

# **Kravhantering – nyckeln till ett lyckat IT-projekt?**

Camilla Byman

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informationsteknik
Identifikationsnummer:	2963
Författare:	Camilla Byman
Arbetets namn:	Kravhantering – nyckeln till ett lyckat IT-projekt?
Handledare (Arcada):	Hanne Karlsson
Uppdragsgivare:	Nyländska Jaktklubben r.f.
<p>Sammandrag:</p> <p>Detta examensarbete behandlar kravhantering inom IT-projekt och belyser ämnet med hjälp av en fallstudie utförd inom segelföreningen Nyländska Jaktklubben r.f. Kravhantering inom programvaruutveckling består av tre huvudprocesser: kravinsamling, analys och specifikation. Kravinsamlingen sker först och med hjälp av olika tekniker samlas material om kundens önskemål och behov in. Analysen utnyttjar kravinsamlingens material och bearbetar behoven till krav. Specifikationen är den sista delen och använder analysens resultat för att skapa definitionerna på hur kraven borde förverkligas i lösningssystemet. Examensarbetet lägger mera vikt på de två första eftersom specifikationen inte var en del av fallstudien. Olika kravinsamlingstekniker presenteras i detalj och jämförs. Tre olika analysmetoder, strukturerad analys, objektorienterad analys och problemdomänorienterad analys, diskuteras och jämförs.</p> <p>Fallstudien bestod av ett registerprojekt vars mål var att förnya medlems- och båtregistret och registerprojektet utvecklades till att innefatta även fakturering och resurshantering. Kravhanteringen inom fallstudien jämförs i examensarbetet med teorin. Kravinsamlingen, analysen och valet av leverantör behandlas.</p> <p>Problemformuleringarna i examensarbetet tangerar rubriken och tar ställning till hur göra kunden nöjd, med vilka metoder det lönar sig att utföra kravhanteringen, varför den är en kritisk del inom IT-projekt och hur fallstudien löstes. Dessa frågor uppmärksammas genom hela arbetet och besvaras i helhet i avslutningen.</p>	
Nyckelord:	Programvaruutveckling, kravtyper, kravinsamling, kravspecifikation, IT-branschen, informationssystem
Sidantal:	80
Språk:	Svenska
Datum för godkännande:	6.5.2010

DEGREE THESIS	
Arcada	
Degree Programme:	Information technology
Identification number:	2963
Author:	Camilla Byman
Title:	Requirements engineering – the key to a successful IT project?
Supervisor (Arcada):	Hanne Karlsson
Commissioned by:	Nyländska Jaktklubben r.f.
<p>Abstract:</p> <p>This thesis deals with requirement engineering in IT projects and illustrates the subject by using a case study conducted within the sailing club Nyländska Jaktklubben. Requirement engineering in software development consists of three main processes: requirements elicitation, analysis and specification. Requirement elicitation is the first and by using various techniques material on the customers wishes and needs is collected. Analysis requirements based on elicitation findings and process needs are formulated. The specification is the last part and uses the analysis results to create definitions of how the requirements should be realized in the solution system. The thesis puts more emphasis on the first two since the specification was not part of the case study. Different requirements elicitation techniques are presented in detail and compared. Three different methods of analysis are presented. Structured analysis, object oriented analysis and problem-oriented domain analysis are discussed and compared.</p> <p>The case study consisted of a project whose goal was to renew a register for members and boats. The project evolved to include also billing and resource management. Requirements engineering within the case study is compared in the thesis with the theory. Requirements elicitation, analysis and the selection of the supplier are discussed and processed.</p> <p>The problems discussed in the thesis are how to make the customer happy, by what methods it is worthwhile to perform requirements engineering, why it is a critical part of IT projects, and how the case study was solved. These issues are addressed through the whole thesis but answered in full in the end.</p>	
Keywords:	Software development, requirement types, requirement elicitation, requirement specification, IT industry, informationsystems
Number of pages:	80
Language:	Swedish
Date of acceptance:	6.5.2010

# INNEHÅLL

<b>1</b>	<b>Inledning</b>	<b>7</b>
1.1	Är kravhantering viktig?	7
1.1.1	<i>Misslyckade IT-projekt</i>	8
1.2	Problemformulering	9
1.3	Introduktion till fallstudien	10
1.4	Definitioner	10
<b>2</b>	<b>Kravhantering</b>	<b>12</b>
2.1	Allmänt	12
2.1.1	<i>Vad slutresultatet är och vad det används till</i>	12
2.1.2	<i>Kravställarna</i>	13
2.2	Kravindelning	14
2.2.1	<i>Användarkrav</i>	14
2.2.2	<i>Systemkrav</i>	16
2.2.3	<i>Funktionella krav</i>	18
2.2.4	<i>Icke-funktionella krav</i>	18
2.2.5	<i>Domänkrav</i>	21
2.3	Kravinsamling	21
2.3.1	<i>Vilken information samlas in</i>	21
2.3.2	<i>Varifrån informationen samlas in</i>	22
2.4	Kravinsamlingstekniker	22
2.4.1	<i>Val av kravinsamlingsteknik</i>	23
2.4.2	<i>Bakgrundsforskning</i>	24
2.4.3	<i>Intervjuer</i>	24
2.4.4	<i>Enkäter</i>	29
2.4.5	<i>Dokumentgranskning</i>	30
2.4.6	<i>Observation</i>	30
2.4.7	<i>Användningsfall</i>	32
2.4.8	<i>Brainstorming</i>	33
2.4.9	<i>Kravåteranvändning</i>	34
2.5	Analys	34
2.5.1	<i>Strukturerad analys</i>	35
2.5.2	<i>Objektorienterad analys</i>	35
2.5.3	<i>Problemdomänorienterad analys</i>	36
2.5.4	<i>Kravdokumentet</i>	40
2.6	Specifikation	41

2.6.1	<i>Extern design</i>	42
2.6.2	<i>Specifikationsdokumentet</i>	44
2.7	Kravvalidering	45
<b>3</b>	<b>Fallstudien</b>	<b>47</b>
3.1	Bakgrund	47
3.1.1	<i>Existerande system</i>	48
3.1.2	<i>En segelförenings varierande behov</i>	48
3.1.3	<i>Avnämare och målgrupper</i>	49
3.2	Kravinsamlingen	50
3.2.1	<i>Bakgrundsforskning</i>	50
3.2.2	<i>Intervjuer</i>	51
3.2.3	<i>Dokumentgranskning</i>	53
3.2.4	<i>Observation</i>	53
3.2.5	<i>Brainstorming</i>	54
3.2.6	<i>Problemdomänen</i>	54
3.3	Analys	55
3.3.1	<i>Domänkrav</i>	55
3.3.2	<i>Medlemsvy</i>	56
3.3.3	<i>Båtvvy</i>	58
3.3.4	<i>Resursvy</i>	58
3.3.5	<i>Funktioner och tjänster</i>	60
3.4	Specifikationen	60
3.5	Offertrundan	60
3.6	Sammanfattning	62
<b>4</b>	<b>Slutsatser</b>	<b>63</b>
	<b>Källor</b>	<b>67</b>
	<b>Bilagor</b>	<b>69</b>
	<b>Bilaga 1. Intervjudokumentet för fallstudien</b>	
	<b>Bilaga 2. Användarkraven för fallstudien</b>	
	<b>Bilaga 3. Processbeskrivningen för fallstudien</b>	

## Figurer

Figur 1. Kravhanteringens huvuddelar. ....	12
Figur 2. Kravdokumentanvändare enligt Sommerville (2004 s 137). ....	13
Figur 3. Kombinationsmöjligheter för kravtyper modifierad och baserat på Wohlin (2005 s 100). ....	14
Figur 4. Notationssätt enligt Sommerville (2004 s 131). ....	17
Figur 5. Olika typer av icke-funktionella krav enligt Sommerville (2004 s. 122). ....	19
Figur 6. Icke-funktionella kravs måttenheter. ....	20
Figur 7. Kravinsamlingstekniker baserat på Bray (2002 s 50). ....	24
Figur 8. Exempel på aktörer i ett användningsfall. ....	32
Figur 9. Problemramen för ett enkelt informationssystem (Bray 2002 s 103). ....	38
Figur 10. Problemramen för ett mera invecklat informationssystem (Bray 2002 s 105). .....	39
Figur 11. Riktlinjer för kravdokumentet (Bray 2002 s 106). ....	40
Figur 12. Förhållandet analys - specifikation – design. ....	42
Figur 13. Resultatsammanställning för fråga 8. ....	52
Figur 14. Kandidatjämförelse vid offertrundan. ....	61

## Förkortningar

BF	Finlands Båtförbund r.f.
FSF	Finlands Seglarförbund r.f.
NJK	Nyländska Jaktklubben r.f.
OOA	Objektorienterad Analys
PDOA	Problemdomänorienterad Analys
SA	Strukturerad Analys
SQL	Structured Query Language

# 1 INLEDNING

Detta examensarbete behandlar kravhanteringsprocessen inom IT-projekt, varför kravhanteringen är en viktig och kritisk del av ett IT-projekt samt med vilka metoder och tekniker kravhanteringen kan utföras.

En fallstudie vid segelföreningen Nyländska Jaktklubben r.f. (hädanefter NJK) utfördes i form av kravhanteringen inom ett registerprojekt. Fallstudien utfördes utan djupare detaljkunskaper om ämnet och i detta examensarbete analyseras fallstudien teoretiskt och jämförs med existerande metoder.

## 1.1 Är kravhantering viktig?

Det finns personer som inte lägger något större värde på kravhanteringen och kanske tycker att det är kundernas sak att säga vad de vill ha. Det finns nog projekt som inte kräver någon större investering i kravhanteringen, t.ex. då

- problemet är litet och enkelt
- problemet är allmänt känt eller tidigare dokumenterat
- eller om det inte spelar någon roll ifall det uppstår något fel.

För sådana projekt är det sällan kravspecifikationen som har misslyckats ifall projektet inte lyckas. Då projekten blir större eller problemen mera invecklade lönar det sig dock att lägga tid på att få kraven rätt från början. Det sparar mycket tid och problem då det från början utvecklas rätt saker och man inte halvvägs i projektet konstaterar att det lagts ner tid på fel saker eller i värsta fall, då lösningen eller produkten är klar konstateras detsamma. Oftast är det dessutom så att ju större projekt, desto större kostnader medför en dåligt gjord kravspecifikation.

Man kan se kravhanteringen som grunden till ett hus. Ifall grunden är gjord med felaktiga instruktioner eller dåliga material kommer inte huset heller att stå stadigt och rakt eller kan i värsta fall rasa ihop. Problem som orsakats av dålig eller bristfällig kravhantering är dessutom inte något som kan åsidosättas som något fenomen som inte idag

mera är aktuellt, eftersom en stor procent av alla IT-projekt misslyckas i dagens läge. T.ex. då Standish Group (1995) undersökte 8000 misslyckade mjukvaruprojekt, stod felaktiga krav för 56 procent av orsakerna.

Även Sommerville (2004 s 121) konstaterar att otydlighet i kraven är orsaken till många mjukvaruutvecklingsproblem. Det är naturligt för en systemutvecklare att tolka ett mångtydigt krav till det enklaste alternativet som tyvärr oftast inte är vad kunden hade önskat sig. Då kunden är missnöjd måste nya krav skrivas och följaktligen ändringar göras till systemet enligt de nya kraven. Detta medför extra kostnader och fördröjningar till projektet, eftersom designen för systemet i värsta fall måste göras om eller ändringar i programkoden utföras.

### **1.1.1 Misslyckade IT-projekt**

Ifall man idag slår upp misslyckade IT-projekt på webben hittar man alltför många träffar, som bevisar att detta problem inte på något sätt är inaktuellt. Några exempel på misslyckade IT-projekt enligt Bray (2002 s 6-7):

- Performing Rights Society, PROMS projektet. Projektet övergavs år 1992 efter att ha använt 11 miljoner pund. En bristande kravspecifikation var en av de övervägande orsakerna till att projektet misslyckades på grund av att kraven inte var i ett sådant format att de kunde kontrolleras av vanliga människor och att de var missledande.
- London Stock Exchange TAURUS projektet lades ner år 1993 efter ha använt 75 miljoner pund (dock uppskattades det att totala kostnaderna för det misslyckade projektet blev närmare 480 miljoner pund). Många problem kunde spåras tillbaka till oenigheter gällande motstridiga krav.
- London Ambulance Service utskickningssystem. Systemet stängdes av efter två dagar i bruk med usel kravanalys inom den sociala domänen som orsak.
- Swanick Air Traffick Control. Detta projekt hade tidsgräns 1998 men hade inte ännu år 2001 färdigställts (med en ytterligare kostnad av 180 miljoner pund). Fortskridande med realisering då det inte fanns en robust kravspecifikation var



den största orsaken till att projektet misslyckades enligt en officiell undersökningsnämnd.

Dessa är exempel på stora projekt där kravspecifikationen har varit felgjord eller bristfällig. Många andra har kommit till samma slutsats såsom Robert Glass konstaterar (1998 s 21) då han skriver att det råder föga tvivel om att kravhanteringen är den enskilt största orsaken till problem på programvaruprojektfronten. Studie efter studie har kommit fram till att där det finns misslyckande, är oftast kravhanteringsproblem kärnan i frågan.

Dessa är endast några exempel som visar på att en korrekt utförd kravhantering är nödvändig och kritisk för större IT-projekt. Denna process kan även ge mervärde till mellanstora och mindre projekt. Om det är frågan om riktigt små projekt, eller sådana som har en entydig funktion utgör kravspecifikationen dock inte längre den största riskfaktorn.

## **1.2 Problemformulering**

Målet med examensarbetet är att besvara några utvalda problemformuleringar. Den första problemformuleringen är starkt knuten till examensarbetets rubrik och svarar även på frågan varför examensarbetet skrevs:

- Varför är kravhanteringen en kritisk del för ett lyckat IT-projekt?

Den andra problemformuleringen tangerar IT-projektens stora problem och kvalitet, hur göra kunden nöjd men begränsar sig till kraven inom projekten:

- Hur försäkra sig om att kunden får vad den vill ha?

Den tredje problemformuleringen fokuserar mera kring hur kravhanteringen inom IT-projekt skall genomföras:

- Vilka metoder lämpar det sig att utnyttja inom kravhanteringen?

Den fjärde och sista problemformuleringen handlar om fallstudien. Eftersom fallstudien gjordes innan jag hade en djupare kunskap om kravhantering skall även följande fråga analyseras och besvaras:

- Hur löstes fallstudien jämfört med teorin och vad kunde ha gjorts bättre?

### **1.3 Introduktion till fallstudien**

Som ett praktiskt exempel och fallstudie i detta examensarbete används kravhanteringen som utfördes för NJK:s registerprojekt. Termen registerprojekt definieras i examensarbetet som ett projekt med målet att förnya medlems- och båtregistret inom föreningen. Projektet utvecklades till att innehålla även tunga komponenter såsom fakturering och resurshantering.

NJK är Finlands största segelförening och fallstudien utfördes under perioden november 2009 till mars 2010. I fallstudien ingick kravinsamlingen, analys av materialet och produktion av nödvändiga kravdokument. Fallstudien avslutades med en offertrunda och val av leverantör. Från början hade registerprojektet som mål att finna en färdig produkt som med möjliga modifikationer skulle kunna täcka NJK:s behov.

### **1.4 Definitioner**

#### **Användningsfall**

Användningsfall är en teknik använd inom kravinsamling. Ett användningsfall baserar sig på en eller flera tänkta funktioner inom det planerade systemet och kan ha aktörer av olika slag. Användningsfall går inte in på tekniska specifikationer.

#### **Avnämare**

Person med intresse att bevaka och mottagare av vara, synonym med *intressent*, på engelska används termen *stakeholder*.

#### **Fallstudie**

En fallstudie fokuserar på en undersökningsenhet för att få mer detaljerade kunskaper och syftar till att ge djupgående kunskaper om den undersökningsenheten och hur pro-

cesser utförs. Fallstudie utförs i undersökningsenhetens omgivning och kan genomföras med många olika metoder(Eliasson 2002).

### **Intern design**

Nedbrytning av ett system i dess olika delsystem för att förverkliga dess konstruktion, skapandet av interna strukturer och mekanismer för det slutliga systemet (Bray 2002 s 396).

### **Kund**

Med termen kund syftas i examensarbetet till beställaren, d.v.s. alla avnämare som en grupp.

### **Naturligt språk**

Naturligt språk syftar på användningen av vanlig text och inte teknisk jargong (Wohlin 2005 s 102).

### **Registerprojekt**

Med denna term syftas på fallstudien inom NJK som var ett projekt med målet att förnya medlems- och båtregistret inom föreningen. Registerprojektet utvecklades till att innehålla även tunga komponenter såsom fakturering och resurshantering.

## 2 KRAVHANTERING

### 2.1 Allmänt

Kravhanteringen är det första tekniska skedet i ett IT-projekt, där målet är att omvandla kundens behov och önskemål till krav och specifikationer som sedan används som indata till resten av projektet.

Själva kravhanteringsprocessen är i viss mån projektspecifik då den är beroende på hurdant och hur stort projekt det är fråga om. Som grundläggande moment i kravhanteringen kan sägas att kravinsamling, analys och specifikation sker, dock i olika mängder och på olika sätt beroende på metoden. Figur 1 ger en översikt över processen.



