedestilleerd uit bekende algemene goals. Het ontdekken van de goals door de requirements engineer spitst zich vooral toe op het kijken naar het probleem domein en de eisen van de stakeholders. Hij kijkt niet alvast naar mogelijke oplossingen voor deze problemen.

Scenario-based requirements engineering

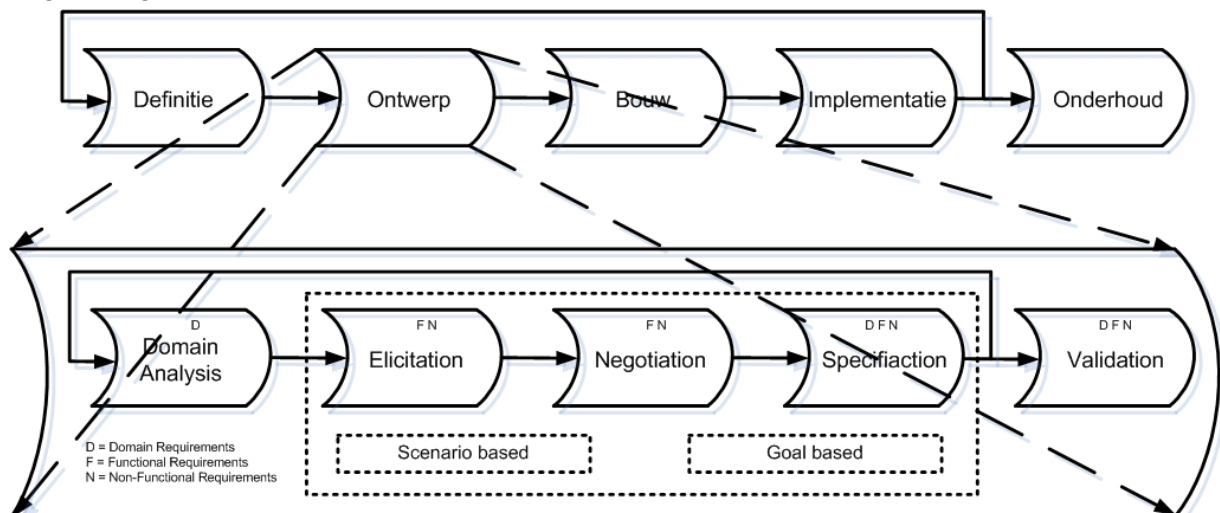
Het is vaak moeilijk voor gebruikers om zich precies voor te stellen wat zij willen in een nieuw systeem. Als hulpmiddel kunnen dan scenario's worden ingezet. Deze richten zich op de taken die een gebruiker moet uitvoeren op het systeem en kunnen de functionele requirements goed beschrijven. Deze scenario's worden ook vaak gebruikt om een aspect van het systeem beter te kunnen doorgronden.

5.4 Fases

Het RE-proces bestaat uit een aantal fases.[9] Dit proces kan, indien de requirements nog niet aan de voorwaarden voor requirements of het requirements document voldoen, meerdere malen geïtereerd worden. Achtereenvolgens worden deze fases hier kort beschreven. (zie ook figuur 2)

Domain analyse

De omgeving waarin het systeem moet worden gebouwd wordt in deze fase onderzocht. De



stakeholders betrokken bij het systeem worden geïdentificeerd, alsmede problemen in het domein. Voordelige mogelijkheden worden onderzocht en de doelen van het systeem worden globaal vastgesteld.

Elicitation

Mogelijke alternatieven voor het toekomstige systeem worden onderzocht. Hierbij worden de beoogde requirements van de stakeholders in acht genomen. Dit gaat meestal aan de hand van een aantal scenario's om het voor de stakeholders zo duidelijk mogelijk te maken. Doordat er meerdere alternatieven worden onderzocht, scherpt dit meestal de grenzen aan tussen het systeem en de omgeving.

Negotiation

Het komt vaak voor dat stakeholders conflicterende requirements hebben. Dit heeft tot gevolg dat er keuzes gemaakt moeten worden welke requirements het systeem gaan vormen. Met alle stakeholders worden de risico's en alle mogelijke alternatieven geëvalueerd, totdat er een uitkomst is waar alle partijen zich het meest in kunnen vinden.

Specification

Deze overeengekomen uitkomst dient nog formeel te worden opgeschreven. De specificaties die hier uitkomen, zullen moeten worden gecontroleerd aan de hand van de eisen opgesteld voor requirements en het requirements document.

Validation

Tijdens de ontwikkeling van het systeem zal steeds teruggegrepen worden naar de opgestelde requirements om te kunnen oordelen of de ontwikkeling nog steeds op de juiste koers zit.

