

# Betydelsen av användarmedverkan i systemdesign: En fallstudie

Kandidatuppsats, 10 poäng, inom Systemvetenskapliga programmet

<i>Framlagd:</i>	Juni 2007
<i>Författare:</i>	Danisa Gomez Sagredo Vín Thorsteinsdóttir
<i>Handledare:</i>	Hans Lundin
<i>Examinatorer:</i>	Anders Svensson Lars Fernebro

1	INLEDNING .....	2
1.1	Syfte .....	3
1.2	Frågeställning .....	4
1.3	Avgränsningar .....	4
2	METOD.....	5
2.1	Tillvägagångssätt.....	5
2.2	Metodval.....	5
2.3	Fallstudier.....	6
2.4	Observationer .....	6
2.5	Intervjuer .....	7
2.6	Studiens validitet, reliabilitet och generaliserbarhet .....	8
3	TEORETISK REFERENSRAM .....	10
3.1	Definitioner.....	10
3.2	Användbarhet.....	10
3.2.1	Definition .....	10
3.2.2	Nyttan med användbarhet.....	14
3.2.3	Olika nivåer i användbarhetsarbete .....	16
3.3	Mätning av användbarhet .....	17
3.3.1	Användbarhetsinspektion .....	17
3.3.2	Riktlinjer för användbarhetsheuristiktest .....	17
3.3.3	Riktlinjer för användbarhetsheuristiktest, utvecklat av Nielsen (1993).....	18
3.3.4	Användbarhetstester .....	19
3.4	Användarmedverkan .....	20
3.4.1	Definition .....	20
3.4.2	Användarmedverkan i systemdesign.....	20
3.4.3	Maktförhållanden .....	21
3.4.4	Kommunikation.....	22
3.4.5	Motivation .....	23
3.5	Användarcentrerad design.....	24
3.5.1	Användbarhetsdesign .....	24
4	EMPIRI –BLODACCESSREGISTER .....	27
4.1	Bakgrund.....	27
4.2	Minivärlden .....	27
4.3	Utvecklingsverktyg .....	29
4.4	Utvecklingsmetod.....	29
4.5	Projektgruppen.....	30
4.6	Kravanalysen .....	31
4.6.1	Användaranalys .....	31
4.6.2	Mål 1: Statistik om fistelöverlevnad .....	31
4.6.3	Mål 2: Användbarhet i verksamheten .....	32
4.6.4	Mål 3: Användbarhet i organisation av verksamheten.....	32
4.6.5	Säkerhet.....	32
4.6.6	Uppgiftsanalys.....	33
4.6.7	Arbetet med kravanalysen.....	33
4.7	Designprocessen.....	34
4.7.1	Databasen .....	34
4.7.2	Design av gränssnitt .....	36
4.8	Evaluering av användbarhet .....	44
4.9	Projektets slut .....	45

5	DISKUSSION .....	47
5.1	Missförstånd .....	47
5.2	Samarbetsproblem .....	48
5.3	Motivationsbrist .....	49
5.4	Nyttan av användarmedverkan .....	49
5.5	Lärdomar .....	50
6	SLUTSATSER .....	53
7	FRAMTIDA FORSKNING .....	55
8	REFERENSER .....	56
9	BILAGOR .....	58
9.1	Observationer .....	58
9.1.1	Observation 1 .....	58
9.2	Enkät: .....	60
9.2.1	Frågor om systemet .....	60
9.2.2	Frågor om metoden .....	61
9.2.3	Frågor om kommunikation .....	62

**ABSTRAKT:** I denna uppsats tänker vi analysera ett systemutvecklingsfall där en användare hade väldigt stor roll i utvecklingsarbetet. Syftet med studien är att bidra till ökade kunskaper om vilka faktorer påverkar nyttan av användarmedverkan i systemdesign. Vi gjorde en fallstudie om ett projekt där författarna hjälpte en läkare att utveckla en applikation för att hålla reda på blodaccesser hos dialyspatienter enligt en användarcentrerad metod. Det visade sig att användarmedverkan gick mycket bra och var avgörande för kravspecifikationen och design av databasen men design av gränssnitt gick mycket sämre. Analysen visade att problemen uppstod främst när gränssnittsdesignen utfördes utan användaren. De faktorer som mest påverkade nyttan av användarmedverkan var utvecklingsmetoden, utvecklarnas motivation och kommunikationsfärdigheter samt maktfördelningen inom projektgruppen.

*Nyckelord:* användarmedverkan, fallstudie, användbarhetsdesign, blodaccessregister

# 1 Inledning

'It is intuitively appealing that user participation leads to system success. However, the empirical evidence on the merits of user participation has as much clarity as would a law of gravity stating that a falling object may sometimes come down, occasionally go up, and periodically drift to the side' (Saleem, 1996 s. 145)

Användarmedverkan anses vara viktig för framgångsrik systemutveckling (till exempel Barki & Hartwick, 1994, Gulliksen & Göransson, 2002). Slutanvändare av informationssystem kan medverka i alla faser av systemutvecklingen och dela med sig av sin kännedom av de arbetsuppgifter som systemet ska stödja.

Det finns en mängd utvecklingsmetoder där användarmedverkan utgör en viktig del och ingår under bestämda former, till exempel deltagande design, joint applikation design och användbarhetsdesign (Davidson, 1999, Gulliksen & Göransson, 2002). Dessa metoder kallas ofta för användarcentrerade metoder (Gulliksen m.fl., 1999)

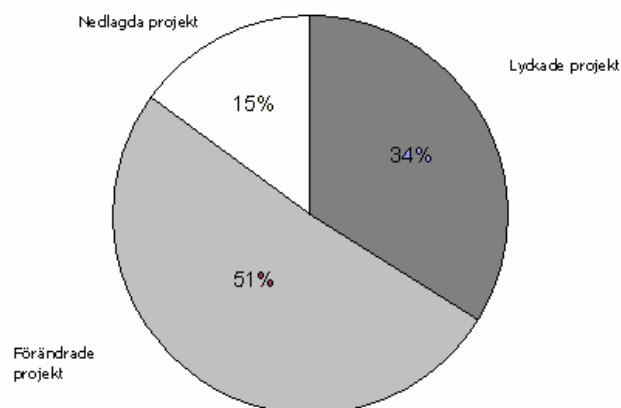
Enligt förespråkare för användarcentrerad systemutveckling borde användare medverka genom hela systemutvecklingen för att undvika misstag som beror på att man inte känner användarnas behov tillräckligt väl (Gulliksen & Göransson, 2002). Enligt Gulliksen och Göransson (2002) finns det dock inte många studier som evaluerar sambandet mellan användarmedverkan och användbarhet. (Användbarhet är ett begrepp som används för att beteckna systemkvalitet. Det definieras i avsnitt 3.2.1). Lin och Shao (2000) gjorde en metaanalys av flera fallstudier i systemutveckling och fann i endast ett fåtal fanns det ett positivt samband mellan användarmedverkan och systemkvalitet. I dessa fall fann man att användarmedverkan hade en positiv inverkan på användarnas attityd, involvering i och tillfredsställelse med systemet.

Däremot finns det många exempel på misslyckade användarcentrerade projekt (Gulliksen & Göransson, 2002, Lin & Shao, 2000). Gulliksen och Göransson förklarar detta med att även om många projekt sägs vara användarcentrerade är det sällan att de använder användarcentrerade metoder genom hela utvecklingsprocessen. I praktiken är det så att användare ofta bara är bjudna till samarbete i början och slutskedet på ett utvecklingsprojekt. Detta är olyckligt eftersom i de skeden som ingen användare medverkar, tas det många beslut som påverkar systemets användbarhet. Dessa beslut skulle i många fall inte ha tagits om användarna hade rådfrågats (Gulliksen & Göransson, 2002).

Trots bristen på handfasta bevis för nyttan av användarmedverkan finns det tecken på att användarcentrerade metoder börjar få ett allt större fotfäste i industrin (Gulliksen & Göransson, 2002). Företag är inte nöjda med resultaten av IT investeringar. I en studie av över 8000 IT utvecklingsprojekt hos 365 bolag i USA som görs vart annat år av Standish Group (CHAOS) kom det fram att många systemutvecklingsprojekt misslyckas även om situationen har förbättrats sedan den förra studien 1995 (Standish\_Group, 2003).

Figur 1.1 visar CHAOS rapportens resultat år 2003. Systemutvecklingsprojekten har förbättrats mycket vad gäller att hålla sig innanför tid- och kostnadsramar, från 16 % år 1995, till 34 % år 2003 medan nedlagda projekt är hälften så många år 2003 jämfört med 1995. Samtidigt är budgetunderskottet hos projekt som genomfördes med förändrade planer tre gånger mindre 2003 än 1995, även om frekvensen är den samma.

- 34 % genomfördes planenligt -lyckades
- 51 % genomfördes med förändrade planer - misslyckades
- 15 % av företagens IT-projekt avbröts - nedlagda



**Figur 1.1 CHAOS rapporten (gjord efter data från Standish Group, 2003)**

En av de viktigaste framgångsfaktorerna för de projekt som lyckas är användarmedverkan, tillsammans med ledningens stöd, en tydlig kravspecifikation och bra planering (Standish\_Group, 1995, 2003)

I Sverige finns inga jämförbara studier men det finns ingen anledning att anta att situationen är annorlunda här (Gulliksen & Göransson, 2002). I dag ägnas inte mer än 1 % av projektiden hos flera stora svenska företag åt användbarhetsrelaterat arbete. Tyvärr finns det inga studier på användbarheten hos de system som produceras (ibid).

Ovannämnda resultat tyder på att större användarmedverkan skulle kunna ge bättre systemkvalitet, givet att andra viktiga framgångsfaktorer också uppfylls men det behövs fler studier för att hitta de faktorer som bestämmer hur användarmedverkan påverkar systemkvaliteten.

I denna uppsats tänker vi analysera ett systemutvecklingsfall där en användare hade väldigt stor roll i utvecklingsarbetet. I vår analys skall vi försöka urskilja hur användarmedverkan påverkade projektets utfall. Vi börjar med att lägga ned de teoretiska referensramarna för att kunna få en heltäckande bild av vad det är vi studerar. Sedan beskriver vi fallet i empirikapitlet. Det handlar om ett projekt som startades 2003, där två systemdesigners (författarna) hjälpte en läkare (Daniel) på Universitetssjukhuset i Lund att utveckla en applikation för att hålla reda på blodaccesser hos dialyspatienter. Vi beskriver utvecklingsprocessen med betoning på användarens roll i utvecklingen. Till slut diskuterar vi hur användarens medverkan påverkade systemutvecklingen, hur detta fall överensstämmer med litteraturen och vilka lärdomar man kan dra av detta projekt.

## 1.1 Syfte

Vi har funnit i litteraturen att förespråkare för användarcentrerad utveckling hävdar att det är nödvändigt att användare medverkar i alla faser av systemutvecklingen men också att empiriska bevis för påståendet är oklara. Vi anser att det är viktigt för beställare och systemutvecklare att få klara besked om hur användare bör medverka i systemutveckling för att lyckas skapa ett användbart system som accepteras av framtida användare och effektivt stödjer verksamheten.

Syftet med studien är därför att bidra till ökade kunskaper om vilka faktorer påverkar nyttan av användarmedverkan i systemdesign.

## 1.2 Frågeställning

Vi vill veta hur, och i vilka delar av systemutvecklingsprocessen, användare bör medverka i för att skapa ett så användbart system som möjligt. Vi vill därför försöka svara på frågorna:

1. Hur påverkar användarmedverkan olika faser av systemutvecklingsprocessen samt systemets kvalitet?
2. Vilka faktorer påverkar nyttan av användarmedverkan i systemdesign?

## 1.3 Avgränsningar

Vi avgränsar vår studie till en analys av faktorer som påverkar nyttan av användarmedverkan i systemdesign i en fallstudie. Fallstudien bygger på en systemutvecklingsprocess och i denna har vi valt att göra vår analys i design- och analysfaserna. Att lägga analysen i alla faser i processen hade varit ett för stort arbete som vi inte hade kunnat hantera under den tid vi disponerade för studien.

Tyngdpunkten i denna studie ligger på analysen av de faktorer som under utvecklingsprocessen förändrades i takt med förändringar i nyttan av användarmedverkan i systemdesignen. De var därför potentiella faktorer som påverkar nyttan av användarmedverkan. Under studiens gång blev det för oss uppenbart att det var speciellt tre faktorer som visade sig vara kopplade till nyttan av användarmedverkan, nämligen maktförhållanden, kommunikation och motivation. Vi har därför fokuserat mest på dessa faktorer, både i teorin och i diskussionen.

Vi är medvetna om att många yttre omständigheter kan påverka de faktorer vi valt att studera. Men vi har inte haft möjligheten att inkludera dessa på grund av tidsbrist och risk för att studien blir för omfattande. Vi har helt enkelt inte haft möjlighet att ta med dessa och har därmed beslutat oss för att bortse från dem i denna studie.

De tekniska aspekterna med systemet, både vid design och i andra faser i processen kommer vi inte att ta hänsyn till. Detta delvis på grund av tidsbrist men också av praktiska skäl, då vi i praktiken tänkte lägga detta arbete till slutet av utvecklingen. Vissa mindre tekniska aspekter kommer ytligt att beröras för att ge en mer enhetlig bild av alla relevanta aspekter.