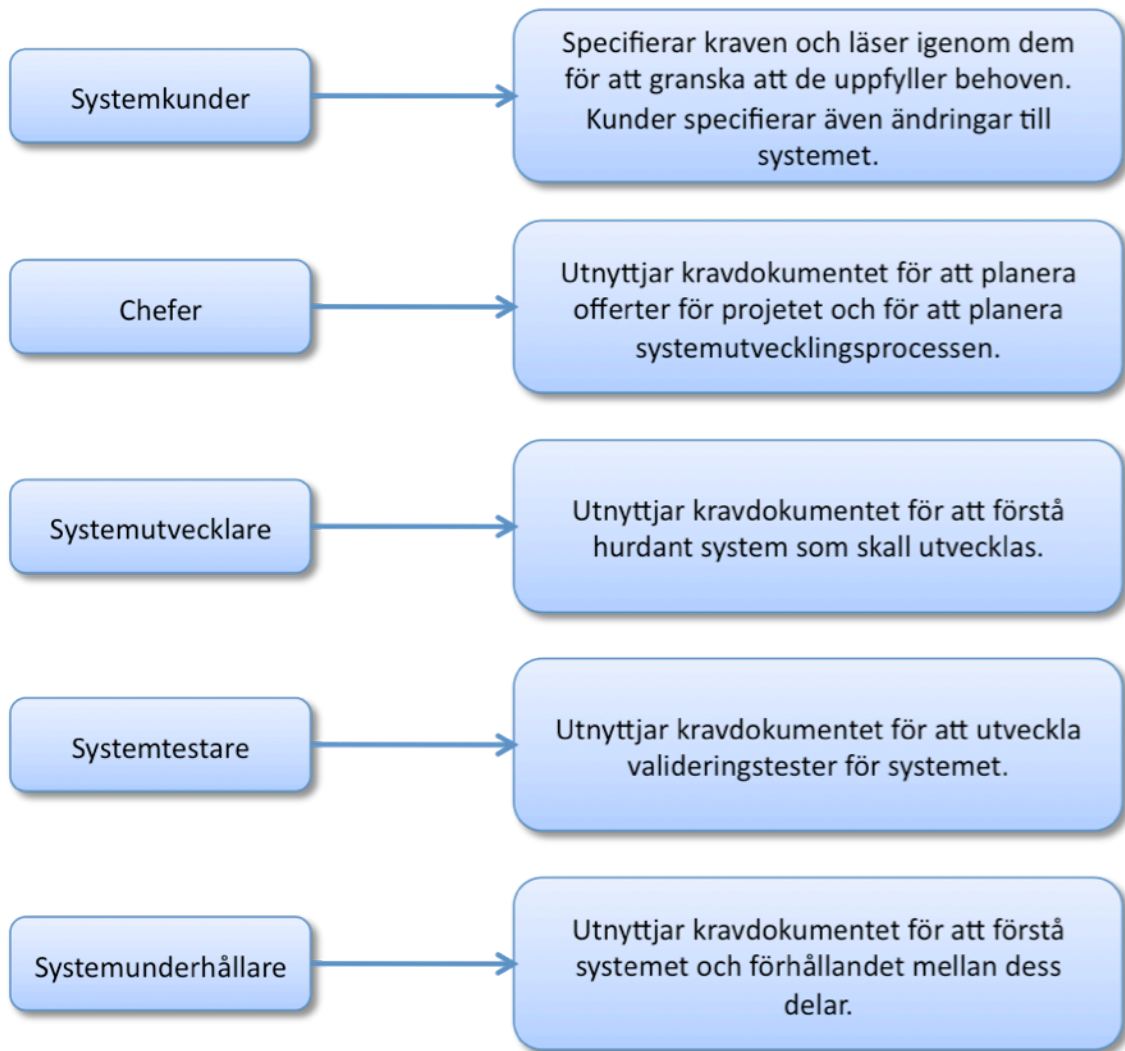
Figur 1. Kravhanterings huvuddelar.

Det finns många olika tillvägagångssätt då kravhanteringen skall utföras och dessa presenteras i korthet senare i kapitlet. Valet mellan dem baseras bl.a. på projektets typ och storlek, arbetsplatsens standarder, kundens engagemang och projektgruppens kunskaper och erfarenheter.

#### 2.1.1 Vad slutresultatet är och vad det används till

Slutresultatet av kravhanteringsprocessen är själva kravspecifikationsdokumentet som används som bas och källa till resten av programvaruutvecklingen i projektet. Såsom Wohlin (2005 s 95) uttrycker det: den aktivitet i programvaruutveckling som bestämmer vad programvaruprojektet skall göra. Nästa skede, planeringen av systemet, bestämmer sedan hur kraven skall realiseras. Det är dock många som har nytta av själva kravspecifikationsdokumentet, se Figur 2. Sommerville (2004 s 137) framhäver också att eftersom kravspecifikationsdokumentet har så många olika användare, måste det vara en kompromiss mellan förmedling av kraven till kunden så att denne förstår dem och specificering av även de minsta detaljer så att utvecklarna kan dra nytta av dokumentet.

Dokumentet bör även innehålla information om eventuella vidare utvecklingar av ett system.



Figur 2. Kravdokumentanvändare enligt Sommerville (2004 s 137).

### 2.1.2 Kravställarna

Enligt Lindegren (2003 s 88-89) skall kunden ställa kraven och leverantören skall analysera och utveckla kraven. Det vill säga att källan till kraven kommer från kunden och leverantören använder denna information till att utveckla användarkraven. När leverantören dessutom själv utvecklar och analyserar kraven lär denne sig mycket om det system som skall utvecklas. Som bas till detta påstående använder Lindegren (2003 s 29) termen *domänkunskap*. Kunden förstår i största delen av fallen alltid bättre den omgivning eller de problem som det blivande systemet skall stöda. Med samma motivering

säger Lindegren att man inte kan kräva av utvecklarna att de är experter på någon annan domän än deras egen, d.v.s. programvaruutveckling.

## 2.2 Kravindelning

Mjukvarukrav kan indelas i användarkrav och systemkrav. Dessa kan bestå av funktionella, icke-funktionella och domänkrav. Det går enligt Wohlin (2005 s 98) att dela in krav på många sätt, beroende på kravställare, olika kravnivåer, olika kravtyper och om kraven är riktade till själva programvaran, programvarumiljön eller om de gäller själva programvaruomgivningen.

Krav delas in i kategorier eftersom olika kravtyper har olika målgrupper. Användarkraven är till för avnämarna d.v.s. användarna eller kunden och systemkraven är till för utvecklarna. Dessa krav skrivs även på olika sätt och med olika mängder detaljer beroende på målgruppen. Ett exempel enligt Wohlin visas i Figur 3 på olika typer av krav och deras kombination.

	Funktionella krav	Icke-funktionella krav	Domänkrav
Användarkrav			
Systemkrav			

Figur 3. Kombinationsmöjligheter för kravtyper modifierad och baserat på Wohlin (2005 s 100).

Detta åskådliggör hur kraven inte nödvändigtvis alltid hör till en kategori t.ex. kan ett användarkrav vara både ett funktionellt och icke-funktionellt krav.

### 2.2.1 Användarkrav

Ett systems användarkrav skall beskriva de funktionella och icke-funktionella kraven så att de kan förstås av systemets användare utan detaljerad teknisk kunskap. Användarkraven skall endast specificera systemets beteende och undvika i mån av möjlighet att begränsa den inre systemdesignen, se Sommerville (2004 s 127).

Sommerville konstaterar även, att då användarkraven skall kunna förstås av tekniskt okunniga personer, måste kraven skrivas på naturligt språk. Detta orsakar många risker:

- Brist på tydlighet. Det är ibland svårt att använda naturligt språk för att specificera kraven exakt och entydigt och samtidigt göra dokumentet läsbart.
- Sammanblandning av kraven. Funktionella krav, icke-funktionella krav, systemmål och designinformation blir inte alltid tydligt separata.
- Kravsammanslagning. Många olika krav kan uttryckas tillsammans som ett enda krav.

Sommerville rekommenderar därför att separera användarkraven från de mera detaljerade systemkraven i kravdokumentet, eftersom en sammanslagning av kraven kan ge den icke-tekniska läsaren en sämre förståelse av kraven då texten blir för tungläst.

Användarkrav som är för detaljerade eller innehåller för mycket information begränsar utvecklarens kreativitet och blir oftast svåra att förstå. Ett användarkrav borde helt enkelt koncentrera sig på att klargöra den viktigaste funktionaliteten och huvudpoängen bakom kravet. Det är även en bra idé att skriva ut varför ett visst krav behövs, eftersom det minskar risken för mångtydighet och missförstånd för utvecklaren. Sommerville rekommenderar därför dessa riktlinjer då användarkrav skrivs:

- Utveckla ett standardformat och se till att alla kravdefinitioner följer detta format. Nyttan med detta är att sannolikheten för att allt relevant kommer med ökar och kraven är lättare att granska.
- Använd ett enhetligt språk, gör alltid tydlig skillnad mellan obligatoriska och önskvärda krav. Nyckelorden är ”skall” för obligatoriska krav och ”bör” för önskvärda krav. (Lindegren 2003 s 103)
- Använd textformatering för att understryka vikten av nyckelpunkter i kraven, t.ex. olika färger eller fet text.
- Undvik tekniska termer om möjligt.

## 2.2.2 Systemkrav

Sommerville (2004 s 129) definierar systemkrav, som utvidgade versioner av användarkraven och som används som utgångspunkt för systemplaneringen av programvaruutvecklarna. De är mera detaljrika och förklarar hur användarkraven skall tillämpas i systemet. Eftersom de kan användas som en del i kontraktet mellan kunden och leverantören, bör de vara fullständiga, konsekventa och täcka hela systemet.

Enligt Lindegren (2003 s 112) fungerar systemkraven som indata till den interna designen och själva programmeringen. Sommerville poängterar igen att systemkraven skall beskriva den externa designen och i mån av möjlighet undvika att låsa fast den interna designen. Systemkraven skall inte ta ställning till hur systemet internt planeras eller tillämpas. I praktiken är det dock nära på omöjligt att göra systemkraven tillräckligt detaljerade utan att i någon mån definiera aspekter i den interna designen. Sommerville räknar upp några orsaker till detta:

- Man kan vara tvungen att utforma en första version av systemets arkitektur för att hjälpa till att strukturera kravspecifikationen. Kraven kan då vara organiserade enligt de olika delsystem som planerats. Detta kan även vara aktuellt då återanvändning av moduler skall användas.
- Oftast är systemet tvunget att kommunicera med andra gränssnitt och detta lägger begränsningar på designen och således även på kraven.
- Den kan vara nödvändigt att använda en specifik arkitektur för att möta icke-funktionella krav.

Då både användar- och systemkrav skrivs är det vanligt att naturligt språk används. Då det gäller systemkrav kan detta emellertid medföra problem, eftersom systemkraven måste beskrivas i detalj. Sommerville definierar:

- Mångtydigheten i naturligt språk orsakar problem då läsare och författare inte har samma tolkningar på ord och meningar. Detta leder till missförstånd då författaren försöker säga en sak och läsaren tolkar det på något annat sätt.



- En kravspecifikation helt och hållet på naturligt språk blir för flexibel på ett negativt sätt då samma sak kan sägas på många olika sätt. Läsaren har då svårt att tolka om de olika kraven betyder samma sak, eller om de är separata.
- Det kan vara svårt att gruppera krav skrivna på naturligt språk samt problematiskt att se vilka krav som hör ihop.

På grund av dessa orsaker kan det vara svårt att skriva ett kravspecifikationsdokument enbart på naturligt språk utan risk för missförstånd och mångtydighet. Ifall dessa fel upptäcks först i ett sent skede av projektet finns det en stor risk att projektet blir väldigt dyrt.

Trots att det är viktigt att skriva användarkraven på ett språk, som icke-specialister inom programvaruutveckling förstår, kan systemkraven skrivas med ett mer tekniskt och specialiserat språk. Sommerville beskriver de olika notationssätten enligt följande: strukturerat naturligt språk, beskrivningsspråk för design, grafiska notationer och matematiska specifikationer. Se Figur 4.

<b>Notation</b>	<b>Förklaring</b>
Strukturerat naturligt språk	Detta sätt är beroende av att definiera standardformulär eller mallar för att uttrycka kravspecifikationen
Beskrivningsspråk för design	Detta sätt använder sig av ett språk som liknar programmeringsspråk, men med mera abstrakta funktioner för att specificera kraven genom att definiera en operativ modell av systemet
Grafiska notationer	Ett grafiskt språk med tillägg av text kommentarer används för att definiera de funktionella kraven för ett system. Vanliga metoder är användningsfall och sekvensdiagram
Matematiska notationssätt	Detta notationssätt är baserat på olika matematiska koncept, och är ett entydigt sätt att definiera kraven, men de flesta kunder förstår inte formella specifikationer och godtar inte detta.

Figur 4. Notationssätt enligt Sommerville (2004 s 131).

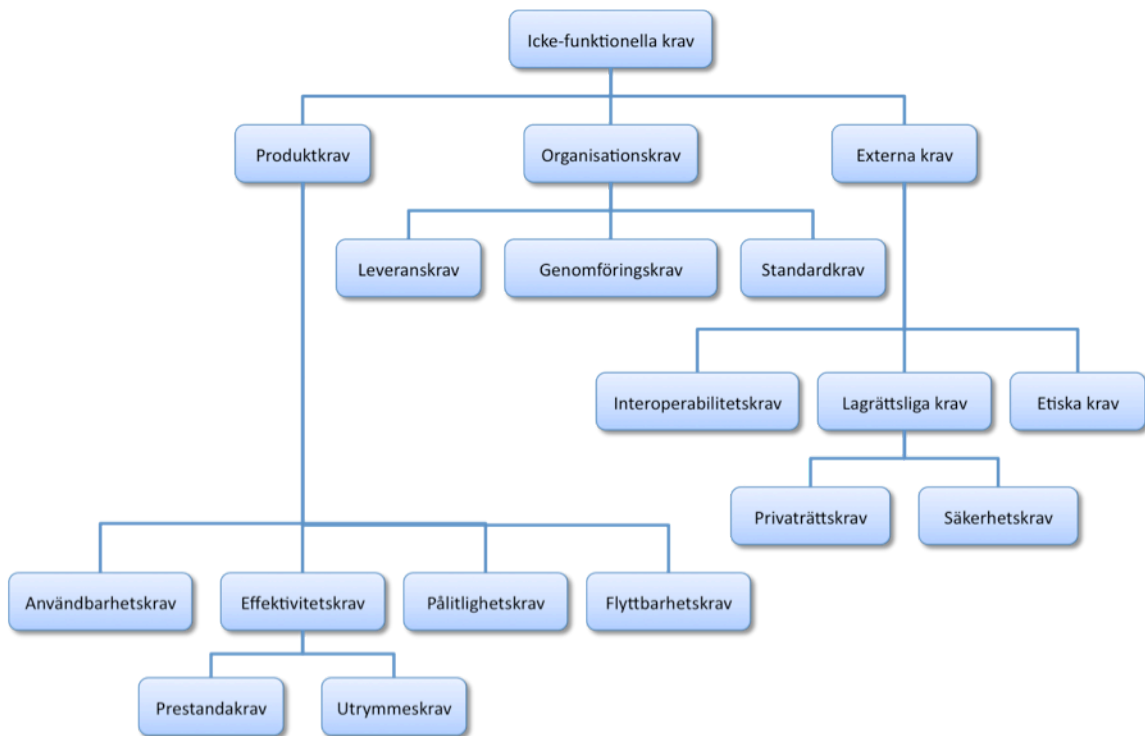
### **2.2.3 Funktionella krav**

Funktionella krav innebär kort sagt *vad* systemet skall göra. Sommerville (2004 s. 120-121) definierar funktionella krav som tjänster systemet skall erbjuda, hur systemet skall reagera på specifik indata och hur systemet skall bete sig under specifika omständigheter. I vissa fall kan även de funktionella kraven specificera vad systemet inte skall göra. Funktionella krav kan skrivas på olika sätt, men de borde alltid vara fullständiga och konsekventa. Med fullständiga menas, att alla tjänster kunden önskar skall vara definierade och med konsekventa menas, att kraven inte skall vara motstridiga. I praktiken är det dock närapå omöjligt att uppnå båda dessa krav, eftersom det för det första är lätt hänt att någon detalj inte tagits i beaktande, samt att olika avnämare oftast har olika motiv och behov och därav olika krav, vilka lätt blir motstridiga.

### **2.2.4 Icke-funktionella krav**

Icke-funktionella krav är begränsningar för tjänsterna eller funktionerna som systemet erbjuder. Till dessa räknas t.ex. tidtabellsbegränsningar och begränsningar berörande utvecklingsprocessen eller standarder. Icke-funktionella krav gäller oftast för hela systemet och inte en enskild tjänst eller funktion, se Sommerville (2004 s 121).

Icke-funktionella krav, såsom benämningen antyder till, har inget med själva systemets funktioner att göra, utan relaterar mera till bl.a. pålitlighet, svarstider och utrymmeskrav. Detta medför också att de kan vara mer kritiska än funktionella krav. Figur 5 beskriver olika nivåer och typer av icke-funktionella krav. Figuren visar även hur icke-funktionella krav kan komma från olika källor såsom produktkrav, organisationskrav och externa krav. Dessa är även de tre olika typer av icke-funktionella krav som Sommerville specificerar.



Figur 5. Olika typer av icke-funktionella krav enligt Sommerville (2004 s. 122).

Produktkraven specificerar produktbeteendet, såsom t.ex. hur snabbt systemet skall startas upp, hur mycket minne det får utnyttja och dess användbarhetskrav.

Organisationskraven härleds från strategier, politik och procedurer av både kunden och leverantörens företag. Exempel på sådana krav är olika standarder, genomföringskrav såsom programmeringsspråk eller design metoder och leveranskrav såsom när produkten skall levereras.

Externa krav täcker alla de krav som härleds från yttre faktorer till systemet och utvecklingsprocessen. Dessa krav kan vara interoperabilitetskrav, som definierar hur systemet skall interagera med andra gränssnitt, samt lagrättsliga krav, som bestämmer hur systemet skall följa lagen och etiska krav.

Ett allmänt problem med icke-funktionella krav är att de kan vara svåra att verifiera då en del av dem inte har en definitiv mätskala. Ofta kan de ses mera som mål än krav, t.ex. att systemet skall vara lätt att använda, inte få innehålla fel och bör vara snabbt att använda. Dessa är svåra krav då de inte går att mäta och utvecklarna kan tolka dem på ett mycket annorlunda sätt än kunden. Sommerville rekommenderar därför, att icke-funktionella krav skall skrivas kvantitativt, så att de går att testa med något på förhand

bestämt värde. Det går då att mäta dessa krav och på så sätt besluta ifall kraven har blivit uppfyllda.