5.5 Tot slot

In dit hoofdstuk is beschreven hoe men doorgaans omgaat met het opstellen van requirements. In welke vorm, wat voor soorten er zijn en welke fases hier meestal bij doorlopen worden. Het valt op dat requirements engineering vooral gericht is op wat het systeem moet kunnen doen en wat zijn beperkingen zijn (de techniek en wat voor een informatie erin is opgeslagen). Ook is bekeken welke eisen aan een requirement en een requirements document worden gesteld, om een

goede basis te hebben waarmee we de rest van het software engineering traject in kunnen gaan.

Er wordt helaas weinig aandacht besteedt aan het begrijpen van de relaties die het systeem zal hebben met zijn omgeving en hoe de omgeving daardoor zal veranderen. Dit is ook maar mondjesmaat terug te zien in de andere software engineering fases. Door dit niet mee te nemen in de requirements fase, kan er uiteindelijk een informatiesysteem worden opgeleverd dat wel een goed systeem is, maar niet hét goede systeem. Hierdoor komt het voor bij grote projecten dat men regelmatig niet tevreden is met de uitkomst van het software engineering proces. [12]

6. Sociotechnische Systemen

Alle complexe objecten kunnen worden gezien als een systeem. Een systeem is iets dat bestaat uit verschillende delen met een aantal relaties en een omgeving waar het ook relaties mee kan aangaan.

6.1 Twee deelsystemen

Een sociotechnisch systeem (STS) bestaat in feite uit twee verschillende soorten systemen, een sociaal en een technisch systeem. Deze twee zijn onlosmakelijk met elkaar verbonden en hebben elk een verschillende set van onderdelen en relaties. De theorie van de sociotechniek beschrijft dat deze twee moeten zoeken naar een optimale samenwerking. De verschillen tussen de twee systemen moeten herkend en gerespecteerd worden, anders zal het systeem nooit optimaal kunnen functioneren. [14]

Om dit doel te bereiken beschrijven we een aantal richtlijnen en principes, welke rekening houden met het feit dat verschillende gebruikers en groepen specifieke behoeften hebben zodat gebruikers grote veranderingen enthousiast en zonder druk kunnen ontvangen. Daarom is een goed begrip van het domein van de stakeholders, net zo belangrijk als de identificatie van de technische en organisatorische eisen bij het ontwikkelen van een nieuw systeem.

6.2 De vier variabelen

Bij het implementeren van een dergelijk systeem dient vooral rekening gehouden te worden met de vier variabelen waaruit een sociotechnisch systeem, dus het sociale én technische systeem, uit bestaat. Technologie, personen, organisatie en taken (*informatie* over de taken), zoals Mumford [14] ook beschrijft. Deze vier variabelen vinden we ook terug in onze beschrijving van een informatiesysteem.

“The introduction of a new technical system is likely to disturb each of these variables. A new level of technology will bring with it a new man-machine relationship incorporating both opportunities and constraints. Because tasks are influenced by technology, the task structure of functions or departments using the system will be altered. New tasks mean that new demands are made of people and this will affect job satisfaction positively or negatively depending on whether the new situation meets their values of what is desirable work. Finally, the technology, tasks and people variables interact with an internal environment which provides a structural context for the achievement of the organization’s objectives and this interaction may start the looping process again by making demands of technology”

7 Sociotechnische Principes

De sociotechniek wil een optimale samenwerking tussen twee meestal uiterst dynamische systemen creëren, het technische en het sociale. Vanuit dit oogpunt kunnen een aantal (ontwerp)principes van sociotechnische systemen worden opgesteld.

1. Een sociotechnisch systeem bestaat uit menselijke en technische componenten die relaties onderling aangaan binnen en buiten het systeem om een gemeenschappelijk punt te bereiken.
2. Acties en beslissingen die door stakeholders worden genomen, in het ontwikkelproces van een systeem, hebben een grote impact op de uitkomst.
3. De behoeftes van de organisaties, gebruikers en andere stakeholders moeten in het systeem gereflecteerd

worden, alsmede de politieke processen die tussen de stakeholders spelen.

4. Problemen moeten bij de bron opgelost worden.
5. De systemen en het ontwerp hiervan moeten eigenaarschap hebben bij de stakeholders.
6. Systemen hebben continu te maken met onzekerheid.
7. Systemen moeten onderling compatibel zijn.

De doelstellingen voor het opstellen van de principes zijn vierledig. [15]. Ten eerste vooral om vragen op te werpen over de beoogde opgestelde systeemeisen. Ook bieden ze de mogelijkheid om met verschillende perspectieven naar het domein te kijken. Daarnaast bieden ze een framework om de eisen te evalueren. En als laatste is er een voorspellende factor. Men kan bijvoorbeeld voorspellen dat als er onvoldoende rekening gehouden wordt met een bepaalde zaak, dit voor problemen kan gaan zorgen in het verdere verloop van de ontwikkeling van het systeem.