Ytterligare en avgränsning vi har fått göra på grund av tidsbrist är att fokusera på vår användare som slutanvändare, trots att han utöver användare även innehar rollen som beställare och även meddesigner. Att avgränsa oss på detta vis har gjort vår studie lättare att överblicka.

Slutligen har vi i denna fallstudie inte haft problematiken eller fördelen förvisso att göra ett urval av de användare vi ville studera. Vi har nästan enbart samarbetat med en användare som även är uppdragsgivaren och deltagare i utvecklingsprojektet. Därav att vi inte behövt beakta det litteraturen säger om hur man går tillväga för att välja användare. Vi behandlar därför inte detta ämne i uppsatsen.

## 2 Metod

### 2.1 Tillvägagångssätt

Vi börjar med att lägga ned de teoretiska referensramarna för att kunna få en heltäckande bild av studiens bakgrund. Sedan beskriver vi fallet i empirikapitlet. Till slut diskuterar vi hur detta fall överensstämmer med litteraturen och vilka lärdomar man kan dra av detta projekt.

Vi använder oss av kvalitativa metoder, en strukturerad berättelse av egna erfarenheter utformade som en fallstudie. På detta sätt försöker vi fånga upp subtila faktorer som kan tänkas påverka nyttan av användarmedverkan.

Projektet delades upp i två faser: I första fasen var det en användare som designade gränssnittet med hjälp av en interaktionsdesigner. I den senare var användaren inte med att designa gränssnittet men tog fortfarande aktivt del i de iterativa analys- och utvärderingsfaserna. Detta gav oss tillfälle att jämföra designfaserna, med och utan användarmedverkan.

### 2.2 Metodval

I denna studie tänker vi undersöka vilka faktorer påverkar nyttan av användarmedverkan i systemdesign. Enligt Gulliksen (2002) har tidigare fallstudier oftast inte använt riktigt användarcentrerade metoder, eftersom användarna inte medverkade hela tiden. Vi vill därför göra en fallstudie där användarna medverkar i hela utvecklingsprocessen. Det finns olika andra sätt att utföra denna studie, genom exempelvis frågeformulär eller intervjuer med medlemmar i projektteam. Vi anser dock att det uppstår problem med sådana metoder då det är så många variabler att ta hänsyn till att man lätt kan få motsägande resultat från situationer som på ytan verkar snarlika. Vi vill studera på djupet hur interaktionen mellan användare och systemutvecklare påverkar utvecklingsprocessen och slutproduktens användbarhet.

Enligt Gulliksen (2002) är grunden i ett användarcentrerat arbete att skaffa sig en klar bild av, och en djup förståelse för de användare som man utvecklar systemet för. Det räcker inte att konstatera att det finns användare utan man måste ta hänsyn till användarnas speciella förutsättningar när man utformar systemet. Vidare säger Gulliksen (2002) att det finns flera sätt att skaffa sig denna typ av information men att alla sätt handlar om att på ett eller annat sätt gå ut till användarna och studera hur deras arbetssituation ser ut, d.v.s. genom fallstudier. Gulliksen (2002) menar att man genom fallstudier verkligen kan gå ut till användarnas arbetsplats och studera, intervjuar eller bara tillbringa tid med användarna. Detta är oerhört viktigt för att skapa sig en bra och rättvis bild av vilken typ av system som bör utvecklas. Då vi fann det svårt att hitta tillgång till deltagare av ett användarcentrerat projekt, d.v.s. ett projekt där användarna medverkar genom hela utvecklingsprocessen, var vår nästa bästa alternativ att själva genomföra ett användarcentrerat systemutvecklingsprojekt.



### 2.3 Fallstudier

En fallstudie är en metod som kan utnyttjas för att systematiskt studera en företeelse och är att föredra när man ska undersöka aktuella eller nutida skeenden men då det inte går att manipulera relevanta variabler. Dess unika styrka är dess förmåga att hantera många olika typer av empiriskt material (dokument, artefakter, intervjuer och observationer). Vilken typ av frågor man ställer, vilken grad av kontroll man har och hur man tänker sig att slutresultatet ska bli är faktorer man måste ta hänsyn till då man ska avgöra om en fallstudie verkligen är det som passar bäst för att undersöka något av intresse. En annan avgörande faktor är om man kan identifiera ett *avgränsat system* som fokus för undersökningen. Ett avgränsat system som man vill studera på djupet (Merriam, 1994).

En systemutvecklingsprocess är alltid unik eftersom den utgörs av ett komplext förhållande mellan människor med olika kunskaper och beteendemönster och en utvecklingsuppgift som kan lösas på många sätt. Vi använder fallstudieformen därför att den ger oss möjlighet att få en helhetsbild av användarmedverkan i samspel med andra faktorer i utvecklingssituationen. Vårt avgränsade system är därför utvecklingssituationen, där en eller flera medlemmar av projektgruppen arbetar med uppgiften att skapa ett datoriserat system. Detta är en naturalistisk värdering (Merriam, 1994), där vi försöker bedöma och förklara hur användarmedverkan påverkar utvecklingsprocessen och nyttan med systemet.

Vår studie är värderande, vilket innebär att den inbegriper beskrivning, förklaring och bedömning (ibid.). En rad forskare såsom (Guba & Lincoln, 1981; Patton, 1980; Stake, 1981; Goetz & LeComte, 1984; i Merriam, 1994) har gått igenom olika typer av rapporter och de drar slutsatsen att fallstudier är den bästa formen att redovisa studier av naturalistiska värderingar (Merriam, 1994). Detta för att fallstudier kan ge en ”tät beskrivning”, är grundade i ett empiriskt material, är holistiska och verklighetsnära. Dessutom förenklas informationen för en läsare och olika innebörder belyses. Framför allt är dock den här typen av fallstudier viktig genom att den tillhandahåller information utifrån vilken man kan göra en bedömning, som är det slutgiltiga resultatet av en utvärdering.

I denna fallstudie har vi använt deltagande observationer som pågick under hela projekttiden och dokumenterades löpande. Det har också utförts halvstrukturerade intervjuer med projektdeltagarna av en utomstående forskare. Detta har bäst lämpat sig i vår undersökning då vi även deltagit i själva utvecklingen av systemet. Vi är medvetna om problematiken i att granska sig själv. Vi har försökt vara objektiva och självkritiska men den möjlighet finns att vi omedvetet låtit våra egna känslor och åsikter påverka hur vi presenterar fallet. För att minska risken för detta har samma forskare som utförde intervjuerna granskat och godkänt vår tolkning.

### 2.4 Observationer

Deltagande observation är enligt Bryman (1997) den mest kända av de metoder för insamling av information som förknippas med kvalitativ forskning. Denna innebär att forskaren under lång tid ägnar sig helt åt undersökningspersonerna i syfte att få en heltäckande och djupgående bild av det som studeras. Det är dock sällan som deltagande observatörer enbart observerar, de brukar även samtala med aktörerna och även rentav genomföra intervjuer (Bryman, 1997).

Vi (Vin och Danisa) har i denna fallstudie haft tre olika roller: helt och hållet forskare, forskare och deltagare samt helt och hållet deltagare.

- Som helt och hållet forskare observerade vi skeenden där vi inte deltog alls.
- Som forskare/deltagare var vi med i en viss aktivitet men då som forskare mer än som deltagare.
- Som helt och hållet deltagare agerade vi spontant och personligt som projektdeltagare, och analyserade först i efterhand den aktivitet vi deltog i.

Fördelen med detta arbetssätt var att vi kunde koncentrera oss på en enda uppgift åt gången. Var man exempelvis helt och hållet forskare fick man koncentrera sig på att observera och använda alla sina tankar på att analysera och anteckna intressanta förlopp. Vi alternerade mellan dessa roller, ibland var det Danisa som observerade medan Vin interagerade med användaren, ibland omvänt. Detta för att inte fastna i någon av rollerna vilket skulle kunna begränsa vår tolkning av situationen (se diskussion om validitet nedan).

Vi använde även observationsmetoden med inslag av att tänka högt. I detta fall rörde det sig om att observatören satt 1 meter bakom användaren och var så tyst som möjligt. Användaren var instruerad att tänka högt så mycket som möjligt men inte så mycket att det störde honom i arbetet. Observationen spelades inte in, utan noterades av observatören. Denna metod växte fram under projektets gång och upplevdes som bekväm och effektiv av såväl utvecklare som användare. Vid observationer följdes Göranssons och Gulliksens (2000) riktlinjer.

## 2.5 Intervjuer

Vi utförde intervjuer efter projektets slut. Den typen av intervjuer som bäst lämpade sig var halvstrukturerade intervjuer (eller forskningsintervjuer), eftersom de har bästa möjligheter att nå över helheten av deltagarnas upplevelse av projektet (Kvale, 1996) medan strukturerade intervjuer kan begränsa de uppgifter man får och ostrukturerade samtal kan göra att samtalsämnet spårar ur. Forskningsintervjun ger kvalitativa texter i stället för kvantitativa data (ibid.). Den speglar även alternativa föreställningar om vad som ska vara föremål för samhällsvetenskaplig forskning. Tekniskt sett är den kvalitativa forskningsintervjun halvstrukturerad, det vill säga varken ett öppet samtal eller ett strängt strukturerat frågeformulär. Intervjun går utöver det spontana vardagliga utbytet av åsikter och blir ett sätt för intervjuaren att genom omsorgsfullt ställda frågor och lyhört lyssnande erhålla grundligt prövade kunskaper. Forskningsintervjun är inte ett samtal mellan likställda parter, eftersom det är forskaren som definierar och kontrollerar situationen (ibid.).

Vi utförde två intervjuer med projektdeltagarna, en respektive tre månader efter projektets slut. Skillnaden i deltagarnas åsikter mellan dessa intervjuer var liten och de slogs därför ihop inför analysen. Vi fick en utomstående forskare att utföra intervjuerna för att minska eventuell bias på grund av de speciella relationerna som hade bildats mellan projektdeltagarna och för att minska sannolikheten att intervjuarens egna åsikter påverkade intervjun.

I den kvalitativa forskningsintervjun bygger man upp kunskap (Kvale, 1996). Det rör sig bokstavligen om ett samspel, om ett utbyte av synpunkter mellan två personer som samtalar om ett ämne av gemensamt intresse. Arten av denna kunskap resulterar i empirisk kunskap om vardagsvärlden.

Forskningsintervjuer brukar handla om två personer, men det är också möjligt att intervjua flera personer åt gången (Kvale, 1996). I vårt fall deltog alla projektdeltagarna i intervjuerna för att

stimulera diskussionen och ge deltagarna chansen att jämföra sina upplevelser, men också för att återge teamets åsikter eftersom ett team är mer än bara summan av deltagarna (Hägerfors, 1995).

Vi använde oss av fokuserade diskussioner, där alla deltagarna förde fram sina åsikter, och av enskilda utlåtanden vid utvärderingen av hur användarmedverkan påverkade utvecklingsprocessen. Diskussionerna spelades in och relevanta delar nedtecknades av intervjuaren som vi ansåg vara opartisk eftersom hon inte hade deltagit i projektet innan.

Dessa diskussioner ägde rum vid två tillfällen efter projektets slut, vilket har både för- och nackdelar. Det är klart att det som står nära i tid ihågkommes bättre men vi anser också att det kan vara bra för objektiviteten att få en distans på händelserna. Det hade varit att föredra att spela in diskussioner regelbundet under projektets lopp, som vi inte gjorde, och även föra diskussioner vidare efter dessa tillfällen, men detta gjordes inte heller. De delar av intervjuerna som nedtecknades är inbakade i fallstudien..

## 2.6 Studiens validitet, reliabilitet och generaliserbarhet

I en vetenskaplig undersökning påverkas arbetet av det perspektiv som man i rollen som forskare utgår från. Det finns två huvudspår som man brukar skilja mellan: det positivistiska och det hermeneutiska (Bryman, 1997). Positivism handlar om att endast dra slutsatser från uppgifter som är säkra, exakta och entydiga, och som härrör från kvantitativa, mätbara data. Hermeneutiken kan betraktas som motpol till positivismen, och med ett sådant perspektiv antar man en helhetssyn i vilken fokus ligger på kvalitativa data.

Vi har i vår undersökning valt att utgå från ett hermeneutiskt perspektiv, eftersom vi har ansett att vår undersökning kräver förståelse på djupet och därmed har vi använt oss av en kvalitativ undersökningsmetod. Denna har resulterat i en strukturerad berättelse av egna erfarenheter utformade som en fallstudie.

Enligt Bryman (1997) måste följande problem has i åtanke när man använder kvalitativa undersökningsmetoder:

1. Problem med tolkning - forskarens förmåga och möjligheter att se verkligheten genom andra människors ögon och tolka det som sker utifrån deras perspektiv.
2. Teori och forskning - studera förhållandet mellan teori och forskning inom den kvalitativa traditionen
3. Fallstudier och problemet med generalisering - ta ställning till i vilken mån kvalitativa forskningsresultat kan generaliseras

Problem med tolkning - att vara empatisk utan att vara sympatisk kan vara ett problem. Frågan är om forskaren verkligen kan komma fram till en beskrivning av sina aktörers perspektiv och hur man i så fall ska kunna veta om tolkningarna av deras perspektiv är giltiga.