<b>Egenskap</b>	<b>Måttenhet</b>
Hastighet	Bearbetade processer/sekund Användare/händelsesvarstid Skärmuppdateringstid
Storlek	Kbyte Antal RAM-chip
Användbarhet	Inlärningstid Antal hjälprutor
Pålitlighet	Tid mellan fel Sannolikheten för otillgängligt i systemet Antal fel Tillgänglighet
Stabilitet	Tid för omstart efter fel Procent för antal felorsakande händelser Sannolikhet för datakorruption vid fel
Portabilitet	Procent för målspecifika utsagor Antal gränssnittsystem

Figur 6. Icke-funktionella kravs måttenheter.

Sommerville ger exempel på hur olika krav kan mätas i Figur 6 enligt olika måttenheter. Han framhäver dock i samma mening att det kan vara svårt för kunder att översätta sina mål till dessa krav då de kanske inte har tillräcklig domänkunskap för att t.ex. säga hur stor procent för felorsakande händelser är tillräckligt liten för ett stabilt system. Även kostnaden för att testa och verifiera dessa krav kan bli stor och kunden ser inte alltid det som en godtagbar utgift.

Icke-funktionella krav kan ibland även orsaka konflikter med andra icke-funktionella och funktionella krav. Då är det bra att kunna kompromissa mellan olika krav och fatta beslut enligt kompromissen.

### **2.2.5 Domänkrav**

Domänkrav härleds snarare från applikationsdomänen i systemet än av systemanvändaren. De uttrycks oftast med specialiserad domänterminologi eller med referenser till domänkoncept. Domänkraven är viktiga eftersom de oftast relaterar till grundläggande systemaspekter och ifall dessa krav inte uppfylls kan det hända att hela systemet inte fungerar tillfredsställande. Oftast är domänkrav skrivna i problemdomänens språk och termer och kan vara svåra att tolka rätt av systemutvecklarna. (Sommerville 2004 s 126)

## **2.3 Kravinsamling**

På engelska används termen *requirement elicitation* och verbet *elicit* översätts till svenska enligt Ordboken (1994) till att locka fram, få fram eller framkalla. Kravinsamling är ett mera logiskt ord att använda för denna delprocess, men tyvärr ger ordet insamling lätt intrycket av att krav är någonting som finns färdigt att "plockas" eller samlas in. Detta är inte fallet, då bakgrunden till kraven nog finns hos avnämarna, men kraven kräver behandling och analys innan de kan skrivas ner och utnyttjas.

Kravinsamlingen är det första skedet i kravhanteringsprocessen och den kan genomföras med många olika tekniker. Den vanligaste och mest dominanta tekniken är intervjuer. Det är dock rekommenderat och vanligt att intervjuerna stöds av någon annan teknik för att komplettera och bekräfta intervjuresultaten. Andra tekniker är t.ex. bakgrundsforskning, enkäter, dokumentgranskning, observation, användningsfall, brainstorming och kravextraktion. I kapitel 2.4 presenteras de vanligaste teknikerna mer ingående.

Resultatet av kravinsamlingen, kravinsamlingsanteckningarna, används som grund vid analysen, som är följande skede i kravhanteringsprocessen.

### **2.3.1 Vilken information samlas in**

Vilken information som skall samlas in är mycket långt beroende på hurudant projekt och system som skall utvecklas. I stora drag kan det dock sägas, att det behövs en beskrivning av problemdomänen, en lista på problem som kräver lösning och de avgränsningar på funktionaliteten eller strukturen av systemet som kunden vill ha.

Oftast börjar man med lite information. Då kravinsamlingen framskrider ökar informationsmängden och samtidigt fås en noggrannare bild av vilken information som ytterligare behövs för att kravspecifikationen skall kunna skrivas. (Bray 2002 s 42)

### **2.3.2 Varifrån informationen samlas in**

Huvudsakliga källor för informationen om problemdomänen är:

- kunder, både nuvarande och potentiella
- kunders egna specifikationer
- ett tidigare existerande system och dess dokumentation, förutsatt att det hanterar liknande uppgifter som det tänkta systemet skall klara av
- användare av tidigare existerande system
- potentiella nya användare av det tänkta systemet
- tidigare liknande system som utvecklaren har skapat
- konkurrenters produkter
- experter för tidigare system
- dokumentering av huvuddragen och funktionaliteten hos gränssnitt
- relevanta tekniska standarder och lagstiftningar

Alla dessa källor behöver inte utnyttjas inom ett och samma projekt och vissa utesluter även varandra, t.ex. ifall det inte finns ett tidigare system kommer det inte heller att finnas dokumentation eller användare av ett tidigare system. (Bray 2002 s 42-43)

## **2.4 Kravinsamlingstekniker**

För att kunna utveckla och bearbeta krav, måste det finnas material som analysen stöds på. Fastän kunden oftast berättar vad de vill ha, behövs en djupare forskning för att få fram alla viktiga detaljer och aspekter.

Tekniker finns många och alla kan inte alltid utnyttjas inom ett projekt. Teknikerna fungerar bäst ifall de kombineras rätt i enlighet med projektets typ och karaktär. Exempel på tekniker karakteriseras i Figur 7.

#### **2.4.1 Val av kravinsamlingsteknik**

Valet av kravinsamlingsteknik beror till en stor del på vilka källor som är tillgängliga och vilken information söks. Teknikerna är även olika till sin natur; vissa ger generell information, medan andra ger specifik information för ett visst område. Teknikerna har olika behov, både i pengar, i kunskaper och i andra resurser.

Då det skall beslutas om vilka tekniker som skall utnyttjas i ett IT-projekt måste det först och främst tas i beaktande om det finns något tidigare system, det vill säga går det att göra en bakgrundsforskning och finns det tidigare användare att intervjua. Det är även bra att börja med bakgrundsforskningen då tillräckligt med information samlas in för att t.ex. kunna fortsätta med intervjuer och göra upp bra intervjufrågor eller -ämnen. Ifall det är någon viss funktion som är kritisk för systemet kan det vara en bra idé att använda sig av användningsfall. Figur 7 ger en översikt över olika kravinsamlingstekniker samt deras fördelar och nackdelar.

	Bakgrundsforskning	Intervjuer	Dokumentgranskning	Uppgiftsobservation	Användningsfall	Brainstorming	Krav återanvändning
<b>Möjliga begränsningar:</b>							
Kräver existerande system	x			x			x
Kräver kundens kravdokument							x
Kräver betydelsefull expertis					x	x	
I huvudsak användbar i första skeden	x			x			x
I huvudsak användbar i senare skeden					x	x	
Relativt dyrt		x			x	x	x
Oflexibel			x				
Tar fram endast tyst kunskap		x	x		x		x
Tar fram endast dokumenterad kunskap	x			x			x
<b>Möjliga fördelar:</b>							
Bred räckvidd		x	x				x
Bra för att få fram bakgrundsinformation	x	x					
Bra för att få fram problemdomäninformation	x	x	x		x		
Bra för att få fram krav		x			x	x	x
Bred kommunikation		x	x				
Enkelt att tillämpa	x	x	x	x			
Relativt billigt	x		x	x			
Flexibelt		x			x	x	x
Kan nå en stor målgrupp			x				

Figur 7. Kravinsamlingstekniker baserat på Bray (2002 s 50).

## 2.4.2 Bakgrundsforskning

Denna teknik kanske inte alltid nämns, då den är ganska självklar, men det är en viktig och oftast inledande teknik för att börja samla in kraven. Dokument såsom affärsplaner, operativa förfaranden, tekniska manualer för gränssnittssystem och användarmanualer för existerande system kan användas som källor. En viktig detalj att observera då material läses igenom är att anteckna den viktigaste och mest väsentliga informationen. Det är då lättare att i efterhand dra nytta av och dela med sig bakgrundsforskningen. (Bray 2002 s 223)

## 2.4.3 Intervjuer

Intervjuer är kanske den vanligaste och mest dominerande tekniken som används idag, för att få fram krav. Detta vidhåller både Sommerville (2004 s 152-153), Bray (2002 s



224-228) och Maciaszek (2001 s 82). Deras tankar genomsyrar hela detta underkapitel. Intervjuer är en mycket effektiv teknik för att få fram tyst kunskap, det vill säga kunskap som inte är dokumenterad utan existerar inuti avnämarnas huvuden. Oftast används andra tekniker som stöd för intervjuerna. Till intervjuens styrka hör

- mångsidig kommunikation som kan bestå av bl.a. verbal kommunikation, kroppsspråk och skisser
- flexibilitet då olika ämnen kan tas upp i farten/ad hoc.

Man brukar ibland skilja på strukturerad och ostrukturerad intervju. Frågor som låser intervjuens gång och innehåll görs upp för en strukturerad intervju. För en ostrukturerad intervju bestäms endast de olika ämnen som skall tas upp och intervjugången är sedan beroende av intervjuarens erfarenhet och sociala kompetens. I praktiken kombineras oftast dessa två former av intervjuer, eftersom en intervju är bra att planera på förhand. Det lönar sig dock inte att planera i minsta detalj, eftersom avstickare och sidoämnen under en intervju oftast är välkomna och gynnsamma. Ibland kallas intervjutyperna även öppna och slutna intervjuer. Fördelen med en ostrukturerad eller öppen intervju är att den ger intervjupersonen möjlighet att berätta om ämnen och ställa frågor som intervjuaren kanske inte skulle ha förstått att ställa.

Intervjuer är bra för att få en generell förståelse för vad avnämarna gör, hur de kommunicerar och vilka problem de har med det nuvarande systemet. Oftast är intervjupersonerna engagerade att prata om sitt arbete och samarbetar gärna.

För att få fram domänkrav är intervjuerna inte den bästa tekniken eftersom alla applikationsspecialister använder termer och uttryck som är unika för deras domän. Det är omöjligt för dem att diskutera domänkrav utan denna terminologi. Eftersom de oftast använder denna terminologi på ett diskret sätt kan det lätt leda till missförstånd. Vissa domänkrav är även så vardagliga och självklara för intervjupersonerna att de antingen tycker att det är för svårt att förklara dem eller så självklara att det inte är värt att nämna dem.

Kännetecknen för en bra intervjuare är:

- fördomsfrihet. Intervjuaren undviker förutfattade åsikter om kraven och är villig att lyssna på avnämarna. Ifall avnämaren uttrycker ett överraskande krav, skall intervjuaren kunna ändra sin åsikt om systemet.
- uppmuntrar intervjupersonen till diskussion genom att ställa frågor, kravförslag eller genom att föreslå samarbete med prototypen. Oftast fås inte mycket resultat genom att be intervjupersonen att berätta vad de vill ha, människor har lättare att diskutera om ämnet är definierat och inte flummigt.
- kan lyssna och snappa upp det viktigaste under intervjun och vid behov improvisera då intervjun stöter på ett viktigt ämne för intervjupersonen.
- kunna kommunicera på intervjupersonens egen nivå utan att nonchalera eller få intervjupersonen att känna sig okunnig.
- få intervjupersonen känna sig som en expert och inte som om det vore ett förhör.

Frågeställningar som en bra intervjuare borde undvika är:

- påstridiga frågor där intervjuaren uttrycker direkt eller indirekt sina egna åsikter
- partiska frågor. Dessa liknar påstridiga frågor men till skillnad märks intervjuarens åsikt mycket tydligt
- påtvingande frågor där intervjuaren formulerar frågan så att svaret styrs i riktning mot intervjuarens åsikt.

Fastän intervjuer är en effektiv och mycket använd teknik skall den inte ensam stå för kravmaterialet. Andra tekniker bör användas som stöd för intervjuerna.

### **Förberedelser**

En stor del av intervjuförberedelserna består av bondförnuft och goda organisationskunskaper. Frågorna och ämnen som förbereds baseras på den informationen som redan erhållits genom kravinsamlingen och kundens önskemål. Eftersom intervjuerna oftast är tidsbegränsade lönar det sig att justera frågemängden så att alla viktiga diskussionsämnen ryms med och att även reservera lite tid i slutet av en intervju för en snabb genomgång av det diskuterade med intervjupersonen. En timme brukar vara en tillräcklig tid för en intervju.

Gällande intervjuens plats, är det bra ifall det orsakar så lite som möjligt störning i intervjupersonens vardag, men på en plats där intervjun kan fortskrida i lugn och ro utan störningar. Vid behov lönar det sig att ha tillgång till existerande system, ifall en oplanerad observation skulle vara lämplig.

Till förberedelserna hör även val av dokumenteringsmetod, d.v.s. hur skall informationen sparas? Det finns i princip tre alternativ, antingen skriver intervjuaren själv ner anteckningar, ber en kollega skriva anteckningar eller använder ljud- eller videoinspelningar under intervjun. Problemet med att själv skriva anteckningar är att man är tvungen att göra två saker samtidigt, både lyssna och skriva. För att inte det skall uppstå långa tystnader som gör situationen obekvämt för intervjupersonen är intervjuaren då tvungen att antingen skriva mycket korta och snabba anteckningar eller sedan kunna skriva parallellt med lyssnandet. Ofta är det de mer erfarna intervjuarna som klarar av denna uppgift. Genom att ha en kollega närvarande, som och skriver ner anteckningar får intervjuaren mer frihet att koncentrera sig på diskussionen. Fastän detta är en fungerande lösning är den tyvärr dyr då den binder två personer istället för en. Ljud- eller videoinspelning är en enkel lösning på problemet, men för med sig sina egna nackdelar. Kroppsspråk och möjliga skisser eller demonstrationer missas med en ljudinspelning. Videoinspelning löser de problemen, men då inspelningen sker från ett stativ får man endast en vinkel på det hela och filmandet kan mycket väl hämma intervjupersonens normala beteende. Båda dessa metoder innebär en extra analyskostnad, eftersom det inspelade materialet måste gås igenom och det viktiga plockas ut. En vanlig och effektiv lösning på problemet är att intervjuaren själv skriver anteckningar samtidigt som ljudet spelas in på band. Då finns en så kallad säkerhetskopia som kan användas ifall anteckningarna innehåller otydligheter.

Själva arbetssättet under en intervju är också kopplat till bondförnuft och folkvett. Samarbetet mellan intervjupersonen och intervjuaren borde löpa smidigt, vilket ligger till stor del på intervjuarens ansvar. Självklarheter som artighet, punktlighet, vänlighet och förmåga att se diskussionerna ur intervjupersonens perspektiv är viktiga. Det är även viktigt att få intervjupersonen att känna sig lugn och kunnig – detta underlättar diskussionens smidighet. Följande signaler bör undvikas:

- min tid är viktigare än din
- jag vet inte vad jag gör
- du vet inte vad du talar om.

Språket under intervjun skall vara anpassat till intervjupersonens miljö och kunskap, alltför många nya eller tekniska ord skall undvikas. Det är även viktigt att fråga ifall det är något som intervjupersonen säger men intervjuaren inte förstår. Detta kan även göras senare i intervjun ifall tillfället inte just då ger möjlighet till det.

Det kan vara värt att lägga märke till ifall intervjupersonen verkar förvånad eller förundrad över någon viss fråga, eftersom detta kan vara ett tecken på att de antaganden som gjorts före frågorna har varit felaktiga.

Själva frågorna kan behandla problemet i fråga, processen för att utveckla en lösning och själva kravinsamlingen. Den första sortens frågor är den viktigaste och de är starkt relaterade till det projekt som kravinsamlingen angår. Andra sortens frågor ger inte direkta resultat i kravdokumentet, men kan vara av betydelse för själva projektet. Sådana är frågor om vem som betalar, vilka de underliggande orsakerna för projektet är och var gränsen mellan kostnad och pålitlighet går. Den tredje sortens frågor, om själva kravinsamlingsprocessen, kan vara bra att ställa för att bekräfta att kravinsamlingen görs effektivt. Några exempel på frågor:

- Verkar frågorna relevanta?
- Är dina svar officiella?
- Är du den rätta personen att svara på dessa frågor?
- Frågar jag för många frågor?
- Borde jag fråga dig något annat?
- Är det någonting du vill fråga?
- Finns det någon annan person jag borde träffa?
- Finns det någon med i projektet som inte behövs?

För att sammanfatta kan konstateras att intervjuer är en populär kravinsamlingsteknik och att de är mycket flexibla. Mycket är beroende av intervjuarens kunskap och erfarenhet i ämnet och sociala färdigheter.

#### **2.4.4 Enkäter**

En enkät är en bra teknik att utnyttja då information från många kunder vill samlas in på en gång. För att minska risken för mångtydighet och missförstånd är det mycket viktigt att planera frågorna noggrant och att formulera frågorna korrekt, då svaren är begränsade genom alternativ. Enkäter är en passiv teknik och detta kan vara både en fördel och en nackdel. Det är en fördel eftersom svarspersonen har tid att tänka över sina svar och svara anonymt. Det är en nackdel då svarspersonen inte har möjlighet att förklara i detalj eller förklara orsaken till svaren eller att något av svarsalternativen inte känns hundra procent rätt. Fastän inflexibiliteten är en nackdel, kan denna teknik med rätta och välformulerade frågor ge ett relativt kostnadseffektivt sätt att få fram kravmaterial av en stor mängd kunder.

Den rekommenderade frågeformen är slutna svar med statistiska svarsmöjligheter. Whitten och Bentley (1998) definierar tre frågetyper:

- Flervalsfråga, där svarspersonen väljer ett eller flere svar från en uppsättning av givna svar. En tilläggskommentar kan även tillåtas.
- Graderingsfråga, där svarspersonen uttrycker sin åsikt om ett visst påstående på en verbal skala t.ex. instämmer helt, instämmer, neutral, instämmer inte, instämmer inte alls och jag vet inte.
- Rankingfråga, där de givna svaren skall rankas med siffror t.ex. 1-5, procentvärden eller liknande sorteringsmöjligheter.