Deze principes zullen in de volgende pagina’s verder uitgewerkt worden alvorens tot een conclusie te komen hoe deze kunnen helpen bij het RE-proces.

7.1 Principe 1

Een sociotechnisch systeem bestaat uit menselijke en technische componenten die relaties onderling aangaan binnen en buiten het systeem om een gemeenschappelijk punt te bereiken.

Een sociotechnisch systeem staat voortdurend doelbewust in wisselwerking met zijn “transactionele” en “contextuele” omgeving. [16] De transactionele omgeving bestaat uit de specifieke stakeholders, welke met het systeem interacteren en ook doelstellingen daarbinnen hebben. De contextuele omgeving zijn alle veranderbare componenten buiten het systeem die wel relevant zijn, maar deze zijn er niet direct aan gerelateerd. De sociotechniek neemt aan dat door systemen op deze manier te modelleren, men tot een beter inzicht in het gehele systeem komt. Deze manier kan gezien worden als een sociaal systeem dat interacteert met een technische systeem. Er is ook een duidelijk onderscheid tussen de transactionele

en contextuele omgeving. Dit betekent dat de sociotechniek een duidelijke grens trekt tussen het systeem en zijn omgeving. [17]. De mate waarop deze met elkaar interacteren ligt aan de waarden van het systeem en van zijn omgeving, welke terug zijn te vinden in de missie, visie en doelstellingen van een organisatie/systeem.

Deze waarden zijn cruciaal voor sociotechnische systemen. Ze moeten zo geformuleerd zijn dat binnen het transactionele systeem het menselijke en technologische systeem altijd in balans is. [18]

7.2 *Principe 2*

Acties en beslissingen die door stakeholders worden genomen, in het ontwikkelproces van een systeem, hebben een grote impact op de uitkomst.

Het ontwerp van een systeem is een iteratief proces. Als het ontwerp op tafel ligt, is er de behoefte aan een beter ontwerp. Het ontwerpteam zal zelf moeten evalueren of er genoeg iteraties zijn doorlopen om een helder en duidelijk systeem te kunnen opleveren. [18]. Als de keuze gemaakt is om een systeem te implementeren, zijn de acties en beslissingen, met als doel een nieuw informatiesysteem te introduceren in de werkomgeving, van sterke invloed op hoe het systeem gaat interacteren met zijn omgeving. [19].

7.3 *Principe 3*

De behoeftes van de organisaties, gebruikers en andere stakeholders moeten in het systeem gereflecteerd worden, alsmede de politieke processen die tussen de stakeholders spelen.

Stakeholders werkzaam in een organisatie zijn onderhevig aan de in het systeem heersende waarden en normen, alsmede hun eigen ambities. Dit kan soms voor conflicten zorgen om het gezamenlijke doel te bereiken. Om het ontwerp van een systeem in goede banen te leiden, moet hiermee rekening worden gehouden. De machtsbalans tussen de verschillende stakeholders zal moeten worden geïdentificeerd en compromissen zullen moeten worden gesloten. [20]

7.4 *Principe 4*

Systemen hebben continu te maken met onzekerheid.

Onzekerheid heeft altijd een centrale rol gehad bij de sociotechniek sinds de “turbulent environment”-discussie is geïntroduceerd door Emery en Trist [21]. In een continu radicaal veranderend systeem, zal er rekening moeten gehouden worden met grote onzekerheid. Dit zorgt voor moeilijk te ontwerpen systemen, die met een “critical minimum specification” [18] moet worden ontworpen. Dit kan van twee kanten bekeken worden. Aan de ene kant zegt het dat niets meer ontworpen moet worden dan absoluut nodig is. Aan de andere kant moeten juist de essentiële dingen worden geïdentificeerd. Door een overmaat aan te gedetailleerde ontwerpbeslissingen te maken, zorgt men ervoor dat het systeem minder goed kan inspelen op de turbulente omgeving.

7.5 *Principe 5*

De systemen en het ontwerp hiervan moeten eigenaarschap hebben bij de stakeholders.