Samma frågeställning uppstår också då vi ställs inför två forskares olikartade uppfattningar av en och samma situation. Om målet är att betrakta situationen genom aktörens ögon borde man kunna förvänta sig en viss överensstämmelse i tolkningarna och resultaten.

I denna studie har vi försökt återge alla deltagares åsikter jämnt. När det har varit olika åsikter har deltagarna kommit fram till en gemensam position, men de ursprungliga olikheterna skildras ändå i studien.

Teori och forskning – att specificera teorin i förväg brukar vara något man undviker på grund av risken att dra för snabba slutsatser eller skapa en felaktig förhandsbild av de frågor som ska studeras. Bearbetningen och tillämpningen av teorier innan en kvalitativ undersökning startar kan störa forskarnas förmåga och möjlighet att se världen genom aktörernas ögon. Teorin kan i stor utsträckning utgöra ett hinder för kvalitativa forskare och göra de blinda inte enbart för aktörernas uppfattningar och synsätt utan också för ovanliga och oväntade aspekter av den sociala verkligheten (ibid.)

Inga speciella åtgärder vidtogs för att förhindra detta problem, förutom att försöka vara så objektiva som möjligt i dokumentation och analys. Det finns därför en fara att våra förutfattade meningar har påverkat hur vi tolkade händelser och vilka händelser vi fokuserade på.

Fallstudier och problemet med generalisering – forskning som härrör från studier av en enda miljö kan utgöra ett problem som rör i vilken utsträckning det är möjligt att generalisera resultaten. Enligt Bryman finns ett antal lösningar på dessa problem:

1. Den kvalitativa forskaren kan studera mer än ett fall. Fler än en forskare studerar ett eller flera fall. Undersökningen får därmed karaktär av ”teamforskning”. Fördelen är att en grupp har större möjligheter än en ensam forskare att ta sig an och bearbeta alla aspekter av den miljö eller situation som studeras. Den större bredd som ett team kan täcka utgör en mycket viktig fördel, men man kommer ändå inte från problemet att man trots allt är begränsad till det enstaka fall man studerat.
2. Ett tredje angreppssätt när det gäller problemet med generalisering av fallstudieresultaten går ut på att man letar efter fall som är ”typiska” för ett visst kluster av kännetecken.

I vårt fall är det två forskare som studerar ett fall, vilket gör att studien får en större bredd, forskarna tolkar inte alltid situationer på samma sätt, en ser ibland vad den andra missat. Fallstudier kan ha hög validitet för en avgränsad situation men som skildrat ovan är generaliserbarheten ofta låg. Dock kan många fallstudier tillsammans ge en högre generaliserbarhet. Även om vi bara studerar ett fall, analyserar vi det med hänsyn till andra fall i litteraturen. Den kan därför ge indikationer och stödja eller ifrågasätta resultat från andra studier.

Vår fallstudie grundar sig på en ganska unik situation, vilket minskar generaliserbarheten (Kvale, 1996). Den kan dock ge mervärde till andra studier genom att tyda på en utökad eller avgränsad generaliserbarhet hos andra studier. Om andra studier får liknande resultat tyder studien på att deras resultat också gäller i andra situationer och deras generaliserbarhet stöds, samtidigt som validiteten och generaliserbarheten hos vår studie får stöd.

Reliabiliteten är låg i en enkel fallstudie (Bryman, 1997) eftersom det ofta är så många variabler som inte går att styra och som kan påverka studiens resultat. Det är därför inte säkert att om vi utförde samma studie igen att vi skulle hitta samma faktorer som i denna studie påverkade nyttan av användarmedverkan.

Det går därför inte att dra några definitiva eller generaliserade slutsatser av denna studie i isolering, men den är av värde i sammanhang med resultat från andra studier och teori om ämnet.

## 3 Teoretisk referensram

### 3.1 Definitioner

Med *användare* menas en slutanvändare som använder systemet för att utföra sitt dagliga arbete (Gulliksen, 2002). Vidare säger han att användaren nästan alltid är skild från den kategori som agerar beställare av system. Distinktionen är viktig därför att beställare vanligtvis inte vet hur slutanvändare arbetar och reagerar. I vårt fall, som presenteras nedan, är dock beställaren och användaren samma person. Utöver detta är personen även meddesigner av systemet.

*Design* är ett besvärligt begrepp som används i många olika sammanhang. Gulliksen och Göransson använder termen användarcentrerad systemdesign, UCSD (User Centered System Design) i stället för användarcentrerad design, UCD (User Centered Design) som i princip kan handla om design av vad som helst, inte bara datorsystem eller mjukvara. Göransson och Gulliksen (2000) menar att begreppet design, använt i sammanhanget att utforma ett datorsystem, hänvisar både till utvecklingsprocessen och till representationer av systemet, prototyper, modeller eller simulationer. Vi kommer att använda båda sätten omväxlande och läsaren kan förutsätta att när vi pratar om design är det systemdesign vi menar.

*System* använder vi i meningen mjukvara som brukligt är i vår utbildning. Ett informations-system är en mjukvara som tar emot och bearbetar data samt returnerar någon slags information.

*Användarcentrerad systemdesign* är en process som fokuserar på användare och användbarhet genom hela utvecklingsprocessen och vidare genom hela livscykeln (Gulliksen, 2002).

*Livscykeln* är ett begrepp som används för att beskriva de faser som alla systemutvecklingsprojekt går igenom oavsett utvecklingsmetod. Alter (2002) beskriver ett systems livscykel som bestående av fyra faser, vilka är oberoende av utvecklingsmetod. Den innefattar analys, design, implementation och underhåll. Användarmedverkan kan påverka ett system i alla fyra faser, men analys och designfasen är vanligast associerade med användarmedverkan (ibid.)

*Användarmedverkan*: Behandlas i separat avsnitt.

*Användbarhet*: Behandlas i separat avsnitt.

*Nytta*: Eftersom vi vill undersöka nyttan med användarmedverkan måste vi definiera nytta på ett godtagbart sätt. Nyttan kan betyda olika saker i olika situationer. I vårt fall kan användarmedverkan vara nyttig på två sätt. Den kan antingen öka kvaliteten på slutprodukten d.v.s. systemet eller påverka designprocessen på ett positivt sätt. Vi har koncentrerat oss i denna uppsats på hur användarmedverkan påverkar systemets kvalitet.

### 3.2 Användbarhet

#### 3.2.1 Definition

Ett vanligt sätt att beskriva ett systems kvalitet är genom begreppet *användbarhet*. Användbarhet definieras i ISO (ISO\_9241-11, 1998) så här:

*Användbarhet är den utsträckning i vilken specifika användare kan använda en produkt för att uppnå specifika mål, med ändamålsenlighet, effektivitet och tillfredsställelse, i ett givet användningsområde.*

Enligt Ottersten & Berndtsson (2002) är användbarhet en kvalitetsegenskap hos interaktiva produkter. En produkt har hög användbarhet om den uppfyller beställarens och målgruppernas syften. En produkts användbarhet visar sig i samspelet mellan produkten och dess användare och över en tidsperiod - kort sagt: *i användningen*. För att skapa en användbar produkt behöver man utforma produkten med hänsyn till (Ottersten & Berndtsson, 2002):

1. Det mänskliga systemet - egenskaper som bärs av individerna som använder produkten.

Det finns:

- generella egenskaper - gemensamma mönster för hur vi människor ser, uppfattar och minns information
- specifika egenskaper hos målgrupperna (exempelvis kunskaper, värderingar, attityder och förväntningar liksom eventuella funktionsnedsättningar)

2. Det sammanhang där produkten ska användas. Produkten måste anpassas till det sammanhang som den skall användas i:

- det fysiska sammanhanget, exempelvis dålig belysning och störande ljud
- det psykiska sammanhanget, exempelvis stress
- det sociala sammanhanget, exempelvis kompisrelationer och social ställning
- det organisatoriska sammanhanget, exempelvis huvudkontoret eller ordermottagningen

3. Den nytta som produkten förväntas ge. En interaktiv produkt förväntas ge någon effekt, bidra med någon nytta:

- för den som tillhandahåller produkten, exempelvis samhällsnytta, verksamhetsnytta eller ekonomisk vinst.
- för den som använder produkten, exempelvis förenkling, effektivisering eller nöje.

Användbarhetsbegreppet har historiskt utvecklats från att enbart fokusera på det mänskliga systemet, till att se produkten och dess användning i ett sammanhang i vilket den skall användas (Löwgren, 1993). De flesta metoder och tekniker (såväl användbarhets- som systemutvecklingsmetoder) behandlar dock kopplingen mellan den förväntade nyttan och produktens utformning ytligt, eller inte alls (Ottersten & Berndtsson, 2002). Detta är en allvarlig brist. En produkt som inte uppfyller den förväntade nyttan kan inte anses vara en produkt med hög användbarhet även om produkten är väl anpassad till det mänskliga systemet och sammanhanget. Ottersten & Berndtsson (2002) betonar behovet av att utgå från den förväntade nyttan för att säkerställa en produkts användbarhet.

Användbarhet är ingen objektivt observerbar, inneboende produktenskap så som en färg eller en funktion (Ehn m.fl., 1995). Användbarhet är istället en egenskap som uppstår i produktens användning. Ett begrepp som används är "quality-in-use" eller försvenskat, användningskvalitet där fokus sätts på användningen av produkten, istället för på själva produkten i sig. Då blir det också tydligt att användbarheten är en kvalitetsegenskap och att produktens användbarhet är beroende av det sammanhang i vilket den används. Användbarhet är en interaktiv egenskap och ses som bestående av anpassning till uppgiften, användarvänlighet, användaracceptans och användarkompetens (Allwood, 1998).

Begreppet användbarhet dök upp någon gång i slutet på 80-talet som en ersättare för termen ”användarvänlighet” som blivit kritiserad för att ge upphov till vaga och subjektiva tolkningar. (Nielsen, 1993). Enligt Nielsen är termen användarvänlighet opassande av flera skäl. För det första kräver inte användare att datorer ska vara vänliga mot dem, det räcker att de inte krånglar och står i vägen för de arbetsuppgifter som ska utföras. För det andra antyder termen att användarnas behov kan beskrivas längs en enkel dimension med system som är mer eller mindre vänliga. I verkligheten har olika användare olika behov och ett system som är vänligt mot någon kan upplevas som tjatigt av någon annan. Begreppet användbarhet har även det urholkats något eftersom det i praktiken används av olika människor på olika sätt. Kalén (1997) frågade i en undersökning 300 svenska systemutvecklingsföretag hur de definierar användbarhet och i mer än hälften av fallen associerades användbarhet med ett systems funktionalitet. När man har ställt samma fråga till användare har de ofta nämnt mer kontextuella variabler som arbetsmiljö i sin tolkning.

Shackel (1991) hävdar att design för användbarhet handlar om att uppnå harmoni mellan de fyra komponenterna användare, datasystem, arbetsuppgifter och arbetsmiljö som utgör kontexten för användning. En viktig innebörd av detta perspektiv är att användbarhet väsentligen är en egenskap hos interaktionen mellan användare, deras arbetsuppgifter och datorn i en viss arbetsmiljö. En annan viktig innebörd är att användbarhet är ett mycket kontextkänsligt och dynamiskt koncept.

Det har gjorts flera försök att definiera användbarhet i mer formella termer. En av de mest citerade och inflytelserika definitionerna kommer från Shackel (1991) som beskriver ett systems användbarhet som:

*”The capability in human functional terms to be used easily and by the specified range of users, given specified training and user support, to fulfil the specified range of tasks, within the specified range of environmental scenarios.”*

Dessa ord känns igen i den ovannämnda definitionen av användbarhet av ISO (ISO\_9241-11, 1998).

Det viktiga med definitioner som dessa är att de kan tillhandahålla kriterier enligt vilka användbarhet kan uttryckas i kvantifierbara, och därför mätbara, termer. Till exempel föreslår Shackel (1991) fyra kriterier enligt vilka ett system borde utvärderas för att avgöra dess användbarhet. De fyra kriterierna är *effektivitet*, *lärbarhet*, *flexibilitet* och *attityd*.

#### *Effektivitet*

Med ett systems effektivitet menas det urval av uppgifter som utförts av erfarna användare med avseende på hastighet och antal gjorda fel.

#### *Lärbarhet*

Den tid och ansträngning som krävs av användare för att nå en viss nivå av uppgiftsgenomförande efter utbildning eller omlärning.

#### *Flexibilitet*

Den utsträckning till vilken systemet kan anpassa sig till förändringar i uppgifter och miljöer utöver de som systemet ursprungligen anpassades för.

#### *Attityd*

Den grad av positiv attityd som användarna upplever vid användning av systemet.