Eftersom det även kan användas öppna svar i en enkät kan denna teknik ge omfattande svar, men då ökar även analyskostnaderna. Ifall enkäten är lång och tidskrävande att fylla i kanske svarsprocenten inte blir så hög.

Denna teknik kan dock bra betraktas som en kompletteringsteknik till intervjuer, då den används med måtta. (Bray 2002 s 229 och Maciaszek 2001 s 83-84)

### **2.4.5 Dokumentgranskning**

För att kunna använda dokumentgranskning förutsätts det att det finns ett existerande system och att en stor del av informationen om problemet finns dokumenterad. Dokumentgranskning är en värdefull och lättillgänglig teknik att använda då det finns ett existerande system. För att få fram kravmaterialet skall alla dokument som är indata, utdata eller interndata till systemet studeras. Dock kan det inte antas att dessa dokument ger en översiktsbild över systemets funktionalitet och därför skall denna teknik användas i samband med andra tekniker, såsom intervjuer och observationer.

En risk med dokumentgranskning är den möjliga sammanblandningen mellan det dokumenterade systemet och det aktuella systemet. Det är inte omöjligt att det dokumenterade systemet har vissa brister som användarna har kunnat kringgå genom fyndighet och på så sätt fått systemet att fungera på ett tillfredsställande sätt. Även denna risk förminskas genom att använda dokumentgranskning i samband med andra tekniker. (Bray 2002 s 229-230)

### **2.4.6 Observation**

Det finns situationer där det på basis av intervjuer och enkäter kan vara svårt att få ett grepp om hur det tänkta systemet skall fungera. Kunden kanske inte kan förmedla informationen rätt eller har inte en fullständig bild över hela affärsprocessen. I en sådan situation kan observation vara en bra kravinsamlingsteknik. För att låna Maciaszeks jordnära exempel – bästa sättet att lära sig knyta en slips är att observera processen. Oplanerade observationer, som kan dyka upp t.ex. under en intervju då intervjupersonen visar någon uppgift eller problem i praktiken, kan ge värdefull förstahandsinformation om problemet. Eftersom dessa tillfällen är oplanerade, kan den informationen man får av dem då de sker endast tas tillvara.

Planerade observationer går ut på att följa med och observera, då en person utför en viss uppgift, som oftast har med ett existerande systemet att göra. Denna teknik kan vara mycket givande i kravhantering, då den ger en praktisk inblick i hur det existerande systemet används och gör att möjliga skillnader mellan det existerande och det dokumenterade systemet upptäcks (som kan uppkomma då dokumentgranskning utförs).

Beslutet om vilka uppgifter som skall utföras är oftast svårare att fatta än vad själva utförandet av observationen är. Observation och den efterföljande obligatoriska analysen är kostsamt och risken finns att man blir för ivrig och samlar in mer information än nödvändigt. Rekommendationen är att använda observation sparsamt och koncentrera sig på kritiska interaktiva uppgifter och använda sig av stickprov om möjligt.

Själva praktiska arrangemang kring observation är liknande som under en intervju; målen, personal och plats skall väljas och de lämpliga förberedelserna skall göras med stor noggrannhet. I stil med intervjuer är skriftliga anteckningar kanske den bästa dokumenteringsmetoden, men videoinspelning är även ett godtagbart alternativ. Omgivningen bör vara så naturtrogen som möjligt och oftast är det observatörens uppgift att endast passivt observera, medan intervjupersonen kan ombes förklara med ord delmoment av uppgiften. Detta kan ge mer information än endast passiv observation och kan därmed motiveras fastän omgivningen görs mindre naturlig. Ibland kan även intervjupersonen av olika orsaker inte bete sig normalt i en observationssituation. Olika metoder för att motverka detta kan vara:

- grundligt förklara orsaken till observationen
- utvidgad observation, så att situationen känns mindre nervös
- hemlig observation, inte att rekommendera i alla sammanhang.

Annorlunda beteende kan vara att utföra uppgifter enligt regler och direktiv och inte på samma sätt som vanligen. Detta ger emellertid inte ett realistiskt resultat då den värdefulla informationen kan ligga i olika genvägar personen i normala fall använder sig av, vilket kan vara både positivt och negativt. Observation kan även utnyttjas då en prototyp skall testas. Då måste mera tid budgeteras för personens introduktion till uppgiften och prototypmiljön.

### **Etnografi**

En viss variant av ämnet observation är känt som etnografi. Etnografi betyder att observatören under en längre period blir integrerad i samhället eller kulturen som undersöks. Eftersom mjukvarusystem inte existerar i isolation, utan de används genom social och organisationssamverkan, så borde även kraven fås fram från dessa sammanhang.

Detta är en kostnadskrävande teknik men kan vara motiverad då komplexa och kritiska system skall analyseras och en djup förståelse för problemdomänen är avgörande. Sommerville ger som exempel flygledning, underjordiska järnvägskontrollrum och finansiella system. Etnografi är speciellt effektivt att använda, då krav som baserar sig på hur människor verkligen jobbar och krav som baserar sig på samarbetet och medvetenheten för andra medarbetares arbete, skall tas fram. (Bray 2002 s 230-232, Sommerville 2004 s 157-158 och Maciaszek 2001 s 84)

### 2.4.7 Användningsfall

På engelska används termen *use case*. Detta uttryck används mycket inom branschen, men det korrekta på svenska är att tala om användningsfall.

Användningsfall är en scenariobaserad teknik som introducerades på 1990-talet och är nu en standardegenskap i UML (Unified Modeling Language) för att beskriva objektorienterade systemmodeller. I UML fångas det beteende, som är utifrån synligt och testbart, in i användningsfall. Ett användningsfall utför en funktion som är utifrån synlig för en aktör och kan i ett senare skede testas separat.

En aktör representerar vem eller vad som helst som kommunicerar med ett användningsfall, vilket kan vara en människa, maskin m.m. som förväntar sig ett nyttigt resultat. Normalt brukar dessa aktörer representeras av så kallade streckgubbar i bilder se exempel Figur 8, trots att det då kan handla om t.ex. en maskin som aktör.



Figur 8. Exempel på aktörer i ett användningsfall.

Ett användningsfall representerar en fullständig enhet av funktioner som är av värde för en aktör. I så gott som alla användningsfall handlar det alltså om en aktör som kommunicerar med systemet. Användningsfall kan fås fram genom att gå igenom de olika uppgifterna en aktör vill göra med systemet. En bra fråga att använda sig av är: vad är aktö-



rens ansvar gentemot systemet och vilka är förväntningarna av systemet? Användningsfall kan även fås fram genom direkt analys av de funktionella kraven.

Ett användningsfalldokument kan ha följande struktur:

- Kort beskrivning
- Vilka aktörer är aktuella
- Förutsättningar – vad skall vara gjort innan användningsfallet kan börja
- Fullständig beskrivning av händelseförlopp som består av
  - Huvudhändelseförloppet
    - Mindre händelseförlopp, dessa kan vara flera och även uppdelade för att öka läsbarheten
  - Alternativa händelseförlopp, för att specificera undantagssituationer
- Tillståndet efter användningsfallet – vilka skillnader i systemet har uppstått

(Maciaszek 2001 s 49-53, Bray 2002 s 232 och Sommerville 2004 s 154-156)

#### **2.4.8 Brainstorming**

Brainstorming är ett låneord från engelskan. På svenska finns termen *idékläckning*, som tyvärr inte fått fotfäste, därför används termen brainstorming här (Wikipedia, Brainstorming). Brainstorming går ut på att flera personer i en grupp fritt får diskutera och kasta in idéer. Målet är att okritiskt generera så många lösningar som möjligt på ett problem. Bray rekommenderar relativt korta sessioner på ca en timme eftersom de har visat sig vara de mest produktiva. Eftersom brainstormingsessionerna är kostsamma skall de användas med måtta inom projekt. Några regler för en brainstormingsession är

- Alla får säga sin åsikt
- Ingen kritik tillåten, hur galna idéer som än presenteras
- Ingen debatt, andras idéer får gärna användas som inspiration

Det kan även vara en bra idé att ha en stor tavla där alla kan skriva ner och visualisera sina idéer och försöka få alla deltagare att aktivt delta. Som kravinsamlingsteknik kan tänkas att idéerna skulle bestå av krav. (Bray 2002 s 233)

Efter brainstormingsessionen skall alla krav eller idéer samlas in och med ett kritiskt öga gås igenom och sällas för att sedan utnyttjas till analysen.

#### **2.4.9 Kravåteranvändning**

Kravåteranvändning kan användas om det från tidigare finns en kravspecifikation av en kund eller tidigare liknande system. Enskilda krav återvinns från den ursprungliga kravspecifikationen och används till den nya. Det kan finnas orsak till kritisk genomgång av den tidigare kravspecifikationen, eftersom den kan vara av dålig kvalitet. Man kan då endast utnyttja de välmotiverade viktigaste bitarna och skriva om dem till en välkonstruerad och strukturerad ny kravspecifikation. Kravåtervändning är en bra teknik att använda i början av ett projekt då en överblick av problemdomänen och önskemålen för det nya systemet skall tas fram. (Bray 2002 s 234)

### **2.5 Analys**

Definitionen på analys är enligt Bray (2002 s. 24) att genom en studie av problemdomänen, uppnå förståelse om och dokumentation av egenskaper för domänen och problemen (som kräver lösning) som finns inom den domänen. Denna definition kanske är aningen svårformulerad och i början svår att förstå, men innefattar det väsentliga inom problemanalysen. Alla olika analyser borde ge en förklaring på följande områden (Bray 2002 s 128):

- Problemdomänens struktur med tanke på underdomänen och deras förhållanden sinsemellan
- Problemdomändata
- De medfödda egenskaperna och beteenden av problemunderdomänen
- Betydelsefulla händelser och fenomen inuti problemdomänen
- Kraven (de effekter som systemet skall producera inom problemdomänen)

Det finns dock olika metoder för att få fram denna information. Strukturerad analys har hållit i sig i många årtionden, objektorienterad analys är en modern favorit och problemdomänorienterad analys är en lovande nykomling. (Bray 2002 s 128-129) Dessa tre metoder presenteras nedan noggrannare.

### **2.5.1 Strukturerad analys**

Strukturerad analys är utvecklat från den klassiska systemanalysen från 1960- och 1970-talen. Strukturerad analys (hädanefter förkortat SA) blev populär på 1980-talet och används ännu idag. (Wikipedia, Structured analysis)

Det allmänna SA-tillvägagångssättet är att skapa olika modeller av processerna, dataflöden och sparade datastrukturer i ett system, men det systemet är oftast inte problemdomänen. Inom SA nämns krav inte på något speciellt sätt, utan det finns ett underliggande antagande att det tidigare systemet redan uppfyller alla krav, fränsett att det är ett datoriserat system. Inom modern SA har idén att först göra en modell av det tidigare systemet istället för att genast gå in på att göra en modell av det nya systemet övergivits, eftersom det konstaterades vara ineffektivt. De utnyttjade modellerna verkar dock vara rimligtvis intuitiva och tillgängliga för utvecklare och kunder. Under rätta omständigheter kan den ursprungliga tanken om att göra en modell av det existerande systemet stöda det primära analysmålet om att förstå problemdomänen. Inom SA används metoder såsom bl.a. dataflödesdiagram, data lexikon, ER-diagram, ELH (Entity Life History) och processspecifikationer. (Bray 2002 s 54-59)

### **2.5.2 Objektorienterad analys**

Objektorienterad analys (hädanefter förkortat OOA) är mycket nära förknippat med objektorienterad design. OOA ger en ypperlig grund för den interna designen. OOA har en del gemensamt med SA. De koncentrerar sig båda på strukturella modeller, mycket vikt läggs på att modellera, ingendera nämner kravinsamling specifikt men den skall göras, det finns ingen klar skillnad mellan analys och specifikation och alla problemdomän antas kunna behandlas likadant. Användningsfall är mycket använda inom OOA, fastän de kan betraktas som en självständig teknik.

Grundläggande riktlinjer för OOA är:

- Identifiera objektklasserna inom problemdomänen
- Definiera attributen och metoderna för klasserna
- Definiera beteendet för klasserna
- Ta fram en relationsmodell för klasserna.

OOA har som utgångspunkt att ett kravdokument redan är gjort och problemdomänanalysen även är gjord. Det som OOA koncentrerar sig på mera är högnivådesign av systemet. Förutsatt att problemdomänen är välutforskat eller så självklar att det inte krävs någon djupare analys, är OOA ett mycket starkt alternativ och förklarar varför den har klarat sig så bra på marknaden. (Bray 2002 s 77-92)

### **2.5.3 Problemomänorienterad analys**

Problemomänorienterad analys (hädanefter förkortat PDOA) är en metod som tar vara på gamla synsätt men som ännu håller på att utvecklas. Till skillnad från SA och OOA lägger PDOA mer vikt på beskrivningen än på modelleringen. Då det är lämpligt utnyttjas samma metoder som de andra sätten använder sig av, men i huvudsak används ändå text som metod i PDOA.

PDOA indelar entydigt på beskrivningarna i två kategorier – de som är knutna till problemomänen och de som angår det krävda beteendet i lösningssystemet. Det rekommenderas att två skilda dokument används för denna uppdelning. Det första dokumentet skall innehålla en beskrivning på aktuella delar inom problemomänen och en lista över problem som skall lösas inom domänen, d.v.s. kraven. Det andra dokumentet skall innehålla en beskrivning av beteendet som krävs av systemet för att kunna uppfylla kraven från första dokumentet, d.v.s. specifikationen. Endast det första dokumentet berörs av analysen, det andra är ett resultat av specifikationskedet. PDOA:s riktlinjer är:

- Samla in basinformation och utveckla problemramen eller problemramarna för att kunna fastslå problemomäntypen

- Med problemramen eller ramarnas vägledning insamlas mera detaljer och utvecklas en beskrivning av de relevanta dragen i problemdomänen. Kritiska i detta skede är riktlinjerna utvecklade av Kovitz (1999), som räknar upp vilka delar av problemdomänen som skall utforskas baserat på vilken typ den är
- I samband med de föregående insamlas och dokumenteras kraven för det nya systemet

Jackson (1995 s 158-162) introducerade en ny modell, problemramen, som hjälper både att separera kraven från problemdomänens inneboende egenskaper och att klassificera problemdomänen i olika typer. Inom PDOA behandlas inte olika problemdomän på samma sätt, utan beroende av problemdomänens typ samlas olika slag av information in.

### **Problemramar**

Jackson (1995 s 159) definierar en problemram som en generalisering av en typ av problem och framhäver att alla problem inte alltid passar in i just en problemram, utan man kan vara tvungen att kombinera olika problemramar. Bray (2002 s 93) definierar problemramar som en modell av problemdomänen som består av ett antal inbördes relaterade underdomän, där en underdomän är vilken del som helst av problemdomänen som lämpligen kan pekas ut. Bray understryker även skillnaden mellan sammanhangsdiagram, då sammanhangsdiagram modellerar systemet och problemramar modellerar problemen. Problemramarnas mål är även att samla in betydligt mer information om problemdomänen.

Den stora skillnaden mellan PDOA och andra metoder är sättet som den kategoriserar problemdomänen och hanteras i enlighet med det. Jackson identifierar följande kategorier:

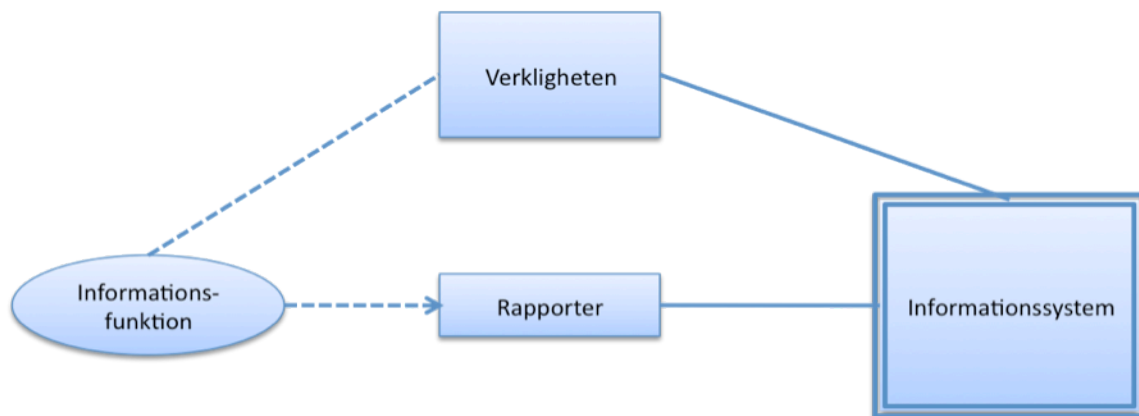
- Verktygssystem – systemet utför styrda operationer på realiserade objekt (som endast existerar inom systemet), t.ex. ett ordbehandlingsprogram
- Kontrollsystem – systemet kontrollerar beteendet för en del av problemdomänen, t.ex. ett hisskontrollsystem

- Informationssystem – systemet hanterar informationsförfrågningar inom problemdomänen, t.ex. olika register
- Omvandlingssystem – systemet omvandlar indata i någon viss form till utdata i någon annan form, t.ex. faktureringsystem som läser in bankfiler
- Förbindelsesystem – systemet hanterar kommunikationen mellan underdomän som inte är direkt kopplade, t.ex. videokonferenssystem

Eftersom den fallstudien som tas upp i detta examensarbete klassas som ett informationssystem enligt denna uppdelning, kommer de andra kategorierna inte att noggrannare beskrivas i detta examensarbete.

### Informationssystemets problemram

Informationssystem existerar för att förse information om delar i problemdomänen och är mycket vanliga, men har trots det även tidigare varit dåligt förstådda. Bray (2002 s 102) nämner två olika typer av informationssystemramar, en enkel variant och en mera komplex variant. Den enkla varianten som förmedlar information automatiskt, oftast kontinuerligt, dess problemram demonstreras i Figur 9. Den mer komplexa varianten demonstreras i Figur 10.