Eigenaarschap wordt onder andere bereikt door participatie van stakeholders al tijdens de ontwerpfase. Tegenwoordig is het gebruikelijk om verschillende personen met verschillende expertises het systeem te laten bouwen. Meestal zijn deze activiteiten niet goed geïntegreerd en gecoördineerd, wat leidt tot eigenaarschap van verschillende groepen van stakeholders van de verschillende fases. [18] De sociotechniek stelt dat degenen die verantwoordelijk zijn voor het systeem, ook betrokken moeten zijn met de bouw. Iedereen die te maken heeft met het systeem heeft er ook enige expertise over. Door te zorgen dat stakeholders deze expertise aanwenden door te participeren bij de bouw van het systeem, zal men eigenaarschap ervaren tijdens de ontwikkeling van het systeem, als ook tijdens de ingebruikname. Dit zorgt er ook voor dat het systeem minder snel wordt verworpen door de stakeholders.

7.6 *Principe 6*

Problemen moeten bij de bron opgelost worden.

Chern stelt dat variaties, onvoorziene gebeurtenissen, bij de bron moeten worden opgelost. Systemen met grote onzekerheid,

kunnen juist meer voorspelbaar worden door problemen vroeg te onderkennen en ze controleerbaar te maken. Dit kan betekenen dat ze door organisatie als vaste mechanismen kunnen worden gezien. Dan kan men steeds dezelfde oplossing voor een bepaald type probleem gaan gebruiken. Hierdoor wordt een rigide situatie gecreëerd die niet meer om kan gaan met een probleem buiten een kleine oplosruimte. [22]. Dit conflicteert met principe 4. Het doel van dit principe is dan ook om het oplossend vermogen in te zetten in de delen van het systeem waar grote onzekerheid bestaat. Dit geeft men de vrijheid om deze problemen op te lossen en zodoende een goed lopende organisatie te creëren.

7.7 *Principe 7*

Systemen moeten onderling compatibel zijn.

Als een nieuw systeem wordt ontworpen is in principe 5 gesteld dat dit een gezamenlijk proces is van alle stakeholders. De mensen die eigenaarschap hebben van het probleem, moeten dit ook hebben bij de oplossing. Dit zorgt ervoor dat alle systemen de karakteristieken bevatten van de organisatie. [18]. Als systemen worden ontworpen binnen een groter systeem, hebben ze allemaal dezelfde basis, zodat ze samen één geheel vormen.

8 Informatiesystemen en sociotechnische systemen

De definities en visies van systemen, en in het bijzonder informatie- en sociotechnische systemen, zijn bekend. In dit hoofdstuk zal een IS en een STS worden vergeleken.

8.1 *Systeemkenmerken*

Alvorens we ingaan of een IS en een STS vergelijkbaar zijn, gaan we eerst in op de exacte kenmerken van een open systeem. De Open University System Group [23] heeft de volgende definitie van een systeem. Zij zien het als een geheel dat vier eigenschappen heeft:

1. De delen zijn verbonden in een georganiseerde manier.

2. Componenten worden beïnvloed door in het systeem aanwezig te zijn en door hun omgeving.
3. Het systeem doet iets.
4. Het systeem moet een doel hebben.

Bij een open systeem, kunnen we bij het tweede aspect stellen dat de componenten binnen het systeem ook beïnvloed worden door hun omgeving.

Alvorens we gaan toetsen of een sociotechnisch en een informatiesysteem vergelijkbare eigenschappen hebben, toetsen we eerst of ze beiden voldoen aan de kenmerken van een open systeem.

De delen zijn verbonden in een georganiseerde manier.

Een STS en een IS volgen beiden een langdurig ontwerptraject. Tijdens dit traject wordt de structuur van het systeem opgezet en wordt bepaald hoe het gaat functioneren. Als het eenmaal in bedrijf is genomen, gaat dit aan de hand van vooraf afgesproken patronen. Met andere woorden, op een georganiseerde manier.

Componenten worden beïnvloed door in het systeem aanwezig te zijn en door hun omgeving. Dat componenten binnen een informatiesysteem worden beïnvloed door zaken in en buiten het systeem is af te leiden uit het tweede deel van de definitie van informatiesystemen, “*door middel van generatie, opslag, interpretatie, transformatie, transport en presentatie van gegevens, in de verschijningsvormen tekst, beeld of geluid.*” Hier staat dat er informatie binnenkomt, wat betekent dat het systeem van buitenaf beïnvloed wordt. Deze informatie wordt ook getransformeerd, en zodoende van binnen beïnvloed.

Dat een STS inwendig beïnvloed wordt, kan teruggevonden worden in principe 1. Hier staat dat de twee deelsystemen relaties met elkaar aangaan. Dat het ook van buitenaf beïnvloed wordt, staat in principe 4. Hierin beschrijft men dat er altijd onzekerheid binnen het systeem is door invloed van buitenaf.

Het systeem doet' iets'.

Dat het systeem iets doet, is ook terug te vinden in de definitie van informatiesystemen en het eerste sociotechnisch principe, hierboven al uiteengezet.