Nielsen (Nielsen, 1993) menar att det är viktigt att inse att användbarhet inte är en specifik endimensionell egenskap hos ett användargränssnitt. Enligt honom består användbarhet av flera komponenter och har traditionellt förknippats med de fem användbarhetsattributen *lärbarhet*, *effektivitet*, *igenkänningsbarhet*, *fel* och *belåtenhet*. Dessa komponenter stämmer bra överens med Shackels fyra kriterier ovan. Precis som Shackel menar Nielsen att genom att definiera det något abstrakta uttrycket ”användbarhet” med hjälp av mer precisa och mätbara komponenter kan man på ett systematiskt sätt utvärdera och förbättra användbarheten hos olika system. Nielsen menar att även om det inte finns något intresse av att genomföra formella mätstudier av användbarhetsattributen så är det nyttigt att fundera över hur systemets användbarhet skulle kunna mätas. Vid utvecklingsarbetet är det ofta bättre att förtydliga de mätbara aspekterna av användbarhet än att endast sträva efter att uppnå en allmän känsla av ”användarvänlighet”. Nielsens (Nielsen, 1993) fem användbarhetsattribut förklaras på följande sätt:

#### *Lärbarhet*

Lättlärdhet är kanske det mest fundamentala användbarhetsattributet och innebär att systemet ska vara lätt att lära sig så att användarna snabbt kan komma igång och utföra uppgifter med hjälp av systemet.

#### *Effektivitet*

Systemet ska vara effektivt att använda, så att en hög produktionsnivå kan uppnås efter det att användarna har lärt sig systemet.

#### *Igenkänningsbarhet*

Systemet ska vara lätt att komma ihåg så att användarna kan komma tillbaka till det efter en tid utan att behöva lära sig allt på nytt igen.

#### *Fel*

Systemet ska ha en låg felnivå så att användarna gör få fel när de interagerar med det. I de fall användarna ändå gör fel ska de lätt kunna återhämta sig från dem. Dessutom måste systemet förhindra att allvarliga fel kan begås av användarna.

#### *Belåtenhet*

Systemet ska vara behagligt att använda så att användarna känner sig nöjda vid och efter interaktion med systemet (Nielsen, 1993).

Man kan argumentera för att två av Nielsens kriterier, *lärbarhet* och *igenkännbarhet*, är två sidor av samma peng. *Lärbarhet* och *igenkännbarhet* är också viktigare i vissa fall än andra. Till exempel är de mycket viktiga i system som används för att utföra en enkel uppgift med något långa mellanrum som till exempel kontantautomater, billjettautomater och nätbutiker. I andra fall, där man behöver kontinuerligt utföra komplexa uppgifter i sitt arbete med hjälp av till exempel vissa bildhanteringsprogram, utvecklingsprogram och simulationsprogram är funktionaliteten viktigare än *lärbarheten*. Därför verkar *belåtenhet* vara ett bättre kriterie för användbarhet än *lärbarhet* eller *igenkännbarhet*. *Belåtenhet* igenkänns i ISO definitionen, Nielsens och Otterstens definitioner. Det är ett problematiskt uttryck eftersom olika människor blir belåtna av olika anledningar och därför att det egentligen säger ingenting om systemets egenskaper. Egentligen betyder kravet på *belåtenhet* att en nödvändig del av användbarhetsdesign är att analysera användarnas behov och fråga dem om deras önskemål. Shackels definition verkar mycket objektivare eftersom han undviker termen *belåtenhet* men det skulle innebära att kravspecifikationen blir det enda måttet på användbarhet, om den uppfylls så är systemet användbart, annars inte. Shackel kommer dock inte helt bort från användbarhetens subjektivitet.



Istället för belåtenhet använder han uttrycket ”easily” eller ”med lätthet” vilket egentligen innebär att användarna inte upplever några problem med användningen av systemet och det är svårt att se att belåtenhet inte skulle fungera lika bra i sammanhanget. Utöver det är graden av positiv attityd, vilket är en synonym med belåtenhet, en av Shackels kriterier för användbarhet även om han undviker termen i sin definition.

Ett av Nielsens kriterier, felnivå eller felfrekvens är en egenskap som bör vägas mot effektivitet och belåtenhet. Naturligtvis bör man sträva efter att göra systemets funktioner så transparenta och förståeliga som möjligt. Utöver det är det möjligt att förebygga många fel genom att använda sig av varningsmeddelanden när användaren försöker göra en övergripande eller oåterkallelig åtgärd och genom att göra vissa funktioner otillgängliga. Detta är i många fall nödvändigt och fungerar bra, men i andra fall går det utöver systemets effektivitet och användarnas belåtenhet. Därför borde låg felnivå inte utan förbehåll ingå i en definition av användbarhet.

ISO definitionens tre kriterier, ändamålsenlighet, effektivitet och belåtenhet omfattar och specificerar Nielsens och Shackels kriterier på ett tillfredsställande sätt. Ändamålsenlighet innefattar kravspecifikationens uppfyllning vilket utgör det mesta av Shackels definition, effektivitet och belåtenhet återfinns också i bådas kriterier. Hittills är därför ISO definitionen överlägsen Shackels definition eftersom den är enklare - ändamålsenlighet och belåtenhet ersätter den helt medan ISO definitionen innehåller också effektivitet vilket är en viktig egenskap som både Nielsens och Shackels kriterier också innehåller. Shackels kriterier, igenkännbarhet, lärbarhet och felnivå, som inte ens ingår i hans definition, är dessutom bristfälliga, se ovan.

Däremot är Shackels sista kriterie, flexibilitet, en mycket viktig egenskap i dagens förändrande miljö. Det är dock diskutabelt om det tillhör uttrycket användbarhet eftersom ett oflexibelt system kan vara mycket användbart vid en viss tidpunkt. Det skulle i så fall handla om förlängd användbarhet. Vi tycker att det är onödigt att komplicera definitionen av användbarhet med tidsdimensionen i detta fall och begränsar oss till att bedöma den användbarhet som finns här och nu. ISO definitionen ägnar sig bra till detta. Den reflekterar också Ottersteins och Berndtsons observation att användbarhet uppstår i interaktionen mellan användare och systemet. Den är därför inte absolut utan beror på de specifika användarnas egenskaper och mål.

ISO-definitionen gör klart att man måste anpassa lösningen till målgrupperna (specified users), användningssituationen (specified context) och målgruppernas förväntade nytta (specified goals). Den säger däremot inget om i vilken utsträckning produkten ska uppfylla beställarens syften för att anses som användbar, utan fokus ligger främst på målgruppernas syften. Användbarhet som den definieras av ISO räcker därför inte som mått på ett systems kvalitet, eftersom användarna har ofta andra mål än verksamheten eller beställaren. Ett system kan därför i princip ha bra användbarhet utan att uppfylla beställarens krav. Ett kvalitetsbegrepp är meningslöst om det inte innefattar beställarens förväntningar. I vårt projekt är användaren och beställaren dock en och samma person och kan därför inte ha motstridiga mål<sup>1</sup>. ISO - definitionen för användbarhet räcker därför i detta fall.

### 3.2.2 Nyttan med användbarhet

Idag när datorer används i mycket större utsträckning och av personer som egentligen inte vet så mycket om datorer så är det viktigt att lägga resurser på att göra livet lättare för användarna (Nielsen, 1993). Det är även lönsamt för företagen. Mer intuitiva gränssnitt och lättutförda

---

<sup>1</sup> Det är möjligt att användaren har motstridiga mål, som orsakar en inre konflikt. Vi har inte utrymme här för att diskutera detta eller dess möjliga påverkan på projektarbetet, men bör ha i åtanke vid utformning av målanalysen.

funktioner gör att jobbet blir gjort snabbare vilket i sin tur leder till bättre effektivitet och minskade kostnader, se nedan.

Själva poängen med att använda datorer är att det ska bli lättare att utföra de uppgifter vi förutsatt oss att utföra. Koncentrationen ska ligga på att lösa de problem som har med en uppgift att göra och slippa ödsla tid på problem som har med datorn eller programmet att göra. Ofta flyttar man uppgifter till datorn och ett visst program för att höja kvalitén på arbetsresultatet. Tidigare trodde man att detta uppnåddes genom att programmen tillhandahåller tillräcklig *funktionalitet*. Det är givetvis en förutsättning men programmen måste även ha god *användbarhet* för att öka kvaliteten på arbetsresultatet.

Detta är något man mer och mer tagit fasta på. Det spelar ingen roll att funktionaliteten i ett program är utmärkt om användarna inte förstår hur de ska använda sig av den. Programmets *effektiva funktionalitet* bestäms alltså av interaktionen mellan programmets funktionalitet och programmets användbarhet (Allwood, 1998).

Dålig användbarhet hos interaktiva system kan resultera i finansiella förluster samt missnöje och stress bland anställda vilket i sin tur kan leda till hög personalomsättning och minskad produktivitet. Dålig användbarhet kan även kräva omfattande utbildning på systemet vilket kan minska de förväntade förtjänsterna från investeringen i systemet. Kvaliteten och användbarheten hos ett interaktivt system är därför av största vikt för produktiviteten och förtjänsten hos organisationer där nya datasystem introduceras (Kalén, 1997).

Ottersten och Berndtson (2002) har tagit samman de fördelar som användbara system inbringar. Här tar vi upp några av dem, med fokus på nyttan som uppstår i verksamheter eftersom fallet handlar om en produkt som ska användas i verksamhetsmiljö:

- System med bra användbarhet bidrar till ökad produktivitet och välmående hos de anställda. Ett datorsystem ska underlätta arbetet och öka produktiviteten. Om produkten är omständlig att använda måste användaren hela tiden anstränga sig för att lyckas. Onödig tid går åt till att fundera ut hur produkten fungerar och på att förstå krångliga arbetsflöden. I värsta fall går mycket av användarens tid åt till att rätta fel som orsakas av produktens dåliga användbarhet.
- Kort inläringstid är viktigt för produkter som inte används dagligen eller där det ofta tillkommer nya användare. Om produkten är utformad så att den utgår från användarens förståelse av ämnesområdet och stöder lärandeprocessen så minskas inläringstiden. Exempel på sätt att stödja lärandeprocessen är att använda för användaren bekanta begrepp, att ge tydlig återkoppling och att produktens utformning och beteende följer ett tydligt mönster. Hjälpfunktioner är också ett sätt att stödja lärande. Det kan även vara en god idé att skapa speciella avsnitt med lärande som enda syfte, dessa kan vara integrerade i hjälpfunktionerna eller vara en egen interaktiv utbildningsprodukt som används i anslutning till huvudprodukten. Kostnaden för inläring av produkter med dålig användbarhet samt att denna orsakar problem att man inte når avsedda effekter i verksamheten glöms ofta bort.
- En av de mer okända nyttorna med att genomföra användbarhetsaktiviteter i utvecklingsprojekt är att tiden för utveckling kan minskas. I IT-projekt är det vanligt att mycket tid det går åt till att diskutera vad produkten "egentligen" ska fylla för syfte. Sådana diskussioner är mycket viktiga, men det behövs konkreta tekniker för att formulera syftet och för att sedan styra processen så att de förväntade effekterna uppstår. Annars riskerar man att projektet styrs av tyckanden, som inte baserats på kunskap om målgrupperna och

användningssituationerna. Om projektet styrs av tyckanden ökar dessutom risken att det blir många ändringar på vägen. Genom att basera alla beslut som rör produktens utformning på kunskap om produktens användning så minskas kostnader och tidsåtgång väsentligt.

- Det är känt att många projekt överskrider tids- och kostnadsramar och att detta till stor del beror på att man inte har tillräcklig kunskap om användarnas behov. Arbetssätt och tekniker för att samla kunskap om användningssituationer och att testa föreslagna lösningar i realistisk miljö är mycket viktiga. Sådana insatser ger dessutom bättre styrbarhet i projekt. Genom att man känner till vilka målgrupper som är viktigare än andra och vilka egenskaper och funktioner som är viktigast är det lättare att göra prioriteringar och styra projekten.
- En annan konkret nytta med användbarhetsinsatser är att beställare och användare engageras på ett effektivt sätt. Även om beställare och användare ibland ingår i arbetsgrupper och beslutsgrupper saknas ofta effektiva tekniker för att hantera beställar- och användarmedverkan. Utan dessa tekniker tenderar beställar- och användarmedverkan att reduceras till tyckanden och pålagor som ofta kommer för sent. Några effektiva tekniker för att skapa insyn och medverkan är kartläggning av idé, målgruppsanalys och användningstest.
- Ett stort problem i systemutveckling är att kostnader för drift, utbildning, support och underhåll ofta glöms bort då man ska bedöma och beräkna kostnaderna. Största delen av kostnaderna i underhållsfasen beror på att produkten inte tillgodoser användares behov. Genom systematiska insatser som exempelvis målgruppsanalyser, interaktionsdesign och användningstester för att säkra hög användningskvalitet minskas livscykelkostnaderna.

### 3.2.3 Olika nivåer i användbarhetsarbete

Enligt Ottersten (Ottersten & Berndtsson, 2002) kan användbarhetsarbete göras på olika ambitionsnivåer i ett projekt. Vilken ambitionsnivå som väljs beror på en mängd faktorer, som exempelvis mognad hos beställaren och utvecklingsorganisation, samt hur kritiskt produkten är för verksamheten.