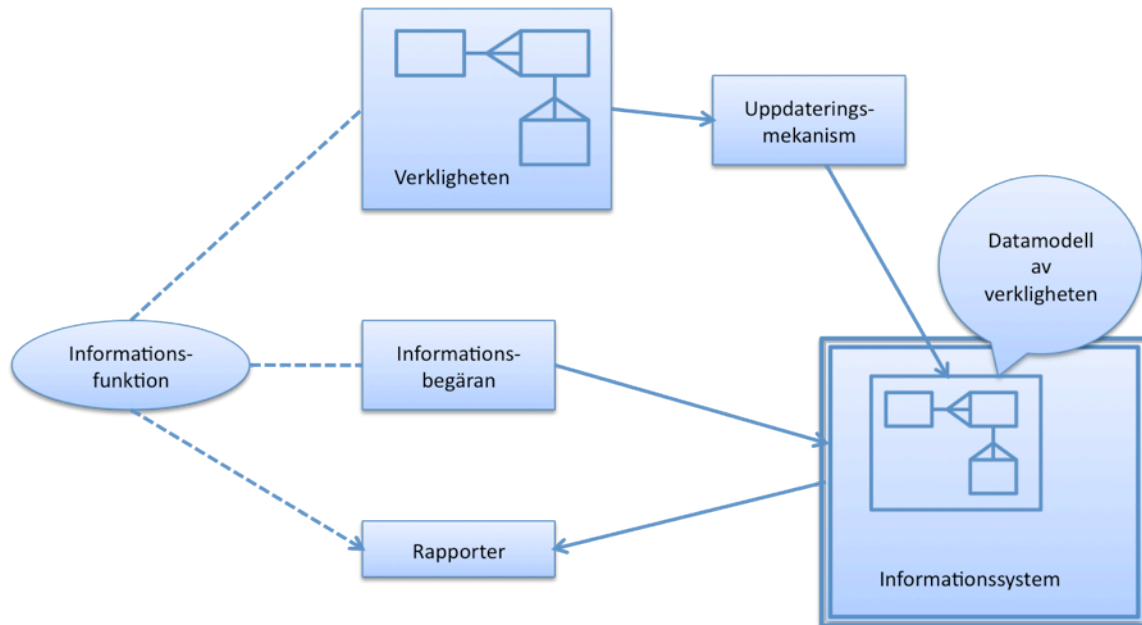


Figur 9. Problemramen för ett enkelt informationssystem (Bray 2002 s 103).

Figur 9 visar hur informationssystemet fördelar information om vissa delar i ”verkligheten” i form av rapporter enligt någon viss informationsfunktion.

I den mera invecklade problemramen demonstrerad i Figur 10 har strukturen utvecklats en hel del. Information fördelas av informationssystemet på basis av informationsbegä-

ran och då systemet hanterar data så består den relevanta delen av verkligheten av dataenheter. Dessa kan representeras av en datamodell, oftast i form av ett ER-diagram. I denna problemram finns det även två typer av indata, uppdateringar gällande problemområdets status och informationsbegäran om problemområdet. Det är även möjligt att verkligheten är en statisk domän, såsom t.ex. en CD skiva, men i de flesta fall är domänen dynamisk, t.ex. en databas och informationssystemet innehåller en datamodell av den.



Figur 10. Problemramen för ett mera invecklat informationssystem (Bray 2002 s 105).

Riktlinjer för kravdokumentets innehåll kan studeras i Figur 11, men observera att endast informationsfunktionerna representerar själva kraven och resten är beskrivningar av problemområdet.

<b>Informationsproblem - kravdokumentet</b>	
<b>Innehåll</b>	<b>Vissa relevanta tekniker</b>
Datamodellen för underdomänen(entiteterna) i problemdomänen som skall beskrivas (deras attribut och förhållanden)	ER-diagram och datadiagram
Egenskaperna för varje problemunderdomän inklusive alla problemdomänhändelser som ändrar på problemdomänens status och alla möjliga sekvenser som de händelserna kan ske i	Text, händelselista och enhet levnadshistoria (ELH)
Hur systemet får tillgång till den aktuella underdomänens status och händelser i problemdomänen	Text
Snedvridningar och förseningar som införts av någon sambandsdomän	Text
Då initialiseringsdata skall återanvändas från existerande filer, de relevanta formaten och tillgångsprocedurerna	Filkartor, strukturkartor och BNF (Backus-Naur-Form)
Informationsfunktionerna, d.v.s. de rapporter som behövs, deras relation till verklighetens status och då det är relevant, förfrågningarna som skall stödas	Ritningar, text och tabeller

Figur 11. Riktlinjer för kravdokumentet (Bray 2002 s 106).

Problemramar ger en del fördelar till PDOA, vilka Bray (2002 s 114-115) räknar upp:

- De uppmuntrar till en tidig fokusering på problemdomänen och kraven
- De hjälper att identifiera vilken typ av problemdomän det handlar om
- De ger rum för förhållanden mellan domänen
- De ger specifika riktlinjer för hur varje enskild problemtyp skall hanteras

#### **2.5.4 Kravdokumentet**

Själva kravdokumentets uppgift är att förmedla den nödvändiga informationen om problemdomänen och kundens krav till utvecklarna av systemet. Denna roll inverkar både på innehållet och på stilen i dokumentet. Målet för kravdokumentet är att vara entydigt, välorganiserat, fullständig, konsekvent, validerbart och modifierbart. Eftersom kunden



skall kunna läsa igenom och granska dokumentet är det viktigt att hålla läsbarheten hög och undvika teknisk jargong.

Kravdokumentet har i princip två huvudparter att redogöra för:

- Hurdan problemdomän det är fråga om, dess olika elements natur och deras växelverkan
- Vad kraven är och vilka effekter kunden vill att det nya systemet skall ha inom problemdomänen

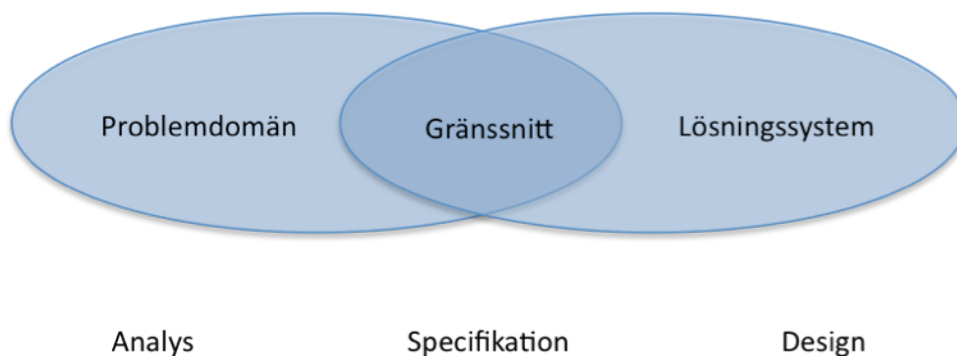
Det kan vara bra att särskilja dessa två punkter i dokumentet. Ett förslag på strukturen i kravdokumentet är enligt Bray (2002 s 131):

- Dokumentdetaljer (titel, myndighet, revisionshistoria o.s.v.)
- Problemdomänbeskrivning
  - Översikt (text + sammanhangsdiagram, problemram, datamodell)
  - Underdomän – deras karaktärer, beteende m.m.
- Krav (i största allmänhet textbaserad)
  - Funktionella krav
  - Icke-funktionella krav
  - Domänkrav
- Dataordlistor
- Referenser (till relaterade dokument, t.ex. specifikationen för en underdomän)

## 2.6 Specifikation

Specifikationen definierar Bray (2002 s 135) som skapandet och definierandet av ett lösningssystemets beteende så att det producerar de krävda effekterna på problemdomänen. Beteendet fortsätter Bray att definiera som gränssnittet mellan lösningssystemet och de delar av problemdomänen som systemet kommer att kommunicera med. Specifikationen är med andra ord systemkraven.

Specifikationen är en kritisk del inom kravhanteringen och därmed hela programvaruutvecklingen. Bray (2002 s 137) lägger de olika skedena i proportion till varandra i Figur 12 där han illustrerar hur specifikationen är både en del av problemdomänen och samtidigt av lösningssystemet.



Figur 12. Förhållandet analys - specifikation – design.

Specifikationen är i grund och botten en definition av lösningssystemets beteende. Bray rekommenderar att bästa sättet är att definiera indata och utdata till systemet och deras förhållanden av både orsaks- och tidstyp till varandra. Fastän det förekommer olika nivåer av formalitet är definitionerna oftast textbaserade. Naturligt språk är ett kraftigt verktyg men vissa av de obligatoriska definitionerna kan ändå vara bäst att presentera med olika modelltekniker.

Då andra metoder inte gör en lika klar skillnad mellan analys och specifikation, är det svårt att beskriva endast specifikationsdelen inom de metoderna, men det är skäl att känna till dem, speciellt formell specifikation enligt Bray (2002 s 171). Eftersom specifikationen inte var en del av fallstudien, tas inte andra metodernas specifikationer upp.

### 2.6.1 Extern design

Själva termen extern design förkortar Bray till edesign och definierar den som skapandet av det yttre utseendet och egenskaperna hos ett system (Bray 2002 s 396).

För att inte djupare gå in på intern design understryker Bray att endast indata och utdata och deras förhållanden sinsemellan skall definieras. Själva kraven som resulterade från analysen skall omvandlas till funktioner. Specifikationen kan skrivas på naturligt språk, men då det är till fördel kan olika modelleringstekniker användas. Bray nämner dataflö-

desdiagram och dataordlistor som exempel. Då indata skall definieras finns det tre olika mekanismer som förmedlar det vidare till systemet: dataström, datapool och parametrar.

Dataströmgränssnitt är de vanligaste och karakteristiskt för dessa är FIFO (först in först ut), buffring och då data är läst raderas det från strömmen. För att definiera dataströmmar krävs att giltiga strömsekvenser (då det finns många kanaler tilldela data till kanalerna) definieras och möjliga timings- eller buffringsomständigheter ses över.

En datapool är oftast en fil eller ett område i minnet som delas mellan de två domänerna i fråga. För att definiera en datapool krävs då det är fråga om mångdimensionella pooler en datakarta, där det framkommer var data definieras och att läs- och skrivprotokollen definieras.

Det räcker inte att specificera indata och utdata utan även deras förhållanden är minst lika viktiga. Dessa kan beskrivas som händelsesvar, eftersom indata egentligen är händelser som systemet skall svara på. Systemet kan svara på några olika sätt, det kan producera någon form av utdata, ändra sin interna status eller kombinationen av dessa två. Bray (2002 s 150) kategoriserar dessa händelsesvar enligt följande:

- Hårdvarugränssnitt – då systemet kommunicerar med sensorer och andra fysiska objekt
- Användargränssnitt
- Operatörgränssnitt – då operatören är en superanvändare eller administratör med rättigheter att modifiera systemet
- API:er (Application Programming Interface) eller mjukvarugränssnitt

Relationerna kan specificeras både funktionellt och proceduralt. Skillnaden är att den förra specificerar slutresultatet och den senare specificerar ett möjligt tillvägagångssätt eller struktur också. Bray (2002 s 159) rekommenderar att funktionella specifikationer skrivs eftersom kunden inte har den domänkunskap som krävs för att kunna validera den procedurala specifikationen och utvecklaren begränsas av specifikationen vid den interna designen och programmeringen.

## 2.6.2 Specifikationsdokumentet

Specifikationsdokumentet är resultatet av specifikationen och ett mycket viktigt dokument, då det är på basis av det som utvecklarna börjar skapa systemet. Bray (2002 s 160-161) räknar upp andra viktiga orsaker varför det är värt att skriva en specifikation:

- För att ge full synlighet åt det krävda beteendet så att det kan gås igenom och formellt godkännas. På så sätt minimeras onödigt arbete.
- För att ge en säker dokumentering av det krävda beteendet och på så sätt minska beroendet av enskilda nyckelpersoner inom projektet. Detta är då mest en nytta för organisationen.
- För att försäkra att rätt information fördelas effektivt och ekonomiskt till projektgruppen och på så sätt minimera kommunikationsmissar.
- För att bistå med en grundlinje mot vilken funktionaliteten kan testas i ett senare skede och möjligtvis av tredje parter.
- För att bistå med den nödvändiga informationen till manualframställandet
- För att minimera eller helt och hållet utestänga missförstånd mellan leverantören och klienten om funktionaliteten
- För att underlätta framtida underhåll
- För att underlätta framtida utvecklingsprojekt med liknande systemfunktionalitet
- För att bistå med en entydig milstolpe inom projektet
- För att i framtiden noggrannare kunna uppskatta utvecklingsprestationer

För att kunna uppfylla dessa roller måste specifikationsdokumentet ha olika egenskaper såsom tydlighet, entydighet, välorganiserad, funktionell syn, fullständighet, konsekvens, möjliggöra automatik (kontroll, testgenerering, prototypgenerering), validerbarhet, modifierbarhet och spårbarhet (Bray 2002 s 161). Bray rekommenderar följande struktur för ett specifikationsdokument:

- Dokumentdetaljer (titel, myndighet, revisionshistoria o.s.v.)
- Översikt (över problemdomänen och lösningssystemet)

- Krav (duplicerade från kravdokumentet)
  - Funktionella krav
  - Icke-funktionella krav
  - Domänkrav
- Systembeteende (i allmänhet det längsta kapitlet)
- Referenser
- Ordlista (Dataordlistor)
- Index

## 2.7 Kravvalidering

Eftersom misstag kan ske i alla skeden av kravhanteringen är det viktigt att valideringen är en naturlig del av hela processen (Bray 2002 s 29-30). Sommerville (2004 s 158-160) definierar kravvalidering som bevis på att kraven verkligen definierar det system som kunden vill ha. Han framhäver också att ju senare i processen felet upptäcks desto dyrare blir de att åtgärda. Sommerville rekommenderar följande kontroller på kraven:

- *Valideringskontroller.* En användare kan tro att ett system behövs för att utföra en viss uppgift, men det kan visa sig efter utredning och analys att andra eller annorlunda funktioner också behövs. Då system har olika användare och alla har sina egna behov blir kraven så gott som alltid en kompromiss av alla behov.
- *Konsekvenskontroller.* Kraven i dokumentet får inte skapa konflikter sinsemellan. Det får inte finnas motstridiga begränsningar eller beskrivningar för samma funktion.
- *Fullständighetskontroller.* Kravdokumentet skall innehålla krav som definierar alla funktioner och begränsningar som lösningssystemet skall omfatta.
- *Verklighetskontroller.* Med hjälp av kunskap om existerande teknik, skall kraven kontrolleras för att försäkra sig om att de är genomförbara. Dessa kontroller borde även ta i beaktande budgeten och tidtabellen för projektet.

- *Kontrollerbarhet.* För att minimera möjliga dispyter mellan kunden och leverantören borde kraven alltid skrivas så att de kan kontrolleras. Kort sagt skall kraven kunna testas och bevisas vara uppfyllda.

Sommerville rekommenderar att följande valideringstekniker används enskilt eller kombinerade:

- *Kravrecension.* Kraven analyseras systematiskt av en grupp granskare som består av representanter för leverantören och kunden. Kan vara formell eller informell och båda baserar sig på kommunikation med och godkännande av kunden.
- *Prototyper.* I denna teknik framställs en fungerande modell av systemet och kunden får testa och utvärdera den för att kunna besluta ifall den uppfyller uppställda krav.
- *Testfallsframställning.* Krav borde kunna testas och ifall testfall utarbetas som en del av valideringsprocessen upptäcks oftast problematiska krav. Ifall testfallet är omöjligt eller mycket svårt att framställa är detta en indikation på att det även kommer att vara svårt att realisera kravet i systemet.

Bray (2002 s 211-213) gör ett tillägg med följande tekniker:

- *Enkla kontroller.* Noggrann genomgång av kravinsamlingsmaterialet och dubbelkoll av dokumenten.
- *Logisk analys.* En formell metod.
- *Utveckling av användarmanualen.* Ifall specifikationen är tillräckligt detaljerad går det att skriva användarmanualen i förväg.

### 3 FALLSTUDIEN

Fallstudien baserar sig på kravhanteringsprocessen för NJK:s registerprojekt och handlar om vilken vikt kravspecifikationen hade på offertförfrågningarna samt på diskussionen med leverantörer. Grunderna för hur offertrundan avslutades och vilken leverantör valdes kommer också kort att beskrivas.

Projektbeskrivningen:

Syftet med projektet är att förnya Nyländska Jaktklubbens (förkortat NJK) register. Registretets kärninformation består av 2600 medlemmar, 800 båtar, 350 hamnplatser och 170 vinterplatser. Projektets tidpunkt är från november 2009 till sommaren 2010. Som mål med projektet är att hitta en ny lösning till nuvarande MS Access baserade databas och att all viktig information för NJK:s verksamhet skall göras enhetligare och mer centraliserad men samtidigt mera tillgänglig.

Eftersom NJK inte var ute efter ett skräddarsytt system har inte den sista delen av kravhanteringsprocessen, specifikationen, utförts inom denna fallstudie. Målsättningen för NJK var från början att hitta en färdig produkt som i mån av möjlighet täcker största delen av kraven och sedan modifiera den för att uppfylla NJK:s krav. Kompromissannolikheterna var medräknade från början. Fallstudien kommer att analyseras i jämförelse med de olika teknikerna och metoderna som presenteras i kapitel 2.