Het systeem heeft een doel.

In de definitie van informatiesystemen staat: “*Informatiesystemen realiseren de informatievoorziening van organisaties, individuen en apparaten*”. Dit geeft aan dat een informatiesysteem een doel heeft.

Bij sociotechnische systemen is dit uit het eerste sociotechnische principe af te leiden. *Een sociotechnisch systeem bestaat uit menselijke en technische componenten die relaties onderling aangaan binnen en buiten het systeem om een gemeenschappelijk punt te bereiken.*

8.2 De vergelijking

Volgens Proper [5] heeft een informatiesysteem te maken met een viertal aspecten. Technologie, mens, informatie en organisatie. Binnen de principes van de sociotechniek zien we ook deze vier aspecten terug.

Zo stelt ST-principe 1 dat een sociotechnisch systeem bestaat uit menselijke en technische componenten. Twee aspecten waar volgens Proper een informatiesysteem ook mee te maken heeft.

Verder moeten de behoeftes van de organisaties, gebruikers en andere stakeholders moeten in het systeem gereflecteerd worden, volgens sociotechnisch principe 3. Dit betekent dat het organisationele aspect (herkenbaar in IS) ook vertegenwoordigd is in de sociotechniek.

Aangezien systemen volgens principe 4 continu te maken hebben met onzekerheid, speelt informatie een grote rol bij het anticiperen van problemen. We zien dit ook terug bij informatiesystemen, waar informatie het belangrijkste aspect vormt.

De vier aspecten van een informatiesysteem zijn ook dezelfde variabelen waarmee een sociotechnisch systeem in aanmerking komt volgens Mumford [14].

Daarnaast bestaan sociotechnische systemen in feite uit twee deelsystemen, een sociaal en een technisch systeem. Deze vergelijking gaat ook op voor informatiesystemen. Het technisch systeem kan men hier zien als de techniek die

achter het informatiesysteem schuilt. Het sociale systeem zijn de gebruikers van het informatiesysteem.

8.3 Conclusie

Beide systemen komen in aanmerking met dezelfde zaken: technologie, mensen, informatie en organisatie. Ook bestaan ze beiden uit twee verschillende systemen, een sociaal en een technisch systeem. Daarom stellen we dat de principes die gelden voor de sociotechniek, ook toe te passen zijn in een aangepaste vorm voor requirements engineering.

9 Sociotechnische principes toegepast op de requirements engineering van informatiesystemen

In hoofdstuk zeven zijn de principes van de sociotechniek behandeld. Door deze toe te passen op het domein van de requirements engineering van informatiesystemen zijn een aantal principes naar voren gekomen die specifiek te gebruiken zijn bij de RE-fase:

1. Requirements opstellen is een proces van alle stakeholders.
2. Requirements opstellen gebeurt met een zo minimalistisch mogelijke kritische specificatie.
3. De eisen van de stakeholders beïnvloeden gaandeweg het project de uitkomst van zowel het technische, als het sociale systeem binnen de organisatie.
4. Bij het opstellen van requirements moet rekening gehouden worden met het type organisatie.

9.1 Principe 1

Requirements opstellen is een proces van alle stakeholders.

Sociotechnisch principe 3 stelt dat er een machtsbalans bestaat tussen de verschillende stakeholders. De personen die gebruik maken van het systeem worden niet alleen door hun eigen ambities gestuurd, maar ook door de

waarden en normen binnen het systeem. Door samen te werken vanaf het begin, worden deze conflicten in het begin al duidelijk en kan er aan gewerkt worden. ST-principe 5 stelt ook dat de stakeholders vanaf het begin af aan moeten samenwerken en communiceren om eigenaarschap te creëren. Zo kan alle aanwezige expertise van de stakeholders worden gebruikt bij het ontwerp van het systeem, wat de kans vergroot dat het systeem ook daadwerkelijk gebruikt wordt.

9.2 Principe 2

Requirements opstellen gebeurt met een zo minimalistisch mogelijke kritische specificatie.

Systemen hebben continu te maken met onzekerheid, stelt sociotechnisch principe 4. Om deze onzekerheid op te vangen, moeten systemen ontworpen worden een minimalistisch mogelijke kritische specificatie. Het is daarom ook nodig requirements zo minimalistisch mogelijk te specificeren om zo goed mogelijk te kunnen reageren op de omgeving van het systeem en onverwachte problemen op te lossen. ST-principe 6 stelt dit ook. Door het oplossend vermogen van deelsystemen te maximaliseren, en dus de beslissingsruimte te vergroten, zal het systeem zo vlekkeloos mogelijk opereren.