1. Låg ambitionsnivå: De personer som ingår i projektgruppen genomför någon eller några aktiviteter för att säkra användbarheten.
2. Medelhög ambitionsnivå: På medelhög ambitionsnivå läggs en ny roll till projektgruppen – en användbarhetsexpert. Denna roll ansvarar för användbarheten och genomför aktiviteterna målgruppsanalys, interaktionsdesign och användningstest.
3. Hög ambitionsnivå: På hög ambitionsnivå ingår användbarhetsaktiviteter som en naturlig del i projektet. Utvecklingens fokus är en produkt i användning, aktiviteter för att specificera och bygga en teknisk produkt är steg på vägen. Det finns en roll som har totalansvaret för produktens användningskvalitet, en användningsarkitekt. Tillsammans med systemarkitekten ansvarar hon för produktkvaliteten. Användningsarkitekten hjälper också projektledaren att planera användbarhetsarbetet. På hög ambitionsnivå finns det även andra roller som ansvarar för delar av användbarhetsarbetet: kartläggare, interaktionsdesigner och testledare.

Aktiviteternas omfattning och inriktning styrs av den produkt som ska byggas. Arbets sättet kräver att man i utvecklingsprocessen värderar användningskvalitet minst lika högt som teknisk kvalitet. Detta ställer i sin tur krav på beställarens och projektmedlemmarnas engagemang och kunskap om hur en användningsfokus påverkar utvecklingsprocessen. Än så länge är detta tyvärr ovanligt (Ottersten & Berndtsson, 2002).

Den största fördelen med den höga ambitionsnivån är att designbeslut som fattas baseras på den blivande produktens förväntade nytta. Detta gör att projektets styrbarhet ökar och att den resulterande produkten får hög användningskvalitet. Andra fördelar är att beställarens insyn ökar, att användarnas medverkan hanteras på ett effektivt sätt, och att förankringsprocessen främjas.

### 3.3 Mätning av användbarhet

Det finns åtskilliga metoder för att utvärdera användbarhet. De metoder som anses ha störst praktisk betydelse, som fältstudier, används sällan eftersom de kostar för mycket men de enkla och billiga som heuristikanalys används hellre (Gulliksen och Göransson, 2002), även om den inte ger någon direkt information om de riktiga användarnas interaktion med systemet (eftersom den utförs av användbarhetsexperten eller utvecklaren).

#### 3.3.1 Användbarhetsinspektion

Användbarhetsinspektion är relativt enkel och billig att utföra eftersom den inte involverar riktiga användare, vilket är också dess största nackdel. Heuristiktest innebär att evaluera systemets gränssnitt och funktionalitet enligt vissa tumregler. Kognitiv genomgång innebär att analysera specifika användningsfall och notera fallgropar och felkällor. Dessa metoder är inte riktiga användbarhetstest men kan fungera som en shortcut när man inte har tid eller råd med användbarhetstest (se tabell 1).

Användbarhetsinspektion						
metod	fördelar	nackdelar	svårigheter	Mängd problem	Problem svårighetsgrad	Mängd användare
Heuristik	<input type="checkbox"/> värderar hela designen <input type="checkbox"/> använder beprövade principer	<input type="checkbox"/> inte riktiga användare <input type="checkbox"/> hittar inte okända användares behov	<input type="checkbox"/> testare måste vara kvalificerad <input type="checkbox"/> 3-5 testare	Mer än andra metoder, upp till 90%	<input type="checkbox"/> inte lika svåra som användbarhetstest <input type="checkbox"/> svårare än kognitiv genomgång	inga
Kognitiv genomgång	<input type="checkbox"/> detaljerad analys av specifika uppgifter <input type="checkbox"/> upptäcker orsaker till fel <input type="checkbox"/> lättare än användbarhetstester	inte riktiga användare	<input type="checkbox"/> kan vara tröttsam och för detaljerad <input type="checkbox"/> bättre om gjort i grupp	Omkring 40%	Mindre svåra	inga

**Tabell 1:** Jämförelse mellan de vanligaste metoderna för mätning av användbarhet (Schultz, 2007/06/05)

#### 3.3.2 Riktlinjer för användbarhetsheuristiktest

Dessa riktlinjer kan användas för utformning av gränssnitt och användarheuristiktest (Schultz, 2007/06/05).

- Ta bort onödiga upprepningar, onödig information och element som ej tillhör någon information, ju färre element jag i onödan måste tolka och bedöma desto bättre.
- Utforma ikoner så att de byggs upp av få och väsentliga detaljer. Syftet med ikoner är att skapa omedelbar igenkänning.
- Skapa grupper genom att använda mellanrum. Om det är platsbrist så kan ramar användas. Dock är detta ofta en dålig idé då ramar är extra information på skärmen, vilken användaren automatiskt registrerar och selekterar bort.
- Gruppera saker som hör ihop. Skapa grupper med de fält, knappar, text o s v som hör ihop betydelsemässigt. Placera fält som innehåller liknande information nära varandra. Placera knappar som utför liknande funktioner nära varandra.
- Om färgkodning används: var konsekvent, använd samma färg till samma sak på alla ställen.

### 3.3.3 Riktlinjer för användbarhetsheuristiktest, utvecklat av Nielsen (1993)

Dessa riktlinjer var utvecklade speciellt för läkarstödsystem.

#### *Gör gränssnittet enkelt och naturligt.*

Användargränssnittet borde vara så enkelt som möjligt. Jo fler saker på skärmen, desto mer förvirrande blir det för användaren. Att göra gränssnittet naturligt betyder att det passar användarens förväntningar. Till exempel använd ett gränssnitt som liknar läkarens noteringsblock, med knappar som ser ut som skiljetabbar.

#### *Prata användarens språk*

Många läkare skulle inte tycka att det var bekvämt med termer från systemutveckling eller sjukhusadministration. Skapa menyer som använder handlingsorienterade uttryck på läkarnas yrkesspråk. Till exempel, använd: ”Lista inläggande patienter” i stället för ”Census”.

#### *Minimera användarnas minnesbelastning*

Be inte läkaren att komma ihåg kommandon och procedurer. Använd enkla instruktioner på skärmen. Presentera tillgängliga alternativ i form av menyer. Skapa knappar på skärmen som duplicerar de mest använda kommandon. Använd pop-up hjälp när den är tillgänglig.

#### *Var konsekvent*

Konsekvens i design är en stor del av om man upplever att det är enkelt att använda systemet eller ej. Liknande funktioner på olika skärmar borde vara gjorda på samma sätt. Använd en underliggande generisk struktur för skärmar.

#### *Ge feedback*

När användaren klickar på en knapp eller skriver data i ett fält, ska någonting hända för att bekräfta händelsen. Var säker på att rådgivande meddelande och felmeddelande är tydligt synliga.

#### *Förse användaren med tydligt markerade utgångar*

Nya användare känner sig vilsna i ett hav av fönster. Det hjälper deras orientering om man förser dem med enkla navigeringsvink. Visa tydligt vägen framåt, vägen tillbaka och vägen till basfönstret.

#### *Förse avancerade användare med genvägar*

När man gör systemet enklare för nya användare, gör man det ofta besvärligt för avancerade användare. Gör genvägar för de mest använda funktionerna, men gör dem gömda eller mindre tydliga för nya användare för att förhindra att de snubblar över dem.

#### *Kognitiv genomgång*

Här utför testaren själv de uppgifter som användarna förväntas göra och noterar alla svårigheter och fel. Är mer detaljerad än heuristiktest och avslöjar orsaker till fel. Processen är ganska svår om testaren är ensam eftersom den är detaljerad och tröttsam, därför rekommenderas grupparbete.

#### *3.3.4 Användbarhetstester*

Det finns många typer av användbarhetstester som har gemensamt att testa riktiga användare i mer eller mindre naturlig användningsmiljö (se tabell 2):

- *Tänka högt:* Här observerar man användaren i sitt arbete, användaren instrueras att återge sina tankar under arbetets gång så att det blir möjligt att analysera vilka tankebanor ledde till ett visst fel eller problem. Denna metod är mycket effektiv om användaren känner sig bekväm i situationen. Annars blir användaren reserverad eller motsätter sig testet, vilket kan i värsta fall göra testet omöjligt att utföra.
- *Konstruktiv interaktion:* Här observerar man också användaren i sitt arbete med skillnaden att i stället för att vara passiv observant interagerar man med användaren på ett konstruktivt sätt för att få reda på vilka problem och funderingar uppstår vid användningen. Denna metod är inte lika effektiv som tänka högt men upplevs som mer naturlig av användaren.
- *Instruktionsmetod:* Här interagerar testaren också med användaren men utöver det hjälper han också användaren med de problem han får vid användningen.
- *Observationsmetod:* Användarna observeras i den vanliga arbetssituationen. Observatören håller sig i skymundan och stör inte användaren. Det kan vara svårt att tolka varför användaren fick vissa problem.

Användbarhetstester						
Metod	Fördelar	nackdelar	Problem	Mängd problem	Problem svårighetsgrad	Mängd användare som behövs
Tänka högt (thinking aloud)	<input type="checkbox"/> preciserar användarnas missuppfattningar <input type="checkbox"/> mycket effektiv	Onaturligt för användare	Bör spelas in	Samma som kognitiv genomgång (cognitive walkthrough)	Mycket svåra	3 till 5
Konstruktiv interaktion	Mer naturlig för användare än tänka högt	?	Bör spelas in	Samma som tänka högt	Mycket svåra	3 till 5
Instruktionsmetod	<input type="checkbox"/> Mer naturlig för användare än tänka högt <input type="checkbox"/> stimulerande för tränaren	Visar inte vilka problem användaren får när han använder produkten ensam	En tränare och en observatör behövs	Färre än andra metoder	?	3 till 5
observationer	visar användarnas riktiga uppgifter i naturliga omgivningar	Ingen experimentell kontroll	<input type="checkbox"/> svårt att sätta upp möte <input type="checkbox"/> bör spelas in på video eller noteras	?	svåra	3 eller fler
Fokus grupper	Spontana reaktioner och grupp dynamik	<input type="checkbox"/> svårt att analysera <input type="checkbox"/> låg validitet	svårt att sätta upp möte	Antagligen få	?	6 till 9 per grupp
Frågelistor	<input type="checkbox"/> avslöjar subjektiva preferenser <input type="checkbox"/> kvantitativa resultat	indirekt metod: låg validitet	Användarna måste komma ihåg	Antagligen få	?	Minnst 30

**Tabell 2:** Jämförelse mellan de vanligaste metoderna för mätning av användbarhet (Schultz, 2000)

### 3.4 Användarmedverkan

#### 3.4.1 Definition

Barki and Hartwick (1989) definierar användarmedverkan som ”de diverse aktiviteter som slutanvändarna eller deras representanter utför i systemutvecklingsprocessen”.

Användarmedverkan betyder i denna studie det inflytande användare har i systemets design, antingen direkt genom att delta i utformningen av systemets kravspecifikation och design eller indirekt genom att få tillfälle att uttrycka sig om systemet på möten eller via något media, så som frågeformulär, email eller liknande. Direkt medverkan är en mycket starkare form av användarmedverkan och vi kommer att fokusera på den i denna studie.

#### 3.4.2 Användarmedverkan i systemdesign

Gulliksen (2002) säger att Sverige är det land i världen som alltid haft unika möjligheter och förutsättningar för användarmedverkan. Han säger vidare att användarens inflytande över utvecklingen av sin arbetsmiljö och därigenom sitt IT-stöd är t.o.m. säkerställd i arbetsmiljölagen. Det är dock sällan vi har hört att en användare tillämpat eller åberopat denna lag. Men även om lagen inte praktiskt tillämpas på detta sätt, så är den betydelsefull för att påtala den viktiga rollen användarna bör ha i utvecklingsarbetet.

Sedan handlar det inte bara att ta med användare i utvecklingsprojektet. Det skall på ett effektivt och målmedvetet sätt (Gulliksen, 2002).

En viktig fördel med användarmedverkan på ett tidigt stadium är att användarna får en idé om hur systemet kommer att påverka deras arbete. Detta betyder att de får tid att acceptera införningen och blir på så sätt mer samarbetsvilliga under utvecklingens gång. När användare medverkar vid systemdesign uppstår ett samlärande som utvecklar och berikar alla berörda parter. Detta är en effekt som inte borde underskattas av systemutvecklare, eftersom den ger erfarenhet som höjer deras kompetens. Newman och Noble (1990; i Gallivan och Keil, 2003) argumenterar för vikten av ”tvåvägs lärande” i systemutveckling. För att användarmedverkan ska vara värdefull måste användarna lära av utvecklarna och omvänt. Majzhak och Beath (2000; i Gallivan och Keil, 2003) betonar att användarmedverkan måste leda till förhandlingar, samlärande och kognitivt utarbetande för att ge mervärde till utvecklingsprocessen. Alla verkar eniga om att användarmedverkan är värdefull i egenskap av att den skapar dialog och samlärande, men samtidigt är den inte bättre än kommunikationens kvalitet.