#### 3.1 Bakgrund

NJK, som är Finlands största segelförening, har sitt kansli och sina klubbtrymmen i hemmahamnen Björkholmen i Helsingfors. Personalen består av klubbchef, två kanslist, chefstränare, två hamnmästare, fastighetsskötare, två ”Match Race Center”-ledare, seglingsmästare och en knatteansvarig. Utöver de anställda har föreningen en kommodor, en styrelse, ett tjugotal kommittéer och sektioner samt ett stort antal frivilliga medlemmar, som ställer upp som funktionärer. Självt har jag jobbat cirka fem år inom NJK som kanslist och registeransvarig och har fungerat som projektchef för registerprojektet. NJK har i cirka 10 år använt sig av en ”Microsoft Access”-databas, till vilken en del formulär och färdiga SQL-frågor är utvecklade. Det saknas ändå en användarvänlig databasapplikation. Access-registret har utvecklats av 2-3 frivilliga medlemmar och är på grund av detta brokig i både design och i det tänkta användningssättet.

### **3.1.1 Existerande system**

Idag är registret beroende av några nyckelpersoner för att göra nya sökningar, massfaktureringar, uppdateringar och för att underhålla registret. Det finns väldigt få manualer eller instruktioner tillgängliga och så gott som all kunskap är tyst och odokumenterad. Registret har låg användbarhet och är svårinlärt eftersom en ändamålsenlig databasapplikation saknas. Själva databasdesignen är från början bra men har med årens lopp föråldrats och är i behov av uppdatering. På grund av dessa orsaker har personalen löst registrets brister med hjälp av andra dokument och system. Detta har resulterat i informationsreduktion och svårigheter att upprätthålla och uppdatera flera dokument samtidigt. Den ursprungliga tanken med Access-databasen var att centralisera all information som är viktig för NJK:s verksamhet, men den tanken uppfylls inte längre av dagens register.

### **3.1.2 En segelförenings varierande behov**

Fastän man skulle tycka att marknaden är full av registerprogram riktade till föreningar och det är endast att välja och vraka vilken lösning man vill ha, visade det sig snabbt, att en segelförening och speciellt den största i Finland har mer utvecklade och krävande behov. Problem finns redan i medlemsregistret då man behöver hålla reda på alla utmärkelse- och förtjänsttecken en medlem mottagit under medlemskapet, vilket inte är en standardegenskap i ett registerprogram. En annan detalj som inte många färdiga programlösningar har är familjeutskick, d.v.s. möjligheten att koppla ihop medlemmarna som hör till samma familj och har samma postadress till en postning av både tidningar och fakturor. I Access-registret finns det en specifik ”betalare” kopplad till varje medlem och på så sätt kan medlemsfakturorna grupperas till en faktura, om t.ex. de minderåriga barnens faktura skall komma på mammans eller pappans namn. Denna funktion måste dessutom vara flexibel och inte begränsas av t.ex. postadressen då det finns många familjer som vill ha en gemensam räkning fastän barnen inte bor på samma adress längre. Ett stort hinder var även båtregistret, som inte normala föreningsregister innehåller. NJK har även ett behov av att administrera många olika typer av resurser, allt från hamnplatser till nycklar.



### 3.1.3 Avnämare och målgrupper

Inom NJK finns ett stort antal personer och grupper som har olika behov av ett välfungerande register. Nuläget är att alla de som skulle behöva ha tillgång till registret inte har det på grund av begränsningar i tillgängligheten och för att tekniska färdigheter saknas. Målgrupperna i korthet består av:

- kansliet
- junioravdelningen
- besiktningsnämnden
- kappseglingsavdelningen
- ”Match Race”-centret
- hamnarna.

För vissa målgrupper handlar det om flera personer, för andra om endast en person. Det är dock viktigt att även dessa räknas med som en grupp, då deras behov inte är mindre viktiga än de andras.

I inledningen definieras avnämare som de personer som har ett intresse att bevaka och är mottagaren för en produkt. För att gå lite djupare in på ämnet definierar Sommerville (2004 s 150) även tre typer av avnämare med olika perspektiv: interaktivt och indirekt perspektiv samt domänperspektiv. Det interaktiva perspektivet representerar de personer eller system som har en direkt växelverkan med systemet. Det indirekta perspektivet representerar de avnämare som inte själva använder systemet men påverkar kraven ändå. Domänperspektiv är igen de egenskaperna och begränsningarna som påverkar systemkraven. Inom registerprojektet gick det tydligt att identifiera de avnämare som var av typen interaktivt och indirekt perspektiv. Eftersom NJK har både en kommodor, verksamhetschef, styrelse och olika kommittéer fanns det många avnämare med det indirekta perspektivet. Resten av avnämarna hade ett interaktivt perspektiv då de endera arbetar med registret eller borde använda registret (men inte i nuläget kan använda registret på grund av begränsningar i tillgängligheten och att tekniska färdigheter saknas). Även en avnämare med domänperspektiv kunde identifieras då Finlands Seglarförbund

(hädanefter förkortat FSF) ställer krav på formatet för registeruppdateringar. Såsom Sommerville konstaterar ställer oftast dessa avnämare med olika perspektiv olika typer av krav, vilket stämde bra i NJK:s fall.

Avnämarna har olika behov för registret men också olika arbetssätt. En stor del av avnämarna sköter inom NJK uppgifter, som inte kräver närvaro i Björkholmens klubbutrymmen. Deras arbete skulle underlättas avsevärt ifall de hade tillgång per distans till registret. Ett praktiskt exempel är NJK:s besiktningsmän, som skall göra en registrering i båtregistret då de har besiktigt en NJK-registrerad båt. Om detta kunde göras per distans skulle det spara mycket arbetstid, då det finns omkring 800 båtar registrerade i NJK:s båtregister och största delen av dem besiktigas årligen.

## **3.2 Kravinsamlingen**

Wohlin (2005 s 100-101) nämner förändringsanalys som en metod att samla in krav med. Förändringsanalysen definierar Wohlin som att grundligt analysera det nuvarande läget, vad problemen består av och vilka funktioner som skulle behövas av ett framtida system. Inom registerprojektet har förändringsanalysen utförts. Den bestod av hela kravinsamlingens resultat och en undersökning av tillgängliga lösningar på marknaden. Undersökningen gjordes samtidigt som analysen för att få en realistisk bild av vad det finns för alternativ och om någon färdig produkt skulle uppfylla NJK:s krav.

Av de tekniker som nämns i kapitel 2.4 har en del använts inom registerprojektet. Intervjuer var den teknik som övervägande användes och därför också planerades grundligt. Bakgrundsforskning, dokumentgranskning, observation och brainstorming var tekniker som nog användes, men utan att man i högre grad varken noterade eller planerade vad man gjorde. De kändes som en helt naturlig del i processen och användes därför.

### **3.2.1 Bakgrundsforskning**

Själva kravinsamlingen började med en bakgrundsforskning, se kapitel 2.4.2, där material såsom databasrelationsschemat, funktioner i databashanteringsprogrammet, årsboken, instruktioner för fakturering, manualer för kansliarbete och övriga dokument ut-

nyttjades för att försöka få fram en heltäckande bild över vilka funktioner som redan existerar och vilka som borde utvecklas. Eftersom jag själv har arbetat i flera år som registeransvarig, hade jag själv från början en ganska klar uppfattning om vad det nuvarande registret klarar av och vad som saknas. Därför behövde funktionsutredningen inte utföras separat. Bakgrundsforskningen gav som resultat bl.a. följande problem:

- Registret har inte någon ändamålsenlig databasapplikation
- Registret är endast tillgängligt i Björkholmens interna nätverk
- Registret möter inte alla behov, externa listor för t.ex. hamnplatserna och skåputhyrningen upprätthålls parallellt
- Registret har gammal och oanvänd data på vissa ställen
- Registrets databasstruktur borde uppdateras
- Registret har låg användarvänlighet
- Det krävs långa instruktioner för att utföra t.ex. massfaktureringar och årsboksutdrag

Bakgrundsforskningen uppmärksammade även det faktum att registret används på olika sätt av olika personer. Detta väckte tanken att göra intervjuer för att få alla avnämares åsikter och önskemål inkluderade i processen.

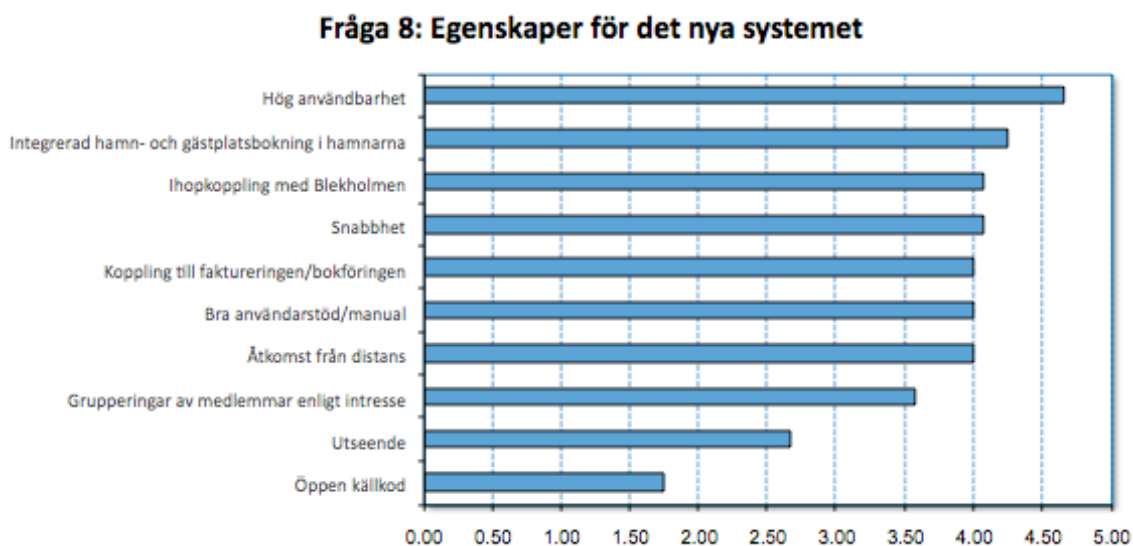
### **3.2.2 Intervjuer**

Intervjuer genomfördes med 12 avnämare av båda könen i åldern 19 – 64 år. Dessa intervjupersoner utvaldes på grund av deras roller inom NJK och representerade avnämare av både indirekta och interaktiva perspektiv. Intervjuerna genomfördes under en två veckors period och så gott som alla ägde rum i NJK:s klubbtrymmen på Björkholmen med ett undantag. Målet med intervjuerna var att få fram olika synvinklar på registret och vilka de olika avnämarnas behov var.

Själva intervjun var en blandning av en strukturerad och ostrukturerad intervju jämför kapitel 2.4.3. Intervjufrågorna förbereddes med omsorg med bakgrundsforskningens resultat som bas. Frågorna kan studeras i helhet i bilaga 1. I intervjun var alla frågor öpp-

na frågor förutom en, som var en rankingfråga (se kapitel 2.4.4) om egenskaper för det nya systemet. Alla frågor i intervjuerna behandlade problemet i fråga, inte kravhanteringsprocessen eller kravinsamlingen, jämför kapitel 2.4.3.

Fråga 1 hade som mål att få en översikt över användarens praktiska erfarenheter om det nuvarande registret. Med hjälp av denna fråga kunde man även klart skilja på de avnämare som hade ett indirekt eller interaktivt perspektiv. Fråga 2 var en följdfråga till den första frågan och lämnades obesvarad ifall svaret till den första frågan var positivt. Frågorna 3-5 var frågor om det nuvarande registret och hade som mål att få fram mera information om problemdomänen. Här frågades även efter både negativa och positiva egenskaper hos det nuvarande systemet för att få fram en mångsidig diskussion. Fråga 6 sökte nya funktioner för det nya systemet och tog även indirekt upp bristfällig funktionalitet i det nuvarande systemet. Fråga 7 koncentrerades i helhet till ett eget ämne som hade orsakat livlig diskussion tidigare bland avnämarna – skall medlemskåren själv kunna kontrollera sina egna medlemsuppgifter på webben. Fastän svaret kunde ha varit endast ja eller nej var det klokt att göra detta till en egen öppen fråga eftersom så gott som alla avnämare hade en åsikt i ämnet. Fråga 8 var en rankingfråga som hade som mål att få in avnämarnas åsikter om egenskaper för det kommande systemet. Följande resultat för frågan finns sammanställda i Figur 13.



Figur 13. Resultatsammanställning för fråga 8.

Analysen av materialet kan studeras i kapitel 3.3. Den sista frågan var en mycket öppen fråga och hade som mål att ta fram alla tankar och idéer avnämarna hade om sitt önske-

system. Där visade det sig dock att den var lite för vagt formulerad, eftersom det krävdes tilläggsfrågor. Detta visade att påståendet ”oftast fås inte mycket resultat genom att be intervjupersonen att berätta vad de vill ha, människor har lättare att diskutera om ämnet är definierat och inte flummigt” (se kapitel 2.4.3) stämde ganska bra.

Själva arbetssättet under intervjuerna följde ganska långt de riktlinjer som presenteras i kapitel 2.4.3. Eftersom jag personligen kände alla intervjupersonerna och hade en bra uppfattning om deras erfarenheter med systemet gick det lätt att justera språket så att en intervjuperson inte kände sig okunnig eller onödig. På grund av tidigare diskussioner och erfarenheter med intervjupersonerna hade jag även en god uppfattning om vad de ville ta upp i intervjun. Trots detta dök det i varje intervju upp minst ett överraskningsmoment, vilket visade att personliga intervjuer är en bra teknik för att ta fram tyst kunskap. Om man jämför med att dessa frågor kunde ha ställts alla avnämare samtidigt i en grupp skulle inte alla små detaljer som visade sig vara viktiga ha kommit fram.

Som dokumenteringsteknik i intervjuerna (jämför kapitel 2.4.3) användes endast anteckningar gjorda av intervjuaren och det fungerade bra.

### **3.2.3 Dokumentgranskning**

Dokumentgranskning skedde oplanerat när uppdateringsfilerna till FSF studerades. Risken med att blanda ihop det dokumenterade och verkliga systemet (se kapitel 2.4.5) undveks ganska bra genom goda kunskaper om systemet och en grundlig genomgång av funktionerna för uppdateringsfilerna.

### **3.2.4 Observation**

Observationer beskrevs i kapitel 2.4.6 och inom fallstudien användes de i form av oplanerade observationer. En hel del av dem skedde i vardagen helt naturligt då jag hjälpte andra anställda med registret och lade märke till hur de använde funktionerna i programmet och vilka problem de hade. En hel del brister framkom ur dessa observationer. Flera observationer skedde även under intervjutillfällen, då intervjupersonen hade lätta-

re att visa något i det verkliga systemet än att förklara det muntligt. Olika problem som t.ex. avsaknad av felkontroll för inmatad data lyftes starkt fram.

I NJK:s fall kan man nästan tala om etnografi då jag själv arbetat för föreningen i många år och på så sätt kunnat i praktiken observera de olika behoven och kraven.

### **3.2.5 Brainstorming**

Brainstormingsessionerna var spontana och oplanerade, skedde oftast på möten med stödgruppen. Ämnen som togs upp var bl.a. olika idéer om möjliga leverantörer eller lösningar.

### **3.2.6 Problemdomänen**

Problem som kom fram under kravinsamlingen var mängden tyst kunskap som olika nyckelpersoner besitter. NJK har en hel del regler och traditioner och dessa är dåligt dokumenterade. Ett exempel är hur medlemsnumret skall skapas. De tre första siffrorna motsvarar inskrivningsårtalet, de tre mittersta siffrorna medlemmens födelseår och de tre sista står för hur många medlemmar det året har skrivits in, exempelvis 206 186 017.

En mycket kort version på en nyttoanalys, som Sommerville (2004 s 144-146) beskriver som en studie som avgör ifall det är värt att fortsätta projektet eller inte, genomfördes även i registerprojektet. I NJK:s fall bestod detta av följande frågor och samtidigt de svar analysen kom fram till:

#### **Vad har medlemmarna för nytta av denna investering?**

- Får mer intressant information per e-post
- Får snabbare information per e-post och brev
- Får mer information som angår just den medlemmen per e-post
- Personligare service
- Påminnelserna av obetalda fakturor underlättas

- Mera korrekt årsbok och fakturor
- Faktureringsystem för t.ex. tillfälliga hamnplatser och produkter
- Bättre möjligheter för medlemmen att betala produkter, tjänster eller skulder i kansliet

### **Vad har NJK för nytta av denna investering?**

- Ifall faktureringen, reskontran och påminnelserna övertas av registret från bokföringsbyrån, sparar NJK en stor del av bokföringsbyråns årliga kostnad
- Bättre kontroll över reskontran och NJK kan effektivare skapa och kräva in fakturor
- Alla som använder systemet sparar tid då informationen finns på ett ställe
- Tidskonsumtionen för olika processer dras också ner
- Färre antal mänskliga fel i t.ex. fakturering

## **3.3 Analys**

Själva analysen skedde till en del samtidigt som intervjuerna men till största delen efter att materialet samlats in. Som metoder användes inga specifika, men likheter kan dras till problemdomänorienterad analys då mycket tankearbete koncentrerade sig på problemdomänen.

### **3.3.1 Domänkrav**

Som domänkrav i registerprojektet kunde det härledas från intervjuerna att utlokalisering av servern är en viktig aspekt eftersom detta medför säkerhet. Gällande servermiljön eller lösningssystemets arkitektur fanns inga krav som begränsade alternativen. Tvärtom, fältet hölls öppet för att inte begränsa en möjlig lösning. Dock var det en viktig egenskap att distansuppkoppling till registerprogrammet skulle vara möjlig. För att inte begränsa kandidaterna ville man inte begränsa själva lösningen på det kravet heller, men ett webbgränssnitt var den ideala lösningen.

Det visade sig att de olika kandidaterna erbjöd tre olika möjligheter till distansuppkoppling. Den allmännaste lösningen var ett webbgränssnitt utvecklat med olika språk och metoder. Sedan fanns det även en kandidat som erbjöd en distansapplikation, där man laddade ner en körbar exe-fil och sedan körde applikationen mot servern utan att installera den på klienten. Det tredje alternativet var inte direkt en distanslösning av leverantören, då enda sättet att få deras klientapplikation tillgänglig var att utnyttja ett virtuellt privat nätverk (VPN) inom NJK. Alla dessa lösningar var genomförbara i praktiken, men den som röstades fram för detta krav var ett webbgränssnitt.