9.3 Principe 3

De eisen van de stakeholders beïnvloeden gaandeweg het project de uitkomst van zowel het technische, als het sociale systeem binnen de organisatie.

Een sociotechnisch systeem bestaat, zoals ook al beschreven is, uit twee deelsystemen. Deze deelsystemen staan voortdurend in wisselwerking met elkaar. Het ligt voor de hand dat de eisen van stakeholders niet alleen in het technische óf in het sociale domein zijn te vinden. Ook al zouden de eisen alleen van technische aard zijn, stelt principe 2 dat deze alsnog het sociale systeem beïnvloeden, en andersom. De eisen van de stakeholders zijn ervoor om het systeem een bepaald doel te laten bereiken. Volgens sociotechnisch principe 1 kan dit doel alleen bereikt worden door een samenwerking van het sociale en technische systeem. Alle beslissingen die gemaakt worden tijdens de ontwikkeling van het proces in het

technisch of sociale proces, zullen dan de uitkomst van beide beïnvloeden, volgens sociotechnisch principe 2. Hiermee dient rekening gehouden te worden tijdens het opstellen van de requirements.

9.4 Principe 4

Bij het opstellen van requirements moet rekening gehouden worden met het type organisatie.

Systemen moeten onderling compatibel zijn, zo stelt sociotechnisch principe 7. Dit betekent dat de eisen van het te ontwikkelen systeem ook rekening moeten houden met het type organisatie. Zo kunnen alle systemen in een organisatie, inclusief de organisatie zelf, één geheel vormen.

10 Het C2000-systeem

De opgestelde principes in hoofdstuk 9 worden getoetst door middel van een casestudy, een evaluatie van het C2000-systeem [24]. In december 2002 heeft de Tweede Kamer de Algemene Rekenkamers verzocht een onderzoek in te stellen naar het C2000-project en invoering van het project Geïntegreerd Meldkamerssysteem (GMS). Met deze projecten wilde het kabinet de communicatie tussen brandweer, ambulancediensten en politie versterken, zodat ze elkaar bij rampen beter kunnen bereiken en de hulpverlening aan burgers sneller en effectiever kunnen uitvoeren.

Concreet bestaat het C2000-systeem uit een landelijk dekkend radionetwerk, netwerkdiensten, radiobediensystemen en randapparatuur. GMS is een softwarepakket voor meldkamers. Het C2000-project is een innovatief project waarbij vele stakeholders betrokken zijn en ook de nodige dingen fout zijn gegaan. Door deze fouten is het project aanzienlijk vertraagd.

Requirements opstellen is een proces van alle stakeholders.

Dit wordt ook duidelijk bij C2000. Het belang van het creëren van draagvlak/eigenaarschap is meerdere malen onderkent, echter zijn er maar in drie van de 25 regio's afspraken over

gemaakt. Als in elke regio structureel aan dit probleem zou zijn gewerkt, zou de uitkomst van het GMS een stuk positiever zijn geweest. De grootte van het project maakte het moeilijk voor actieve participatie. Er zou echter via bepaalde communicatiekanalen, bijvoorbeeld online fora of nieuwsbrieven waar men feedback op zou kunnen geven, wel meegedacht had kunnen worden over het project. De gebruikers van het systeem hadden ook een aantal zorgpunten door hun achtergrond en expertise. Deze zijn onvoldoende in acht genomen en door sommige van deze punten is het project dan ook flink vertraagd. Door iedereen te betrekken in het proces, zou men gezamenlijk betere requirements hebben kunnen opstellen, als ook een systeem hebben kunnen bouwen waarmee het merendeel zich identificeert.

Requirements opstellen gebeurt met een zo minimalistisch mogelijke kritische specificatie.

Het C2000-systeem heeft tijdens zijn ontwikkeling ook met grote onzekerheden te kampen gehad. Dit heeft gezorgd voor vertragingen. Een voorbeeld hiervan zijn de problemen met de verwachtingen over de bandbreedte van het C2000-netwerk. Uit gesprekken met de regio's bleek dat deze verwachtingen te hoog gespannen waren ten aanzien van de mogelijkheden, bijvoorbeeld van datacommunicatie en de beschikbare bandbreedte. Hier hingen een aantal functionaliteiten van GMS vanaf. Echter waren sommige erg vertraagd door aannames in het begin van het project. Deze aannames, die achteraf overbodig bleken te zijn zo vroeg in het project, konden ook later gemaakt worden, toen er meer bekend was over de mogelijkheden. Dit had voor minder vertragingen kunnen zorgen. Het GMS op zichzelf is een afwijking van de originele opdracht om een landelijk dekkend radiosysteem te ontwikkelen. Door deze vele, niet relevante, functies is het gehele project, inclusief de functies die wel noodzakelijk geïmplementeerd moesten worden, vertraagd.