Enligt kommunikationsteori är muntlig kommunikation till stor del guidad av muntlig feedback, deltagarna i kommunikationen tolkar och omtolkar varandras meddelanden hela tiden och nalkas på så sätt varandras mentala representationer (mental representation) (Garrod & Anderson, 1987) eller föreställningar om samtalsämnet. Detta är inte möjligt i skriftlig kommunikation och gör att muntlig kommunikation är effektivare på många sätt, även om den skriftliga har vissa fördelar så som att vara permanent. Även i situationer där muntlig kommunikation är möjlig kan det vara svårt att korrekt återge en annan människas mentala föreställning (ibid). Jo mer komplicerad den mentala bilden blir, desto svårare blir uppgiften.

Kommunikationsteori utgör därför ett resonemang för att användare borde medverka i designfasen. Under designfasen växer systemdesignernas föreställningar om systemet fram på ett konkret sätt vilket ger tillfälle att jämföra dem med användarnas föreställningar och upptäcka missförstånd som uppstått i analysfasen. Det är mycket bättre (och billigare) att upptäcka dem på ett tidigt stadium än att vänta tills produkten presenteras för användartest.

Användarmedverkan är viktig i databas design, den betraktas som en fundamental premis för information engineering (Marenghi, 1982). Enligt den kända databasexperten James Martin är den viktigaste egenskapen hos ett databasverktyg att man kan uppnå snabba resultat (ibid.). Detta är förståeligt med tanke på vikten av användarmedverkan, som kräver en snabb och enkel designprocess.

### 3.4.3 Maktförhållanden

Ett hierarkiskt förhållande mellan gruppmedlemmar förhindrar konflikter eftersom det inte finns några tveksamheter om vem som ska bestämma. Detta betyder dock inte att hierarkisk maktordning är idealisk eller ger bäst resultat. Enligt konfliktmodeller av användarmedverkan leder användarmakt till konflikter men, om rätt hanterat, också till lösning av konflikter (Gallivan & Keil, 2003). Denna process kan öka systemets användbarhet, eftersom utvecklarna inte kan välja bort de problem som de tycker är mindre viktiga eller svåra. Konflikter kan därför vara bra för systemets användbarhet om de är lösta på ett tillfredsställande sätt, däremot försämrar olösta konflikter användbarheten. Makt asymmetri, där endast utvecklarna har makten att besluta, kan leda till missnöje hos användarna och kommunikationsbrister (DeBrabander & Thiers, 1984; i Gallivan och Keil, 2003) vilket minskar sannolikheten av att konflikter löses på ett lyckligt sätt.



Hunton och Beeler (1997) gjorde en studie där de jämför projekt där användare har rådgivande roll vs. ”instrumentell roll” dvs. riktig makt att påverka. Resultatet blev att användare var nöjdare med systemet när de hade haft en instrumentell roll.

Användare bör alltså ha riktigt inflytande på systemets utveckling för att deras medverkan ska öka användbarheten. Detta även om användarinflytande ökar chansen för konflikter i samarbetet. Det är naturligt för de flesta människor att undvika konflikter, det är en av orsaken till grupp tänkande (Hägerfors, 1995), vilket försämrar nyttan av grupparbete. Är det möjligt att utvecklare undviker att ge användare riktigt inflytande i systemutvecklingsprojekt därför att de vill undvika konflikter?

#### 3.4.4 Kommunikation

De flesta forskningsmodeller av användarmedverkan, förutsätter att den kommunikation mellan användare och utvecklare, som är nödvändig för en klar kravspecifikation, äger rum i systemutvecklingen. De antar ofta att denna kommunikation garanterar systemets användbarhet och användarnas acceptans av systemet. Men det räcker inte enbart med kommunikation, kommunikationen måste hålla en viss kvalitet också. Det finns många förhållanden som kan orsaka att kommunikationen har dålig kvalitet, så som motivation och maktförhållanden. Det finns exempel om systemutvecklingsprojekt där man har gjort allting enligt modellen, man har gjort allt för att säkerställa hög användarmedverkan men ändå har projektet inte fått användarnas acceptans pga. ineffektiv kommunikation (Gallivan & Keil, 2003). Gallivan och Keil (2003) har analyserat förhållandena i detta och andra fall utifrån två forskningsrön, användarmedverkan och samverkan i projektgrupp. Deras resultat tyder på att projektledare och systemutvecklare måste se bortom den information som användare förmedlar. De bör också tänka på den information som de inte förmedlar och försöka skapa förhållanden där de känner sig fria att dela sina åsikter. Bra kommunikation kan inte äga rum om användarna känner att negativa åsikter inte är välkomna eller att ingen hänsyn tas till dem.

Gallivan och Keil (2003) har identifierat kommunikationsbrister som kan förekomma på de fyra stegen i deras kommunikationsmodell.

- Steg 1. Användare blir medvetna om ett meddelande de vill förmedla till utvecklarna
- Steg 2. Användare skickar meddelande till utvecklarna
- Steg 3. Utvecklare tar emot och tolkar meddelande
- Steg 4. Utvecklare prioriterar och agerar

Om ett av dessa steg sviker blir meddelandet förvridet eller kommer inte vidare till nästa steg.

De analyserar ett systemutvecklingsfall, sälj- och lagersystemet CONFIG, där alla stegen sviktade trots stor användarmedverkan och systemet blev ett flopp. De visar att användarmedverkan bara förbättrar användbarhet om kommunikationen mellan utvecklare och användare har en bra kvalitet (ibid.).

En vanlig kommunikationsmodell är *processmodellen* som betonar att kommunikation är en tvåvägsprocess där varje person fungerar både som sändare och mottagare (Fiske, 1982; i Hägerfors 1995). Budskapen framförs via en kanal och kan påverkas av störningar. Ord är bara en del av kommunikationen, känslor och relationer kommuniceras huvudsakligen med icke-verbala signaler. Riskerna för missförstånd är stora. För att minska riskerna bör man eftersträva en öppen kommunikation som är jämlik, problemorienterad och spontan (Hägerfors, 1995).

### 3.4.5 *Motivation*

Motivation är en viktig faktor som påverkar kvaliteten av användarmedverkan. Viktiga framgångsfaktorer är:

- Frivillig deltagande
- Användare ser direkt hur deras förslag påverkar systemet
- Användare är kompenserade på något sätt för sin insats

Det är bättre att användare fortsätter att arbeta inom sitt vanliga område och kan se sina idéer förverkligas i produkten under tiden den används. Även om frivillig deltagande är viktig är det bra om användarna är kompenserade på något sätt till exempel i form av högre lön, resor eller bättre position. Annars kan det bli problem med motivation längre fram (Gulliksen m.fl., 1999).

### 3.5 Användarcentrerad design

Enligt ISO/DIS 13407 (1998) internationell standard är användarcentrerad design en syn på mjukvaru- och hårdvarudesign som baseras på fyra grundprinciper:

1. En ändamålsenlig distribuering av funktioner mellan användare och system
2. Aktiv involvering av användare
3. Iteration av design lösningar
4. medlemmar i design team har olika akademiska kompetenser (tvärdisciplinär design)

Enligt dessa principer bygger användarcentrerad ansats på ett iterativt arbetssätt. I ett iterativt arbetssätt skapas prototyper med hänsyn till krav som har lagts upp i medverkan med användarna. I varje iteration testas användarna prototypen och på detta sätt tillåts kunskapen om de verkliga behoven hos användarna att växa fram och förändras under processen. Fokus är hela tiden på användarna och deras arbetsuppgifter.

För att iterativt arbete ska vara möjligt måste utvecklare tillsammans med användare identifiera nödvändiga förändringar, men det måste också finnas möjlighet och vilja att faktiskt göra dessa förändringar. Annars rör det sig inte om ett iterativt arbetssätt (Gould m.fl., 1997).

Gulliksen (2002) menar att man måste uppnå vissa minimikrav för att man verkligen skall kunna kalla den utveckling man bedriver för iterativ:

- En ordentlig analys av användarens krav och användningssammanhanget,
- en prototyputförningsfas
- en dokumenterad utvärdering av prototypens användbarhet som måste resultera i ett medvetet beslut om förändringar som kan påverka den fortsatta prototyputförningen.

Enligt Gulliksen (2002) är användarcentrerad systemdesign: *en process som fokuserar på användare och användbarhet genom hela utvecklingsprocessen och vidare genom hela livscykeln*. I den användarcentrerade designen reflekterar systemutvecklarna med användarna snarare än att teoretisera om användaren (Gulliksen, 2002). Systemutvecklarna reflekterar över sitt eget arbete med användaren.

Användaren ses som en konversationspartner.

Systemutvecklarna kan genom att reflektera över sitt arbete hitta ny kunskap och ofta se användningsområden för sin design som den inte ursprungligen var tänkt för.

#### 3.5.1 Användbarhetsdesign

Användbarhetsdesign är en användarcentrerad designmetod som handlar om att integrera användarcentrerad design i utvecklingsprocessen (Gulliksen och Göranson, 2003). Gulliksen (2002; s.155) definierar användbarhetsdesign som:

*Ett användarcentrerat tillvägångssätt att utveckla användbara interaktiva system på genom att kombinera usability engineering med interaktionsdesign, samt att ha ett långtgående och aktivt användardeltagande i den iterativa processen.*

Denna definition säger endast att användbarhetsdesign är en metod för att utföra användarcentrerad design. Gould et al. (1997) använde uttrycket användbarhetsdesign när de beskrev sina principer för användarcentrerad design. Det är därför lite luddiga gränser mellan dessa två. Dock är användbarhetsdesign här använt som en benämning på en viss designmetod skapad av

Gulliksen (2002) medan användarcenterad design är ett paraplyuttryck för de metoder som baseras på de 4 principer som står i sektionen ovan.

Användbarhetsdesign baseras på följande nyckelprinciper (Gulliksen, 2002):

- *Användarfokus – verksamhetens mål, användarens arbetsuppgifter och behov skall tidigt vara vägledande i utvecklingen.* Genom samförstånd för verksamhetens mål, grunden i användarens situation, dess arbetsuppgifter, varför och hur han utför sina uppgifter, etc. genom detta blir det viktigt att man prioriterar vad som är bra för användaren framför vad som är tekniskt möjligt. Aktiviteter som uppgiftsanalys måste vara en naturlig del av utvecklingsprocessen.
- *Aktiv användarmedverkan i utvecklingen - användaren skall tidigt och kontinuerligt medverka genom hela systemets livscykel.* Möta användaren i dess egen naturliga arbetsmiljö. Identifiera lämpliga faser för användarens deltagande och en tydlig beskrivning av hur detta skall ske, t ex analys, design, utvärdering och konstruktion.
- *Evolutionär utveckling – systemet skall utvecklas iterativt.* Designlösningarna bör kontinuerligt itereras med användaren.
- *Gemensam och delad förståelse – designen skall dokumenteras med en för alla inblandade parter enkelt förståelig representation.* Använd en för användaren bekant terminologi. Så långt som möjligt bör konkreta designrepresentationer som prototyper, alltifrån enkla skisser till mer avancerade mock-up:er, och simuleringar användas för att åskådliggöra den framtida användningssituationen. Abstrakta notationer som användningsfall, UML-diagram eller kravlistor fyller givetvis sin funktion, med leder sällan till en för användaren konkret bild och förståelse över hela den framtida användningssituationen med det nya IT-systemet.
- *Prototyping – tidigt och kontinuerligt skall prototyper användas för att visualisera och utvärdera idéer och designlösningar.* Använd prototyper för att stödja den kreativa processen, att visualisera idéer och lösningar. Börja på låg nivå med t ex pappersskisser innan något över huvudtaget kodas. Arbeta med prototyper tillsammans med användaren i dess egen arbetsmiljö – s.k. kontextuell prototyping. Börja med den konceptuella designen på hög nivå och gå inte ner i detaljnivå för tidigt. Om det är möjligt, arbeta med flera prototyper samtidigt.
- *Utvärdera verklig användning – mätbara mål för användbarheten och kriterier för designen skall så långt som möjligt styra utvecklingen.* Utvärdera designen gentemot användbarhetsmålen och designkriterierna tillsammans med användaren genom att mäta användarens reaktioner på de olika prototyperna.
- *Tvårdisciplinära team – utvecklingen skall utföras av effektiva team med en bredd av kompetenser.* Olika kompetenser bidrar till helheten, t ex programmerare, interaktionsdesigners, experter på fältstudier, etc. Detta för att man bättre täcker alla aspekter i utvecklingsprocessen och hjälpa teammedlemmarna med ett arbetssätt som gör dem effektiva.

Användbarhetsdesign följer Gould et al (1997) grundprinciper som är en förutsättning för att utveckla system med god användbarhet:

- Tidig och kontinuerlig fokus på användarna.
- Empiriska mätningar.
- Iterativ design.
- Integrerad design.

I användbarhetsdesign ska representativa användare aktivt medverka, tidigt och kontinuerligt genom hela systemets livscykel (Gulliksen, 2002). Användarna bör involveras direkt i utvecklingsprojektet men även utanför själva projektets ramar.

Utvecklarna måste först ta reda på användarens kognitiva, beteendemässiga och attitydmässiga inställning för att förstå användaren. Baserat på detta kan man sedan förstå vad användaren behöver för system. Att dessutom göra användaren till medlem i designteamet genom deltagande design eller som konsult är viktigt (Gould et al, 1997).