Webbgränssnittet möjliggör användning av registret per distans oberoende av vilket operativsystem eller webbläsare användaren har. Fastan körningen av en applikation mot servern genom en körbar fil i princip ger samma frihet, begränsas detta alternativ då den körbara filen endast är tillgänglig i exe format och därför inte körbar på något annat operativsystem än MS Windows. Dessutom är man tvungen att ha rättigheter för att köra en applikation på datorn. Genom att endast utnyttja ett VPN har man nog tillgång till registret, men detta medför andra problem då det är fråga om applikationer som lokalt skall installeras på klienter. Uppdateringen och underhållet av dessa lokala installationer är mycket mer kostsam än t.ex. med ett webbgränssnitt, där man har mjukvaran på en server och underhållet sker en gång och är genast tillgängligt för alla användare.

### **3.3.2 Medlemsvy**

NJK har ungefär 2600 medlemmar och det finns ett behov av att knyta ihop data om dessa medlemmar. Under analysen blev det klart att medlemmarna behöver profileras bättre för att massutskicken skall kunna riktas bättre och för att lättare få tag på rätt grupper av medlemmar. Ett praktiskt exempel på problem med det nuvarande registret är medlemsansökningsblanketten, där en stor del intressen och erfarenheter efterfrågas, men informationen utnyttjas aldrig då den inte sparas i registret.

För varje medlem skall grunduppgifter såsom t.ex. medlemskategori, namn, adress, kontaktuppgifter, titel inskrivningsdatum och medlemsnummer sparas. Det finns för medlemmarna även fält där make/maka anges för att ett gift par skall kunna kopplas ihop. Ett eget datumfält behövs som innehåller när medlemmen har blivit eller skall bli stän-



dig medlem, eftersom denna information inte går att räkna ut med någon formel. I medlemsregistret sparas även information om medlemmen önskar höra till FSF eller Finlands Båtförbund (hädanefter förkortat BF). Detta är kopplat till uppdateringslistorna som går till förbunden. Specialfunktionen som kort beskrevs tidigare om betalare skall även ha ett eget fält. I nuvarande registret har detta lösts genom ett fält där man matar in id nummern på den medlem som skall vara fakturamottagare.

För att ha största nyttan av medlemsvyn skall följande fält vara kopplade till medlemmen och vara synliga på medlemskortet:

- alla obetalda fakturor
- senaste reskontratilläggen
- utskick
- postningslistor
- utmärkelsetecken
- pokaler
- kommittétilhörigheter
- aktivitet och intressen inom NJK
- kompetenser
- kurs- och festdeltagande
- ägda båtar inskrivna i NJK
- bokade hamnplatser (både vinter och sommar)
- hyrda skåp
- ”MRC team”-roll.

En stor del av de uppräknade fälten är inte nu integrerade i registret, men finns sparade i externa filer och system. Genom att centralisera denna information får NJK en större nytta av det nya registersystemet och arbetsbördan minskar då det räcker med ett system för att administrera dessa ärenden.

### 3.3.3 Båtvvy

Båtarna som registreras in i NJK skall hanteras av en båtvvy. En medlem kan ha flera båtar och en båt kan ha flera ägare, så förhållandet måste kunna gå i plural form i båda riktningarna. Detta stöds inte av det nuvarande registret, vilket orsakar problem då alla ägare inte enkelt fås fram vid faktureringar och kontaktbehov. Detta kom fram under bakgrundsforskningen då databasens struktur granskades.

För varje båt behövs grunduppgifter såsom båttyp, namn, segel- och registernummer, klass, längd, bredd, djup, vikt o.s.v. Vissa NJK-specifika fält behöver båtregistret även stöda såsom båtregisterklass (alternativ: A, B, C, D), förbund (alternativ: FSF, BF, SB, utländsk, annat), byggmaterial (alternativ: glasfiber, termoplast, aluminium, stål, trä, gummi, järn, betong, övrig). Sedan skall fält såsom klass, konstruktör och varv i grund och botten skall vara fritext-fält men de vanligaste texterna får gärna föreslås för att underlätta ifyllandet och för att undvika skrivfel. Besiktningar är de fält som uppdateras varje år för en båt och de skall innehålla årtal. Ifall en båt är utskriven kan det vara på grund av många olika orsaker, men det kom fram under intervjuerna att det inte räcker med endast ett ja eller nej alternativ för utskriven fältet, utan det behövs även ett årtal ihopkopplat med det. Från båtvyn vore det bra att få en översikt över följande uppgifter:

- Alla registrerade ägare
- Hamnplatser båten har och tidigare har haft (både vinter och sommar)
- Anmälningar till sjösättning, båtlyft och besiktningstillfällen
- Registernummer på möjliga trailers som används under vinter/sommarförvaring

### 3.3.4 Resursvy

Ett av de viktiga resultaten av analysen kombinerat med en undersökning av tillgängliga lösningar på marknaden var, hur begreppet resurshantering kunde tillämpas inom NJK:s register. NJK behöver hantera många olika resurstyper till exempel:

- uthyrning av utrymmen
- bokning av hamnplatser

- reserveringar av följbåtar
- träningstider för ”Match Race”-båtar
- vandringspokaler
- utlösta nycklar

En fullständig lista på dessa olika resurser finns i Bilaga 2. Dessa behov täcks i nuläget med olika listor, dokument och egna lösningar. I värsta fall är dessa behov helt odokumenterade. Detta orsakar många problem bl.a.:

- Redundans och otillförlitlighet. Flera dokument innehåller samma information och det inte är bekant vilket som är det rätta eller uppdaterade. T.ex. finns det två versioner av hamnplatsbokningarna i bruk för samma hamn.
- Onödigt manuellt arbete. Olika system används för både bokning, medlemskap och fakturering. T.ex. då träningstiderna för ”Match Race”-båtarna skall faktureras och träningstiderna inte är kopplade till medlemsregistret som i sin tur inte är kopplat till faktureringen. Detta resulterar i tre skilda listor som skall manuellt jämföras med varandra. Ifall det sker en förändring i den första måste även de övriga listorna även granskas och ändras.
- Tyst kunskap. Det finns inte stöd av systemet vid olika processer, utan de är beroende av personalens egna kunskaper och erfarenheter. T.ex. då en medlem vill skriva ut sig händer det lätt att ingen kommer ihåg att begära returnering av möjlig hamnyckeln, eftersom nyckelregistret endast finns i pappersformat.

Tanken på ett resurshanteringsystem utvecklades från en undersökning av tillgängliga lösningar på marknaden, då det visade sig att en del av marknads registerprodukter har en sådan del integrerad. Denna lösning skulle då kunna täcka en stor del av NJK:s krav och koncentrera många ströfunktioner till ett system. Flera resurshanteringsystem innehåller ofta en grafisk kalender som visar hur alla resurser är i användning och av vem. Denna kalender skulle då ge NJK en bra översikt över hur alla resurser är i användning och ersätta de enskilda kalendrarna som nu underhålls på olika håll. T.ex. kansliet, junioravdelningen och kappseglingsavdelningen har egna kalendrar som inte är synkroniserade.

### **3.3.5 Funktioner och tjänster**

I kapitel 2.4.1, som handlar om strukturerad analys, nämndes som metod processspecifikationer (se sid 34). Denna tanke utvecklades till beskrivning av tjänster under analyskedet då det fanns ett behov av att synliggöra vilka funktioner och tjänster som sköts inom registret. Dessa olika tjänster skall skötas med olika tidsintervall på daglig, månatlig och årlig basis. Till dessa tjänster kan räknas olika massfaktureringar men även små dagliga tjänster och funktioner såsom adressförändringar och informationsökningar. Alla funktioner som dokumenterades i registerprojektet kan studeras i bilaga 3. Detta dokument gav under offertrundan kandidaterna en realistisk inblick i hur registret används inom NJK och lyfte fram viktiga detaljer såsom t.ex. basen för vinterplatsfaktureringen.

## **3.4 Specifikationen**

Eftersom målet från projektets start var att försöka hitta en färdig produkt och möjligen modifiera den för att passa NJK:s behov bättre, utfördes aldrig någon specifikation d.v.s. systemkraven togs inte fram i detta skede.

## **3.5 Offertrundan**

Under analyskedet påbörjades en undersökning av tillgängliga lösningar på marknaden, dels för att stöda analysen, dels för att leta fram en möjlig leverantör för det nya systemet. Som utgångsläge fanns inga begränsningar gällande teknik eller plattform och resultaten från intervjuerna visade att avnämarna inte lade någon större vikt på om systemet skulle vara baserat på öppen källkod eller inte. Mål för marknadsundersökningen var att hitta en leverantör eller produkt som skulle uppfylla kraven och om inte alla krav så åtminstone de mest kritiska. I undersökningen utnyttjades även en e-postförfrågning till de 50 största seglingsföreningarna i Finland och de ledande föreningarna i Norden. Som resultat av förfrågningen kom det fram att majoriteten av de andra föreningarna har liknande problem och har inte hittat någon bra lösning på dem. En del hade på samma sätt som NJK något självutvecklat databashanteringsprogram i bruk, och andra hade ta-

git i bruk kandidat D men inte varit nöjda. Via dessa förfrågningar kom även tipset om kandidat G, som visade sig vara den mest överlägsna kandidaten.

	A	B	C	D	E	F	G	H	I
Medlemsregister	ja	ja	ja	ja	ja	ja	ja	ja	ja
Familjer				ja	ja		ja	ja	ja
Kommitté			ja		ja		ja	ja	
Fria fält		ja	ja		ja	ja	ja	ja	ja
Båtregister		ja		ja	ja		ja	ja	
Hög användbarhet		ja		nja	nja	nja	ja	ja	ja
Sökning/listor					ja	ja	ja	ja	
Resurshantering		ja			ja		ja	ja	
Produktregister		ja			ja		ja	ja	
Fakturering	ja		ja*	ja	ja		ja	ja	*
Reskontra				ja	ja		ja	ja	
Påminnelser				ja	ja		ja	ja	
Tillgång per distans			ja	ja		ja	ja	nja	ja
Historia					ja		ja	ja	
Användarrättigheter	ja		ja	nja		ja	ja	ja	
Kurser&Evenemang			ja		ja		ja		
Webbanm.					ja				
E-fakturor	ja			ja			ja	ja	
Dokument bank						ja			
Sync. Med Office		ja			ja	ja	ja	ja	
Newsletters						ja	ja		
Kassafunktion	ja								
Företagets ekon.					dålig		bra	bra	
Språk	finska	svenska	norska	finska	finska	svenska	svenska	finska	finska

**Pris:** mycket gratis 30€/anv/mån 19€/anv/mån 1€/medl/år mycket mycket

Figur 14. Kandidatjämförelse vid offertrundan.

I Figur 14 kan de olika kandidaterna och hur de möter kraven studeras. Figuren användes som grund vid valet av leverantör och som stöd under offertrundan för att jämföra kandidaterna. I figuren är kraven till vänster i stort sett sorterade i viktighetsordning, men eftersom valet baseras på kompromisser kan man inte följa dem i ordning. Som man kan se i figuren har många kandidater inte ens den obligatoriska möjligheten till ett båtregister, och de eliminerades snabbt ur offertrundan. Till slut stod avgörandet mellan kandidat E, G och H. Offerten från kandidat E avböjdes till slut på grund av att leverantörens omsättning var för liten och användbarheten var låg i programmet. Offerten från kandidat H avböjdes som sista på grund av att lösningen baserade sig på en distanslösning i form av en klientapplikation som körs mot servern och det höga priset på offerten. Som vinnande offert valdes alltså kandidat G som har ett FLEX-baserat webbgränssnitt och produkten utvecklades med NJK:s krav som stöd.

### **3.6 Sammanfattning**

Med hjälp av en djup förståelse av det nuvarande systemet och olika kravinsamlingsmetoder har en stor mängd information kunnat samlas in och analyseras. Processen var givande och väl värd sin tid, då varje intervju gav en ny synvinkel eller någon information som inte ännu hade dokumenterats. I NJK:s fall var det även många olika målgrupper och avnämare som använder registersystemet och deras behov var viktiga att uppmärksammas.

Själva analysen skedde delvis genom hela processen men till största del efter att intervjuerna hade utförts. Själva intervjuanteckningarna var till stor nytta för att försäkra sig om att inte någon synvinkel hade översetts.

Offertrundan var en fråga om kompromisser då inga kandidater täckte alla NJK:s behov och önskemål. Som vinnande offert valdes en lösning med hög användarvänlighet och bra tillgänglighet.

## 4 SLUTSATSER

Fallstudien utnyttjade inte någon viss dokumenterad metod för att samla in och analysera kraven, men många kravinsamlingstekniker användes och vissa drag av problemdomänorienterad analys kunde även identifieras. Eftersom fallstudien utfördes innan jag hade djupare kunskap om dessa tekniker och analysmetoder, kan man dra slutsatsen att en stor del av dem är lätta att närma sig och naturliga i en kravhanteringsprocess av fallstudiens typ. En stor fördel hade jag också av min arbetserfarenhet på NJK. Såsom i kapitel 3.2.4 nämndes, kan man nästan kategorisera mitt deltagande i kravinsamlingen som etnografi, eftersom detta projekt har varit i mina tankar som ett möjligt examensarbete sedan jag började på NJK.

Det intressanta var att i efterhand lägga märke till hur många olika metoder och tekniker som hade utnyttjats inom kravhanteringen utan djupare kännedom om dem. En del tekniker genomfördes planerat och andra utvecklade sig själva och kändes som en naturlig del av processen. Intervjuerna var omsorgsfullt planerade, men de andra teknikerna såsom bakgrundsforskning, dokumentgranskning, observationer och brainstorming användes utan att känna till noggrannare detaljer om hur de skall utföras. De gav ändå goda resultat. Man kan konstatera att de är lätta att ta till sig och naturliga i en kravinsamlingsprocess.

Problemorienterad analys lägger stor vikt på beskrivningen av problemdomänen och delar klart upp beskrivningen i två delar, kraven och specifikationen. Själva kraven består av vad systemet skall ha för funktionalitet och specifikationen ger en noggrannare beskrivning över hur det skall ske. Inom PDOA används naturligt språk som den huvudsakliga dokumenteringsmetoden. Jag anser att på grund av den tydliga skillnaden mellan krav och specifikation och även betoningen av naturligt språk var PDOA den metod som analysen i fallstudien påminde om. Till skillnad från strukturerad analys och objektorienterad analys, som baserar sig tungt på kunskap om modellering och objektorienterat tänkande, har PDOA ett mera logiskt sätt att koncentrera sig på problemdomänen och dokumentera de funktionaliteterna som behövs.

Undersökningen av möjliga färdiga lösningar i fallstudien löpte smidigt, dock var det problem i början då jag inte kände till de olika termer som det tänkta lösningssystemet kunde beskrivas med. Sökord som visade sig vara nyttiga för att hitta en möjlig leverantör var bl.a. CMS (Content Management System), registerhantering, resurshantering och föreningsregister. Det eftersökta egenskaperna är ju en viktig del inom CMS-programvara, men de flesta program av den typen hade dock begränsningar som inte tillät ett register på båtarna.

Målet med examensarbetet var att besvara några utvalda problemformuleringar.

### **Varför är kravhanteringen en kritisk del för ett lyckat IT-projekt?**

Kravhanterings betydelse lyftes fram redan i inledningen då ett antal exempel på misslyckade IT-projekt nämndes. I litteraturen framhävs även vikten av en välgjord kravspecifikation och den kan jämföras med grunden till ett hus. Självklart finns det projekt där framgången inte är beroende av kravhanteringen, men i en stor del av fallen visade statistiken att kravhantering är en viktig del för ett lyckat IT-projekt.

Inom fallstudien kunde detta konstateras i praktiken, eftersom om inte en vid kravsamling och analys hade genomförts, skulle inte kraven ha sett ut som de nu är formulerade. Problemet med NJK var att det fanns så många avnämare med olika behov och önskemål. Med hjälp av dokumentering av dessa krav kunde de samlas ihop och prioriteras. Självklart gjordes det ett försök att ta med alla avnämares önskemål, men med tanke på NJK:s verksamhet kunde man relativt bra rangordna kraven och skära bort dem som sannolikt skulle ha orsakat orimligt höga kostnader. Ett exempel på detta var en avnämares önskemål om att då telefonen ringer skall registersystemet känna igen nummeret och automatiskt hämta upp medlemmens information på skärmen.

### **Hur försäkra sig om att kunden får vad den vill ha?**

Som de olika metoderna framhäver, är det viktigt att kunden förstår och kan godkänna kravdokumentet. I grund och botten är det kravinsamlingen som skall försäkra att rätt information samlas in. Inom intervjuerna observerades detaljer som att använda ett språk som intervjupersonen förstår och kunna läsa situationer som viktiga faktorer. Jag tror att ett bra sätt att försäkra sig om en nöjd kund är att från början göra en ansträng-



ning för att verkligen förstå vad kunden kan vara ute efter. Som tidigare diskuterades i kapitel 2.1.2 har kunden alltid den bästa domänkunskapen men nödvändigtvis inte de tekniska färdigheterna för att kunna uttrycka sig i ord som leverantören normalt använder. I fallstudien kom det tydligt fram då jag kände till alla olika resurser och tjänster som NJK har ett behov av att hålla reda på och administrera, men innan jag hittade ordet resurs hade jag väldigt svårt att förklara detta funktionsbehov. Detta visar att det kan vara fråga om detaljer såsom termer som kan vara avgörande för att kundens krav skall bli rätt definierade.

Kommunikationen mellan kunden och leverantören skall helst vara smidig och det är även viktigt att validera krav för att försäkra sig om att de rätta behoven är dokumenterade och förstådda.