De eisen van de stakeholders beïnvloeden gaandeweg het project de uitkomst van zowel het technische, als het sociale systeem binnen de organisatie.

In de casestudy is dit terug te zien. De Algemene Rekenkamer constateert dat de

planning van het C2000-project, vanaf het kabinetsbesluit in 1996, verschillende malen is aangepast. Volgens binnenlandse zaken (BZK) zal de recentste aanpassing inhouden dat de bedrijfsvaardige oplevering van het technische systeem tenminste met drie maanden vertraagd zal worden. Dit heeft grote uitwerkingen op het sociale systeem. Wat kan worden teruggezien in het vertrouwen in het C2000-systeem. Door zoveel aanpassingen hebben de personen die straks het systeem moeten bedienen, er steeds minder vertrouwen in dat dit systeem ook in hun behoeftes voorziet en zullen ze zich er minder voor inzetten.

Bij het opstellen van requirements moet rekening gehouden worden met het type organisatie.

Het C2000 project is een innovatief project, waarbij vele stakeholders betrokken zijn en een zeer complexe projectorganisatie met zich meebrengt. Wat impliciet betekent dat de onderlinge delen één geheel zouden moeten vormen. Ondanks het feit dat men daar veel aandacht aan geschonken heeft, was het niet genoeg. Er was geen eenduidige aanpak over het gehele proces inzake de requirements opstellen. In alle 25 regio's hebben de regionale autoriteiten onvoldoende instructies gekregen om voor een uniforme aanpak te kunnen zorgen. Ook dit heeft voor vertragingen gezorgd.

11 De principes toegepast op requirements engineering

De principes van de sociotechniek zijn toepasbaar op het RE-proces van informatiesystemen. In deze principes is terug te zien dat requirements engineering niet alleen een proces is wat de requirements engineer betreft, maar effect heeft op het hele systeem. Iedereen moet erbij betrokken worden. Hierbij moet men wel oppassen om niet in een eindeloze spiraal te verzanden, waarbij men ieders wensen wil tegemoetkomen. Dit is terug te vinden in requirements-principe 1 en 2. Er is ook naar voren gekomen in requirements-principe 3, dat men veel aandacht moet besteden aan de requirements zelf. Deze moeten ontworpen worden met een minimalistisch mogelijke kritische specificatie. Requirements

principe 4 benadrukt nog eens de behoefte voor organisational constraints.

Terugblikkend naar hoofdstuk 5 zijn deze vier principes te vertalen naar aanvullingen op het daar uitgelegde requirements engineering proces.

1. *Requirements opstellen is een proces van alle stakeholders.*

Dit betekent dat in de domain analyse fase van requirements engineering het zeer belangrijk is om alle stakeholders goed te identificeren en deze ook gaandeweg het proces betrokken te houden. Een handig hulpmiddel is hierbij is om een lijst van alle stakeholders en hun expertises bij te houden met hun doelstellingen en na te gaan bij een belangrijke beslissing inzake de requirements engineering of deze geconsulteerd zijn. Men moet natuurlijk niet doorslaan in dit proces en willen proberen om iedereen altijd te raadplegen. Dit zal leiden tot situaties waar men geen gemeenschappelijke uitkomst kan vinden.

2. *Requirements opstellen gebeurt met een zo minimalistisch mogelijke kritische specificatie.*

In hoofdstuk 2 staat de eis dat het requirements document compleet moet zijn. Dit is uit te breiden tot de eis dat het ontworpen moet zijn met een zo minimalistisch mogelijke kritische specificatie. Alles wat beperkend is voor de verdere ontwikkeling van het systeem, moet pas gedefinieerd worden op het punt wanneer het ook echt noodzakelijk is. Natuurlijk moet men wel in het achterhoofd houden dat het requirements document SMART moet blijven.

3. *De eisen van de stakeholders beïnvloeden gaandeweg het project de uitkomst van zowel het technische, als het sociale systeem binnen de organisatie.*

Er moet nog meer aandacht komen op de wijze hoe het requirements document omgaat met veranderingen in de loop van het SE-proces. Het bouwen van een informatiesysteem is vooral nog met name een technische aangelegenheid. Er moet meer rekening worden gehouden met hoe men reageert op bepaalde veranderingen in het technische systeem. Tevens moet men kijken wat voor effect het

heeft op de stakeholders wanneer de requirements op een later tijdstip worden aangepast. Hiervoor is beperkt de ruimte. Men kan dit vergelijken met het bouwen van een gebouw, wat het gehele SE-proces voorstelt en requirements engineering de fundering. Als de fundering moet worden aangepast terwijl het gebouw al bijna af is, bestaat de kans dat het instort. Ook hier kan een kanttekening geplaatst worden. Men moet namelijk niet te bang worden om niets meer te veranderen en zodoende met een systeem opgezadeld te worden wat niet aan de wensen voldoet.