Tidigt i utvecklingsprocessen bör man observera och mäta den tilltänkta användarens reaktioner på skrivna scenarior och enkla skisser. Senare i processen bör man använda simuleringar och prototyper för att utföra verkligt arbete. Då bör man observera, samla in data om och analysera hans beteende och reaktioner.

Processen måste vara iterativ: det måste finnas en cyklisk process av design, utvärdering och omdesign, etc. som skall upprepas så ofta som det är möjligt. Empirisk mätning och iterativ design är nödvändiga eftersom designer, oavsett hur duktiga de är, inte kan komma fram till den rätta lösningen första gången.

Alla komponenter som bidrar till användbarhet måste utvecklas tillsammans. Integrerad design förutsätter att man initialt fastställer att användbarhet är viktigt, att det är komplext och att det kräver insatser redan från första början.

Grundtanken med användbarhetsdesign är att, eftersom system är verktyg som ska användas av användare, kan design inte stå ensamt utan användbarhet. En designers skicklighet räcker inte i sig, utan den måste kombineras med en process för användbarhet. Denna process kombineras med interaktionsdesign, som i systemutvecklingssammanhang handlar om navigering, representationer och informationsstrukturer i användargränssnittet. Användbarhetsdesign fokuserar dock mer på resultatet – designlösningen, än vad som är brukligt inom exempelvis usability engineering, men samtidigt betonas användbarhet och människa-dator-interaktion (MDI).

## 4 Empiri –Blodaccessregister

### 4.1 Bakgrund

Våren 2003 började njuravdelningen på Universitetssjukhuset i Lund (USiL) samla information om sina dialyspatienters blodaccesser. En förutsättning för att kunna utföra en bloddialys är att ha en kontinuerlig tillgång till stora kärl med högt blodflöde. Detta kräver en antingen en kateter läggs i centrala kärl eller att en artär och ven är knytt ihop för att bilda en s.k. fistel. . Det är viktigt att hålla register över dessa blodaccesser i kvalitetshöjande syfte. Att kunna få statistik över det som ökar resp minskar en blodaccess ”överlevnad” kan hjälpa till att förbättra behandling i framtiden. En läkare på avdelningen gjorde en databas i Filemaker, med ett tillhörande gränssnitt. Hösten 2004 ombads författarna flytta databasen med blodaccessdata från Filemaker till Microsoft Access. De ombads också normalisera tabellerna och skapa ett gränssnitt som skulle kunna användas av beställaren i första hand men eventuellt av annan personal framöver.

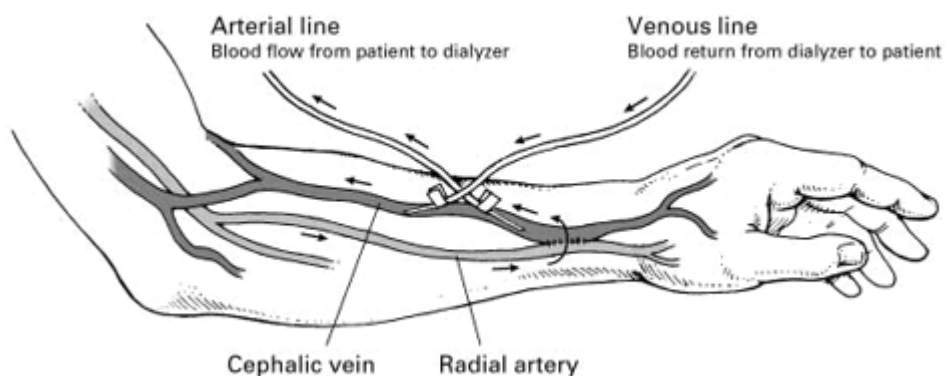
Målet var att skapa ett användbart system på avdelningsnivå som uppfyllde behov för datainsamling och statistisk analys såväl som ge njurspecialisterna överblick över sina patienters blodaccesser, vilket behövdes eftersom dessa kan bli många över åren med många problem och ingrepp. USiL har ett datoriserat journalsystem, Melior, men det är inte anpassat för att hålla reda på denna information.

### 4.2 Minivärlden

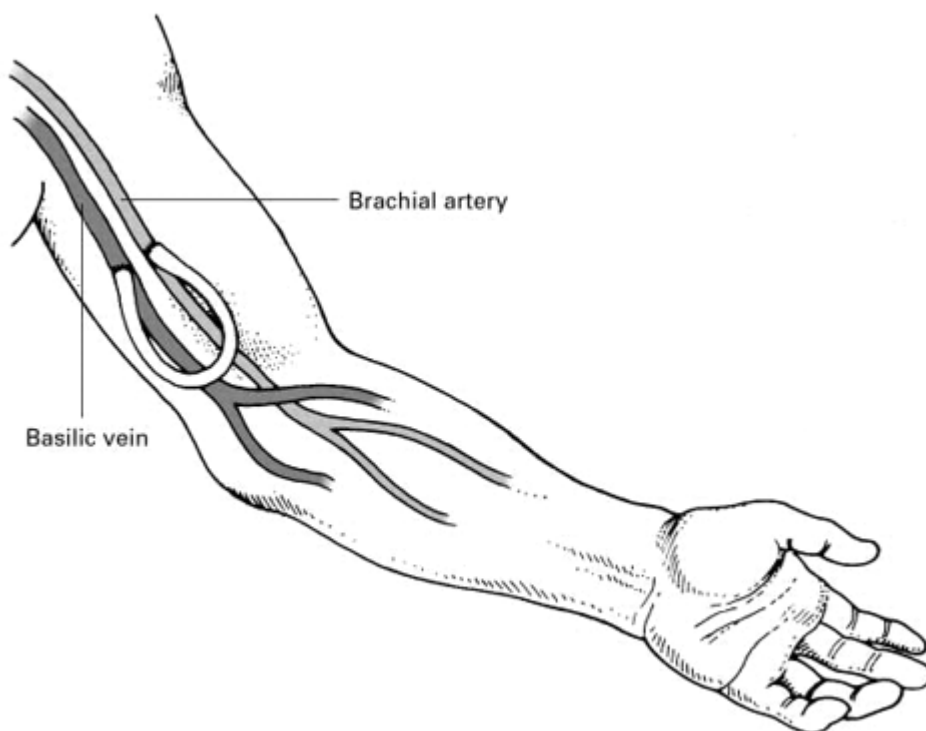
Njurarna är livsviktiga organ som bland annat rensar blodet. Patienter utan njurfunktion behöver livsuppehållande dialysbehandling om de inte får njurtransplantat. Det finns två typer av dialys, bloddialys (HD) och s.k. peritoneal dialys (PD). Vid PD använder man patientens egen bukhinna för att rensa blodet med att sätta inn ren vätska, innehållande salter och glukos, i bukhållan och drenera sedan ”smutsig” vätska ut. PD patienter ingår inte i denna uppsats därför att de inte behöver en blodaccess. Vid hemodialys rensas blodet utanför kroppen i en dialysmaskin (konstgjord njure). Blodet förs ut ur kroppen via blodaccessen och förs genom en filter i dialysmaskinen för att sedan återföras ”rensat” till kroppen.. En vanlig behandling för bloddialys utförs tre till fyra gånger i veckan och varje dialysbehandling tar omkring fyra timmar, men det finns många variationer av det schema. Blodaccesser kan vara av några olika typer. Vanligast och bäst är att koppla samman en artär och en ven, en s.k. fistel (fig 4.1). På det sättet utsätts venen för mycket övertryck och så småningom blir större och mångdubblar sitt blodflöde. Sedan kan nålar stickas i denna övervuxna kärl för att dra blod till dialysenapparaten. Ibland är kärlet så små att det inte går att knyta ihop en artär och ven direkt. Då kan man koppla dessa samman med en konstgjord slang (oftast goretex) och kallas detta för en graft-fistel (fig 4.2). Dessa artär-ven kopplingar kan man sedan placera på olika ställen i kroppen men vanligast är på underarm (bäst) eller överarm (näst bäst) Ett annat alternativ till en artär-ven koppling är att lägga en kateter i en central ven, antingen till längre tid (permanent) eller temporärt. Vanligast är att lägga katetern i stora kärl på halsen eller i lumsken (femorals kateter). Att lägga en kateter är mycket sämre alternativ än fistel eller fistel graft pga infektionsrisken och skadan som slangen gör mot kärlväggen (orsakar förträngningar). Därför är det viktigt att hitta de faktorer och arbetssätt som leder till att dessa används bara i undantagsfall (Asgeirsson, 2004).

Att ha en blodaccess är aldeles naturligt och det är ofta problem med dem. Fistlar kan sluta att fungera pga en propp. Det går ibland att reparera operativt men ibland måste man lägga ner fisteln och inoperera en annan. Proppbildning förekommer också hos katetrar men är ovanligare

och enklare att åtgärda. Vanligare är än långsam funktionsnedsättning som oftast beror på en förträngning någonstans. Om man märker det i tid är det ofta möjligt göra ett ingrepp där man rensar och reparerar fisteln och kan på så sätt undvika nedläggning. Situationen är liknande för andra typer av blodaccesser. Det är därför av stor vikt att övervaka blodaccessens funktion (ibid.). Infektioner är också vanligt förekommande och är mer frekventa där det finns konstgjort materiell i kroppen som vid grafter och katetrar. Det finns alltid risk för blödning som är en mycket dramatisk händelse pga. det höga blodflödet i accesserna men det händer ytterst sällan. Alla dessa problem härstammar från riskfaktorer, antingen hos patienten eller hos vårdkedjan (procedural risk). Det yttersta syftet med att samla statistik över blodaccesser med relaterad information är att kunna göra en faktoranalys för att leta upp påverkbara faktorer som kan förbättra blodaccess management (Asgeirsson, 2004).



Figur 4.1 En fistel (från Ifudu, 1998)



Figur 4.2 En graft (från Ifudu, 1998)

Svikande av blodaccesser är en betydande orsak till morbiditet (sjukdom) och sjukhusvistelser hos hemodialyspatienter (Mayers m.fl., 1992). Därför kräver vården av hemodialysberoende patienter konstant uppmärksamhet för att bibehålla blodaccessernas effektivitet och därmed minska morbiditeten. National Kidney Foundation (2000) rekommenderar ett program för att försäkra kvaliteten. Syftet är att upptäcka hotade blodaccesser, spåra implikationsfrekvens och implementera procedurer som maximerar blodaccessernas livslängd. För att kunna göra detta är det nödvändigt att samla in relevanta data, men det har hittills inte gjorts konsekvent på USiL.

### 4.3 Utvecklingsverktyg

Valet av utvecklingsverktyg var problematiskt. I början ville utvecklarna själva helst programmera i C# i utvecklingsmiljön Microsoft Visual Studio med SQL Server som underliggande databas. De ansåg att detta skulle resultera i en professionell och anpassningsbar produkt. Problemet var att beställaren ville vara med i alla faser av projektet eftersom han ansåg att han skulle själv behöva sköta systemet när det var färdigt. Han tyckte att det skulle vara jobbigt att behöva förlita sig på utvecklarna för att förbättra systemet eller hantera fel. Dessutom var han intresserad av att lära sig programmera. Han ville göra systemet i MS Access med vilken han redan var förtrolig och kunde programmera lite. Han trodde inte att han skulle hinna lära sig ett mer komplicerat utvecklingsverktyg som Visual Studio. Det fanns också den fördelen att utvecklingen skulle ta kortare tid att utveckla formulären, för MS Access sköter självt alla kopplingar mellan formulär och databas. Man kan därför snabbt skapa fungerande formulär och testa olika versioner. Utvecklarna accepterade eftersom de i sin tur hade ont om tid. En avgörande faktor var att utvecklarna ville att utvecklingen av systemet skulle ske i ett så nära samarbete med användaren som möjligt. Anledningen till detta var att de var oerfarna som utvecklare samt att detta skulle underlätta för dem att erhålla en djupare förståelse för minivärlden vilken systemet skulle stödja. På så sätt skulle man säkra en utveckling av ett användbart och därmed lyckat system. Det var därför lämpligt att utveckla systemet i en miljö med vilken användaren redan var förtrolig med. Enligt Alice Balter (2003) är MS Access ett lämpligt verktyg för utveckling av avdelningsspecifika system i stora organisationer. Om datamängden med tiden växer sig för stor för MS Access är det relativt lätt att flytta databasen till MS SQL Server men samtidigt behålla gränssnittet i MS Access (Balter, 2003).

### 4.4 Utvecklingsmetod

Vid design av fisteldatabasen var det av största vikt att hålla felnivån nere. Dessa data skulle användas i vetenskapligt syfte och möjligen så småningom i medicinskt syfte i framtiden. En annan angelägenhet var att inlärningstiden blev så kort som möjligt. I dialysverksamheten är det ibland stor omsättning i personalen så att nya användare måste hela tiden sättas in i produkten. Samtidigt är det en hektisk miljö där det inte ges tid för en lång inlärningsprocess. Därför var ett intuitivt gränssnitt som var väl anpassat till arbetsflödet av stor vikt. Som skildrat ovan ger användbarhetsdesign bra förutsättningar för att uppfylla dessa krav.