### **Vilka metoder lämpar det sig att utnyttja inom kravhanteringen?**

I teorikapitlet presenterades olika metoder för att genomföra kravhanteringen, men som det diskuterades i inledningen, är valet av metod beroende av många faktorer. Om man talar om kravinsamlingsmetoder är det lättare att rekommendera tekniker. Om man ser på Figur 7 i kapitel 2.4.1 kan man konstatera att valet av teknik är mycket långt projektspecifikt och bör övervägas på basen av tillgängligt material, resurser och källor. När det kommer till valet av analysmetod är det till en stor del beroende på leverantörens kunskap och erfarenhet om ämnet.

Jag ser inte heller något fel i att inte begränsa sig till någon viss metod utan att utnyttja delar av dem tillsammans ifall projektet lämpar sig för det. I fallstudien användes sunt bondförnuft, logik och all analys baserade sig på kravinsamlingens resultat och slutsultatet påminde om problemdomänorienterade analysens riktlinjer om man bortser från problemramar. I huvudsak tror jag att kravinsamlingen är kritisk eftersom den ger materialet som analysen grundar sig på. Det är antagligen dock projektspecifikt hurdan material kravinsamlingen producerar.

### **Hur löstes fallstudien jämfört med teorin och vad kunde ha gjorts bättre?**

Fallstudien löstes med ett flertal olika kravinsamlingstekniker och med en analys med drag av problemdomänorienterad analys. I detta fall blev resultatet bra, och kandidater-

na som läste kravdokumenten var nöjda och tyckte att de fick en bra överblick av behoven. Det som skulle ha kunnat förbättras var detaljnivån i kraven och dokumenteringsformen. I efterhand då jag studerade de olika analysmetoderna och deras rekommendationer om hur kraven skall dokumenteras märker jag att fallstudien skulle ha blivit bättre med hjälp av ett mera strukturerat kravdokument. Metodmässigt kan jag inte säga huruvida slutresultatet skulle ha blivit mycket bättre om någon viss analysmetod skulle ha följts.

## KÄLLOR

Brainstorming. 2009, Wikipedia, publicerad 5.9.2009.

Tillgänglig: <http://sv.wikipedia.org/wiki/Brainstorming> Hämtad 6.4.2010.

Bray, Ian K. 2002, *An introduction to requirements engineering*, Storbritannien: Pearson Education Limited, 408 s.

Eliasson Annika. 2002, *Fallstudier*, Malmö högskola – Teknik och samhälle, publicerad 9.4.2002.

Tillgänglig: [http://www.ts.mah.se/utbild/ck2340/Delkurs\\_3/Fallstudie.htm](http://www.ts.mah.se/utbild/ck2340/Delkurs_3/Fallstudie.htm) Hämtad 17.4.2010.

Glass, Robert. 1998, *Software Runaways*, Harlow: Prentice Hall, 288 s.

Jackson, Michael. 1995, *Requirements & Specifications: A Lexicon of Software Practice, Principles and Prejudices*, Storbritannien: Pearson Education Limited, 228 s.

Kovitz Benjamin. 1999, *Practical Software Requirements; A Manual of Content and Style*, Greenwich: Manning, 454 s.

Lindegren, Håkan. 2003, *Programvaruprojekt – Stabilitet, användbarhet och inkrementell utveckling*, Sverige: Studentlitteratur.

Maciaszek, Leszek A. 2001, *Requirements Analysis and System Design*, Storbritannien: Pearson Education Limited, 378 s.

Ordboken, 1994, *Engelsk-svenska/svensk-engelska ordboken*, Norge: Norsteds Förlag.

Requirements analysis. 2010, Wikipedia, publicerad 25.2.2010.

Tillgänglig: [http://en.wikipedia.org/wiki/Requirements\\_analysis](http://en.wikipedia.org/wiki/Requirements_analysis) Hämtad 15.3.2010.

Standish Group. 1995, *Chaos Report*.

Structured analysis. 2010, Wikipedia, publicerad 31.3.2010.

Tillgänglig: [http://en.wikipedia.org/wiki/Structured\\_analysis](http://en.wikipedia.org/wiki/Structured_analysis) Hämtad 14.4.2010.

Whitten, J.L. & Bentley, L.D. 1998, *Systems Analysis and Design Methods*, Irwin McGraw-hill, 724s.

Wohlin, Claes. 2005, *Introduktion till programvaruutveckling*, Sverige: Studentlitteratur.

## **BILAGOR**

Bilaga 1. Intervjudokumentet för fallstudien

Bilaga 2. Användarkraven för fallstudien

Bilaga 3. Processbeskrivningen för fallstudien



---

## Nyländska Jaktklubben

### **Intervju om Nyländska Jaktklubbens nya register**

Datum: Plats:

Ålder: Kön:

1. Hur länge eller hur mycket har du jobbat med NJK:s nuvarande register i MS Access?
2. Ifall du inte bekantat dig med registret, vad har du för uppfattning om det?
3. Vad har du upplevt som brister i det nuvarande Access-baserade registret?
4. Vilka specifika uppgifter är svåra/jobbiga/omöjliga att själv genomföra nu?
5. Vad har du upplevt som positiva egenskaper/funktioner hos det nuvarande Access-baserade registret som gärna får hänga med till nästa version?
6. Vad finns det för uppgifter som du (lättare) skulle vilja kunna utföra i det nya registret?
7. Tycker du att det är viktigt att medlemmar själva skall via NJK:s hemsidor komma åt och se och ändra sina egna uppgifter i registret?



---

## Nyländska Jaktklubben

### 8. Ge värdet 1 - 5 för hur viktiga nedanstående egenskaper är:

(då 1 motsvarar inte alls viktigt och 5 mycket viktigt)

- 1 2 3 4 5 Utseende
- 1 2 3 4 5 Lättanvändbart
- 1 2 3 4 5 Åtkomst hemifrån
- 1 2 3 4 5 Snabbhet
- 1 2 3 4 5 Bra användarstöd/manual
- 1 2 3 4 5 Koppling till faktureringen/bokföringen
- 1 2 3 4 5 Noggrannare grupperingar av medlemmar enligt intresse
- 1 2 3 4 5 Ihopkoppling med Blekholmen
- 1 2 3 4 5 Integrerad hamnplats och gästplats bokning för BJ & BL
- 1 2 3 4 5 Opensource-programvara

### 9. Hur skulle ditt drömregistersystem för NJK se ut?

Tusen tack för ditt intresse och din samarbetsvilja!

## Nyländska Jaktklubbens nya register

Utredning av projektets egenskaper, behov och detaljer

Nyländska Jaktklubbens nya register.....	1
Omgivning .....	2
Krav från förra versionen.....	2
Vad har medlemmarna för nytta av denna investering?.....	2
Vad har NJK för nytta av denna investering?.....	2
Delar/moduler .....	2
Ekonomi .....	2
Webb.....	3
Produkt och resurshantering .....	3
Extern kommunikation .....	5
Databasapplikation .....	5
Krav .....	5
Obligatoriska krav .....	5
Mindre obligatoriska krav.....	6



## **Omgivning**

- Utlokaliserad server för att öka säkerheten
- Inget annat krav på mjukvaran, med tanke på märke och modell

## **Krav från förra versionen**

- Medlemskategorier
- Familjefakturor

## **Vad har medlemmarna för nytta av denna investering?**

- Får mer intressant information per e-post
- Får snabbare information per e-post och brev
- Får mer information som angår just den medlemmen per e-post
- Personligare service
- Påminnelserna av obetalda fakturor underlättas
- Mera korrekt årsbok och fakturor
- Faktureringsystem för t.ex. tillfälliga hamnplatser och produkter
- Bättre möjligheter för medlemmen att betala produkter, tjänster eller skulder i kansliet

## **Vad har NJK för nytta av denna investering?**

- Ifall faktureringen, reskontran och påminnelserna tas över från bokföringsbyrån till oss, sparar NJK en stor del av bokföringsbyråns årliga kostnad
- Bättre kontroll över reskontran och NJK kan effektivare skapa och kräva in fakturor
- Alla som använder systemet sparar tid då informationen finns på ett ställe
- Tidskonsumtionen för olika processer dras också ner
- Färre antal mänskliga fel i t.ex. fakturering

## **Delar/moduler**

### **Ekonomi**

- Kontakt till bokföringsbyrån

- Läsa in en del ekonomi-information om medlemmarna från bokföringsbyrån eller från bankfiler
- Visa senaste faktureringar/transaktioner på medlemskortet
- Visa ALLA obetalda räkningar på medlemskortet
- NJK skall kunna skapa fakturor på basis av våra behov och produkter, enkelt och smidigt. Även massfakturor skall kunna skapas hos NJK
- Skickar över fakturainformation för uppföljning och bokföring till bokföringsbyrån
- Uppdateras tillräckligt ofta så att NJK har pålitlig ekonomisk information
- D.v.s. kommunikation mellan NJK och bokföringsbyrån och NJK tar över faktureringsfunktionen från bokföringsbyrån

## Webb

- Skall finnas teknisk färdighet att i efterskott ta i bruk webbdimensionen, t.ex. att medlemmar själva kan logga in till [www.njk.fi](http://www.njk.fi) (Joomla! 1.5) eller någon annan webbmodul för att kontrollera och till en del uppdatera sin profil
- Anmälningen till evenemang och kurser vore bra att kopplas ihop med registret

## Produkt och resurshantering

- Alla produkter och tjänster hanteras i ett produkt- och resurshanteringsverktyg
- För att underlätta fakturaskapandet, inventariet, bokföring och tidshantering
- **Resurser** är t.ex.
  - Stora salen
  - Styrelserummet
  - Hamnplatserna (både för säsongen sålda och tillfälliga hamnplatser)
  - Vinterplatser
  - Följebåtarna
  - Klubbens utrustning (datorer, betalterminaler, flaggor, pistoler, mötesutrustning m.m.)
  - MRC-båtars träningstider
  - Pokaler
  - Pris
  - Skåp
  - Nycklar
  - Nattvaktsturer

- **Produkter** är t.ex.
  - NJK-produkter
  - Interimsvimpel
  - Nyckelpant
  - Skåphyra
  - Alla avgifter (medlems, båtregister, hamnplatsavgifter och så vidare)
  - Donationer till fonder
  - Festavgifter
  - Kursers deltagaravgifter
  - Tillfälliga försäljningar, t.ex. aktuella försäljningsobjekt
- **Tidsperioder** som modulen måste ta i beaktande:
  - Tills returneras (nycklar och skåp)
  - Per år (pokaler och pris)
  - Per säsong (hamnplatser)
  - Per dag (utrustning, utrymmen, fölgebåtar, tillfälliga hamnplatser)
  - Per timme (fölgebåtar, MRC båtars träningstider, tillfälliga hamnplatser, utrustning)
- **Nyttan** med detta system:
  - En gemensam kalender som skulle vara lättåtkomlig och där man direkt skulle kunna se hur alla NJK:s resurser är i bruk och när
  - Inventariet på allt NJK skall hålla reda på sköts automatiskt av systemet
  - Pålitligare bokföring över vilka t.ex. fölgebåtar används av vilken kommitté och hur många produkter NJK har sålt till vilka priser
  - En enklare och pålitligare grund till fakturor. Scenario: en medlem kommer in och köper en keps, en interimsvimpel, betalar av sin 2 dagars tillfälliga hamnplats och vill samtidigt donera 20 € till juniorfonden. Dessa alla har olika summor, kostnadsställen, kontoföringsställen och moms och alv:er. Ifall all den informationen finns kopplad till en produkt så blir fakturan automatiskt rätt och allt bokförs korrekt utan att göra manuellt arbete däremellan
  - Tar bort tidskrävande handarbete för kansliet och dubbla listor i diverse format

## Extern kommunikation

- Systemet måste kunna kommunicera med andra system, såsom bokföringsbyrå och Finlands Seglarförbund
- Till FSF måste NJK uppdatera medlems- och båtregistret, för att postningar och faktureringar skall gå rätt
- Till bokföringsbyrå behövs en kontakt för att NJK skall kunna se reskontran och göra faktureringar och så att de kan läsa in de fakturor NJK gjort för uppföljning
- För framtida behov vore det bra ifall systemet hade en färdighet att kommunicera med [www.njk.fi](http://www.njk.fi), som är uppbyggt på Joomla! och har en MySQL-databas. Nyttan med detta skulle vara att medlemmarna kan logga in på webbsidan och kontrollera och korrigera sina uppgifter och direkt anmäla sig och/eller sin båt till fester, kurser, evenemang, båtlyft och hamnplats

## Databasapplikation

- Databasapplikationen och användbarheten är en mycket viktig del som inte får underskattas
- Systemet skall kunna tjäna alla målgruppers syften utan att kräva en lång inlärningsperiod men dock måste en kompromiss göras med effektiviteten
- Det skall gå att använda systemet från distans genom en webbläsare eller med VPN-kontakt (Virtual Private Network) till servern. Någon form av distanskontakt är obligatorisk.

## Krav

Med tanke på detta projekts behov av ett registerhanteringsprogram, så kan situationen kräva kompromisser, men detta är den preliminära ordningen på kraven för ett nytt registerhanteringsprogram.

Dessa krav är generella och mera detaljer hittas i bifogade dokument om krav på mera detaljerad nivå.

## Obligatoriska krav

- Medlemsregister med bl.a. kategorier, sambon och betalare. Fria fält är viktiga
- Familjefakturering och -postning
- Båtregister med flere ägare

- Resursvy med kalender
- Produktregister
- Fakturering
- Historia sparad
- Lätt uppgjorda listor och sökningar
- Tillgång till registret per distans
- Användargrupper med rättighetsbegränsningar

### **Mindre obligatoriska krav**

- Webbanmälning till kurser och evenemang
- E-fakturor
- Dokument bank
- Synkronisering med "MS Office"-program
- Newsletters

## **Nyländska Jaktklubbens viktigaste processer**

Målet med detta dokument är att klargöra vilka processer som är viktiga för NJK:s verksamhet och skall med andra ord tas i beaktande i registerprojektet.

### ***Dagliga processer***

#### **Medlemsservice**

Olika uppgifter är

- Adressförändring
- Ändring och uppdatering av kontaktuppgifter
- Fakturautredningar vid problemsituationer

#### **Kommittéstöd**

Olika uppgifter är

- Skriva ut brev
- Betala ersättningar
- Anmälningar till evenemang

#### **Försäljning**

Kansliet har en liten försäljningsverksamhet där NJK-produkter säljs till medlemmar. Intäkterna går till NJK Service Ab. En skild mjukvara hanterar betalningar per kort och kassabokföringen görs för tillfället för hand.

#### **Födelsedagskort**

NJK och kommodoren skickar varje månad ut födelsedagskort till alla medlemmar som fyller jämna år, men minimum 50 år. Kansliet sköter de praktiska arrangemangen.

#### **Steg**

- Räkna ihop alla som under inkommande månad fyller jämt med en förfrågan
- Skriva ut adresslappar
- Skriva födelsedagskort för hand
- Klistra ihop, frankera och posta

## ***Månatliga processer***

### **Adresser till NJK-nytt**

Varje gång NJK-nytt skall postas körs alla mottagare ut från registret och adressuppgifterna skickas till tryckeriet.

### **Uppdatering till FSF/BF register**

Uppdateringen sker på grund av två orsaker; förbunden skall kunna göra en korrekt medlemsfakturering för oss och våra medlemmar har rätt att få en tidning, Nautic och dessa adressuppgifter måste även då uppdateras. För tillfället sker denna uppdatering till FSF/BF några gånger per år manuellt, men om den automatiseras vore det bra ifall det skulle ske månatligen.

## ***Årliga processer***

### **Medlemsfakturering**

Medlemsfaktureringen sker en gång om året, fakturorna skickas ut genast i början av året. Fakturorna baserar sig på en huvudbetalare per familj.

#### **Steg**

- Uppdatera medlemskategorier
- Uppdatera årsavgifterna, ifall det skett en förändring i dem på senaste höstmöte
- Kolla ifall någon som har fribrev skall komma med i faktureringen igen i år
- Generera inhemska medlemsfakturor
- Generera utländska medlemsfakturor
- Det skall gå en faktura per familj, den som är märkt som betalare skall vara mottagare och resten av familjen som tilläggsrader.

### **Båtregisterfakturering**

Alla båtar (ca 800) i NJK:s register betalar en årlig avgift beroende på båttyp, längd och användning. Fakturorna skickas ut på våren varje år.

#### **Steg**

- Dubbelkolla att alla båtar är i rätt kategori (beroende på typ, längd och användning)
- Generera båtregisterfakturor

## **Sommarplatsfakturering**

Ca 350 hamnplatser faktureras under våren. Hamnplatserna är uppdelade för Björkholmen och Blekholmen.

### **Steg**

- Samla ihop all information från respektive hamnmästare
- Hamnavgifter räknas ut på basis av båtens bredd, hamnplats och vissa platser har specialrabatt eller fast pris. Även trailerplatserna skall räknas med

## **Vinterplatsfakturering**

Vinterplatsfaktureringen sker genast efter sista båtlyftet sent på hösten och i fakturan ingår höstlyft, vinterplatshyra och sjösättning på våren. Formeln för att räkna ut summan är  $2 * vikt * 10,5€ + vinterlängd * vinterbredd * 12 €$ . Sedan kan det ännu tillkomma en extra avgift på 50 € per mastlyft med kran.

### **Steg**

- Alla lyfta båtars information samlas ihop
- Fakturorna skapas med hjälp av formeln

## **Årsboksutdrag**

Inför varje årsbok skall det tas ut information om medlemmarna, deras utmärkelsetecken, kommittétillhörighet och båtar. Data skall formateras till ett bestämt format och många olika sorteringar görs i processen för att få det rätt. För att utdraget skall bli rätt måste de olika kommittétillhörigheterna och titlarna ha ett eget bestämt sorteringsvärde.

## **Uppdatering av förtjänsttecken**

Det delas ut ca 10-20 förtjänsttecken per år på höst- och vårmötet. Alla förtjänsttecken skall matas in i registret till rätt medlem. Uppgifter som behövs är vilken typ av tecken och årtal givet.

Specialregler finns t.ex. den fjärde gången man beviljas ett seglartecken blir det automatiskt ett stort seglartecken.