4. *Bij het opstellen van requirements moet rekening gehouden worden met het type organisatie.*

Tijdens de domein analysefase moet ook veel nadruk gelegd worden op het type organisatie, wat ook gereflecteerd moet worden in de organisational constraints, zodat het informatiesysteem er goed op aansluit. Kenmerkende eigenschappen van de organisatie moeten geïdentificeerd worden en met deze eigenschappen moet rekening gehouden worden.

12 Conclusies

Requirements engineering is een belangrijke stap in het ontwikkelproces van een goed informatiesysteem. We moeten daarom alles aangrijpen om dit proces zo optimaal te laten verlopen. In dit onderzoek hebben we de Amerikaanse sociotechniek doorgelicht om te kijken of de ontwerpprincipes ook geschikt zijn voor RE. Vanuit deze principe hebben we vier principes kunnen opstellen voor RE, namelijk:

1. Requirements opstellen is een proces van alle stakeholders.
2. Requirements opstellen gebeurt met een zo minimalistisch mogelijke kritische specificatie.
3. De eisen van de stakeholders beïnvloeden gaandeweg het project de uitkomst van zowel het technische, als het sociale systeem binnen de organisatie.
4. Bij het opstellen van requirements moet rekening gehouden worden met het type organisatie.

De casestudy, beschreven in deze paper, zou beter zijn verlopen als men deze principes beter in acht had genomen.

Natuurlijk zullen deze principes niet voor de ontwikkeling van elk informatiesysteem even zwaar wegen. Ze moeten worden gezien als een leidraad, waar men naar inzicht van af kan wijken.

In een vervolgonderzoek zou men zelf een aantal casestudies kunnen afnemen, om zodoende beter te controleren of deze principes ook herhaaldelijk toepasbaar zijn voor RE en of er vanuit de praktijk geen andere zaken naar voren komen tijdens het toepassen van de principes.

Literatuur

- [1] Dulce Pumareja & Klaas Sikkel, *An evolutionary approach to groupware implementation: the context of requirements engineering in the socio-technical frame*, 2002
- [2] Verkenningcommissie Informatica, *Geen toekomst zonder Informatica –Toekomstverkenning Informatica*. 1996
- [3] *IEEE Standard Glossary of Software Engineering Terminology*, 1983
- [4] Pamela Zave & Michael Jackson, *Four Dark Corners of Requirements Engineering*, 1996
- [5] Erik Proper, *Informatiekunde - Exacte Vaagheid*. 2003
- [6] Ian Sommerville, *Software Engineering 6th Edition*, 2001
- [7] Hans van Vliet, *Software Engineering, principles and practice*, 1993
- [8] B. Nuseibeh & S. Easterbrook, *Requirements Engineering: a roadmap*, 1998
- [9] Gerald Kotonya & Ian Sommerville, *Requirements Engineering, processes and techniques*, 1997
- [10] A. van Lamsweerde, *Requirements Engineering in the Year 00: A Research Perspective*, 2000
- [11] Yijun Yu, Julio Cesar Sampaio do Prado Leite & John Mylopoulos, *From Goals to Aspects: Discovering Aspects from Requirements Goal Models*, 2003
- [12] Robert Giegerich & John Hughes, *Functional programming in the real world*, 1994
- [13] Anders Ekholm, *Modelling of user activities in building design*, 2001
- [14] Enid Mumford, *Sociotechnical Systems Design; Evolving theory and practice*, 1987
- [15] C. W. Clegg, *Sociotechnical principles for system design*, 2000
- [16] Nicholas Trist, *The evolution of socio-technical systems: a conceptual framework and an action research program*, 1981
- [17] Ludwig von Bertalanffy, *Théorie générale des systèmes*, 1980
- [18] A. Cherns, *The principles of Sociotechnical design*, 1976
- [19] Tora K. Bikson & JD Eveland, *Groupware implementation: Reinvention in the sociotechnical frame* 1996
- [20] E. L. Trist, *On Socio-Technical Systems* 1951
- [21] F.E. Emery and E.L. Trist, *The Causal Texture of Organizational Environments*, 1965
- [22] J. Wall, *Evaluation in a sociotechnical context*, 1999
- [23] Open University Systems Group, *Systems behaviour*, 1981
- [24] Algemene Rekenkamer, *Communicatienetwerk C2000 en Geïntegreerd Meldkamersysteem*, 2003