Det var klart från starten att beställaren skulle delta i utvecklingen eftersom han besatt den nödvändiga kunskapen om informationsbehovet och hade dessutom hand om databasen i dess dåvarande form. Beställaren representerade den enda användaren av systemet (i alla fall till en början). Detta satte krav på utvecklarna att hitta en metod och samarbetsform som var lämpligt för nära samarbete med användaren som dessutom var beställare.



Efter noggrann genomgång av olika systemutvecklingsmetoder i litteraturen kom vi fram till att användbarhetsdesign uppfyllde våra krav om stor användarmedverkan och objektiva mätningar av användbarhet som påverkade designprocessen från ett tidigt stadium.

Användarinspektion och användartest utfördes i varje iteration, så som designmetoden föreskriver. Den mätta användbarheten jämfördes över tid och mellan faserna med hög respektive låg användarmedverkan i designfasen. Eftersom testerna var informella och dåligt dokumenterade, fanns det inget objektivt sätt att tolka dem, utan man fick förlita sig på testarnas åsikter. Detta gjorde att jämförelsen över tid var för osäker för att diskuteras här men det var en väldigt stor skillnad i testresultaten mellan projektfaserna med hög respektive låg användarmedverkan. Denna skillnad ansågs vara sann av alla projektdeltagare. De sista testerna var däremot väldokumenterade och presenteras i nedanstående avsnitt, (4.11).

Nackdelen med användbarhetsdesign var att det inte fanns några tydliga riktlinjer om hur man ska gå till väga när beställaren och användaren är en och samma man. Gulliksen och Göranson (2002) säger att om användarmedverkan ska ge en god effekt för användbarheten, måste någon av användarna ha beställarkompetens, vilket de definierar som ”förmågan att planera, formulera, kommunicera och övervaka en systemupphandling och systemutvecklingsprojekt utifrån ett perspektiv som ser verksamheten i termer av aktivitetsnivå och utvecklingspotential”. Det borde därför vara fördelaktigt att användaren också är beställare.

Projektet pågick under sju veckor våren 2004. Den första veckan användes för gränssnittsdesign och datavättning. Den andra veckan påbörjades gränssnittsdesignen. Användaren medverkade i gränssnittsdesignen under 3 veckor men då ändrades metoden till att användaren deltog i den iterativa utvärderingen och analysen, men medverkade inte i designfasen.

Detta upplägg gjorde utvecklarna eftersom användaren har den bästa kunskapen om minivärlden och kunde med sina åsikter guida dem i designarbetet. Utvecklarna ville ha hans objektiva åsikter som en representativ användare och inte en professionell projektmedlem, som lätt blir en försvarare av en viss designlösning som han varit med och designat.

De tänkte att om användaren istället fortsatte aktivt delta i själva designarbetet skulle han inte kunna vara lika objektiv i sin utvärdering av designlösningarna.

## 4.5 Projektgruppen

Användbarhet ingick som en integrerad del i projektet och utvecklarna hade en hög ambitionsnivå i användbarhetsarbetet enligt Otterstens och Berndtsons (Ottersten & Berndtsson, 2002) kriterier (se avsnitt 3.2.3). Under hela utvecklingsprocessen arbetade utvecklarna med användbarhetsaktiviteter. De delade upp roller med ansvar för användbarheten. Då de endast var tre inom projektet kunde en person inneha olika roller samtidigt som en roll kunde bemannas av båda utvecklarna.

Projektgruppen bestämde att den person som ansvarade för kartläggningen också hade ansvar för användbarheten (användbarhetsdesign). De andra fick då inneha ansvar för interaktionsdesignen. De gjorde denna uppläggning då kartläggaren ofta får en bild av en tänkt lösning mycket tidigt under sitt arbete med kartläggning och kan ha svårt för att skapa nya lösningar, detta också med tanke på att användaren redan hade en representation av det tänkta systemet innan projektet hade startat, samtidigt som minivärlden var honom närmare.

De kom fram till att användaren var med vid design, implementering och test. Vin hade ansvar för design, programmering och test medan Danisa fick ansvar för insamling av information, kartläggning och dokumentering av projektet. Dessa roller var mycket flexibla, eftersom de var så få behövde de ofta ställa upp för varandra.

De hade svårt att välja projektledare och tyckte dessutom att de var alla lika ansvariga för projektet. De valde ingen formell projektledare men användaren blev så småningom informell projektledare med överordnat ansvar för projektet.

## 4.6 Kravanalysen

Till en början gjorde utvecklarna en användaranalys av deras förste användare. Tanken var att analysen skulle utökas till att så småningom även omfatta andra användare, andra njurspecialister, sjuksköterskor etc.

### 4.6.1 Användaranalys

Användaren är en blivande njurspecialist. Han jobbar kliniskt med dialyspatienter men använder en del ut av året (fyra månader) till forskning om njurfunktion. Han är en doktorand med bra koll på informationsteknologi och använder mycket Microsoft Office verktyg i sitt arbete, som MS Excel och MS Access. Med dessa förutsättningar hade han inga större svårigheter med att sätta in sig i systemutvecklingen.

Användarens motiv för att delta i/beställa projektet var att han hade åtagit sig att underhålla en databas över blodaccesser på njuravdelningen vid USiL. Denna databas behövde förbättras och han önskade även ett välfungerande gränssnitt till databasen via vilket man kunde få statistik om fistelverksamheten.

Sedan gjorde man en målanalys. Eftersom verksamheten inte hade beställt detta system blev det svårt att tala om något verksamhetsmål. De bestämde därför att lägga ihop avsnitten om verksamhetens och användarens mål.

### 4.6.2 Mål 1: Statistik om fistelöverlevnad

Statistik om fistel-överlevnad var det primära syftet med projekt. Därför måste databasen innehålla alla data som kan tänkas vara relevanta till fistel överlevnad. Eftersom användaren skulle delta i designen av databasen tillsammans med systemutvecklarna behövdes på detta stadium endast en lista över den viktigaste informationen som skulle kunna nås ut. Dataredundans skulle vara minimerad, den redundans som anses vara nödvändig skulle vara noga kontrollerad. Det skulle finnas regler om vilka data som kan matas in för att skydda dataintegriteten, i den mån det var möjligt.

Den huvudsakliga statistiken projektgruppen ville få ut var blodaccessernas livslängd. Denna information ska kunna korreleras med ett antal parametrar, även göra en faktoranalys eller ANOVA. Dessa parametrar ska hämtas från Melior journalssystemet.

Lista över information om fistelöverlevnad som man ska kunna få från systemet:

- Antal patienter
  - Grupperad efter varifrån de kommer, d v s om de kommer utifrån (vanligast femoraliskateter) eller om de är dialyspatienter.

- Access historik
  - Tid från access till dialys start
  - Access överlevnad mellan ingrepp
  - Medelantal ingrepp på access som har levt under en bestämd tid
  - Medelantal ingrepp på accesser efter hur länge de har levt
- Komplikationer
  - Efter typ av access
  - Korrelerad till diagnos, Hb, antikoagulation, etc.
  - Infektioner
  - Vid inläggning

#### 4.6.3 Mål 2: Användbarhet i verksamheten

Det sekundära syftet med systemet är att läkare ska kunna ha överblick över fistlar och ingrepp hos sina patienter. Detta kan uppnås genom att information matas regelbundet in i systemet och för att detta skall vara möjligt krävs det att användarna accepterar att införa denna arbetsuppgift i deras dagliga arbetsrutin. Systemet ska vara användbart i den mening som tidigare har skildrats i uppsatsen d v s det ska hjälpa läkarna att ha överblick över fistlar och ingrepp, det ska öka läkarnas arbets kvalitet och effektivitet och det ska kännas bra och meningsfullt att använda systemet.

Lista över information om blodaccesser som är viktig i sammanhanget:

- All statistik ska kunna tas ut för en vald tidsperiod
- Kunna få upp en lista över kroniska patienter vid en bestämd tidpunkt
- Kunna få upp historik för en patient i tidsordning. Detta kräver tillgång till patientens journal i Melior.
- Kunna få upp historik för bestämda händelser hos en viss patient i tidsordning. Detta kräver tillgång till patientens journal i Melior.

Målet är att det ska finnas data om alla nuvarande och nya patienters fistlar i databasen.

#### 4.6.4 Mål 3: Användbarhet i organisation av verksamheten

Ytterligare ett skäl med detta system är att få ut data om accessverksamheten. Detta är av speciellt intresse för verksamheten eftersom det kan utgöra underlag för kostnadsbedömning, bemanning och annan organisationsverksamhet.

Lista över information om blodaccesser som är viktig i sammanhanget:

- Hur många olika typer av access är lagda per tidsperiod
  - Behövs då att sammanfoga alla olika typer
- Antal insatser (femoraliskatetrar) grupperad efter operatör

#### 4.6.5 Säkerhet

Databasen måste ha bra säkerhet för att skydda patienternas identitet. Den ska ha säkerhet på användarnivå, d v s varje användare ska ha eget användarnamn och lösenord.

#### 4.6.6 Uppgiftsanalys

Här presenteras information som samlades in via de olika metoder som skildrades i avsnitt 4.4. Blodaccessverksamheten har redan presenterats. Här går vi djupare i arbetsprocesserna som ska stödjas av systemet.

Användarens uppgifter delas i tre områden som är kopplade till projektets syfte:

1. Att mata in data (också ändra eller ta bort felaktiga data)
  - a. Att mata in en ny patient (oftast i samband med att lägga till en blodaccess)
  - b. Att lägga till en blodaccess
  - c. Att lägga till ett ingrepp
  
2. Få information för användning i verksamheten (behandling av individuella patienter)
  - a. Få information om var tidigare accesser har lagts
  - b. Information om ingrepp som gjorts
  - c. Information om de problem som uppstått i tidigare accesser
  - d. Information om problem som uppstått vid tidigare ingrepp
  
3. Få information för kvalitetsanalys och förbättring av behandlingsstrategi
  - a. Blodaccesser
    - i. Medellivslängd hos blodaccesser, uppdelat efter typ av access
    - ii. Ingreppsfrekvens efter typ av access
    - iii. Hur många blodaccesser av varje typ har lagts per år
    - iv. Hur många ingrepp har gjorts per typ av access
    - v. Hur är trenderna i val av blodaccess
  
  - b. Ingrepp
    - i. Hur stor del av ingrepp får ett bra resultat
    - ii. Visa relativt resultat per
      1. Operationsläkare
      2. Blodaccesstyp
      3. Patientens ålder
      4. Patientens diagnos

#### 4.6.7 Arbetet med kravanalysen

Projektgruppen påbörjade arbetet med att ta fram en noggrann kravspecifikation med användbarhetsmål. Tanken var att göra detta i ett tidigt skede i projektet och i ett senare skede lägga fram ett designförslag för godkännande.

I kravspecifikationen försökte de tydligt definiera vad det var de ville ha för att undvika onödiga konflikter när systemet var färdigt.

De ville gärna få en så bra bild som möjligt av användarna innan utvecklingsarbetet påbörjades och hade tänkt genomföra några ostrukturerade intervjuer. Då projektet inte var ”officiellt” var

det i stort sett omöjligt för dem att genomföra detta. De fick förlita sig på beställarens fakta om de ”framtida tänkta” användarna. Det var därför svårt för dem att veta om de fattade felaktiga designbeslut.

I övrigt var användaren engagerad och hade tydliga åsikter om kraven. Kraven var i första hand kopplade till hans roll som beställare och inga detaljerade krav som gällde användningen förutom generella krav om användbarhet. Projektgruppen kunde inte skilja verksamhetskraven och användarens krav åt eftersom dessa verkade vara de samma: en viss information skulle hämtas ur systemet för att läkaren skulle kunna uppfylla sin roll som läkare, men det var också i verksamhetens intresse. Det fanns på detta stadium inga intressekonflikter mellan läkarrollen och verksamheten.

Uppgiftsanalysen växte fram i det iterativa arbetet med analys, design, utvärdering och implementering. De gjorde därför inte en inledande uppgiftsanalys för hela systemet utan gjorde den efterhand som designarbetet fortskred. I början avsatte de tid för analys men det blev snart klart att det var mycket effektivare att integrera den i designfasen. Det blev alltså ingen formell tid avsatt för analys, utan det ingick i designprocessen att ta en stund i analys när man började designa en ny funktion eller det kom upp frågor eller funderingar. Den växte alltså fram ur kommunikationen på ett naturligt sätt. Det fanns oftast en ansvarig i gruppen som noterade allting som vidrörde kravanalysen och på så sätt växte det fram en kravanalys. Projektdeltagarna tyckte alla att det var ett mycket bra sätt att jobba.

När de bytte designmetod kunde de inte längre diskutera med användaren under designprocessen. De flyttade då analysfasen till slutet av utvärderingsfasen. Detta var inte lyckat, analysfasen blev nästan helt slopad, enligt deltagarna, efter metodbytet. De ansåg att detta berodde på att samarbetet gick mycket sämre efter metodbytet, speciellt utvärderingsfasen. De var därför irriterade innan analysen började vilket de tyckte påverkade deras motivation. De tyckte också att de inte visste vad de skulle prata om under de iterativa analysessionerna. Danisa sa: ”Vi borde ha varit duktigare att notera frågor”.

## 4.7 Designprocessen

### 4.7.1 Databasen

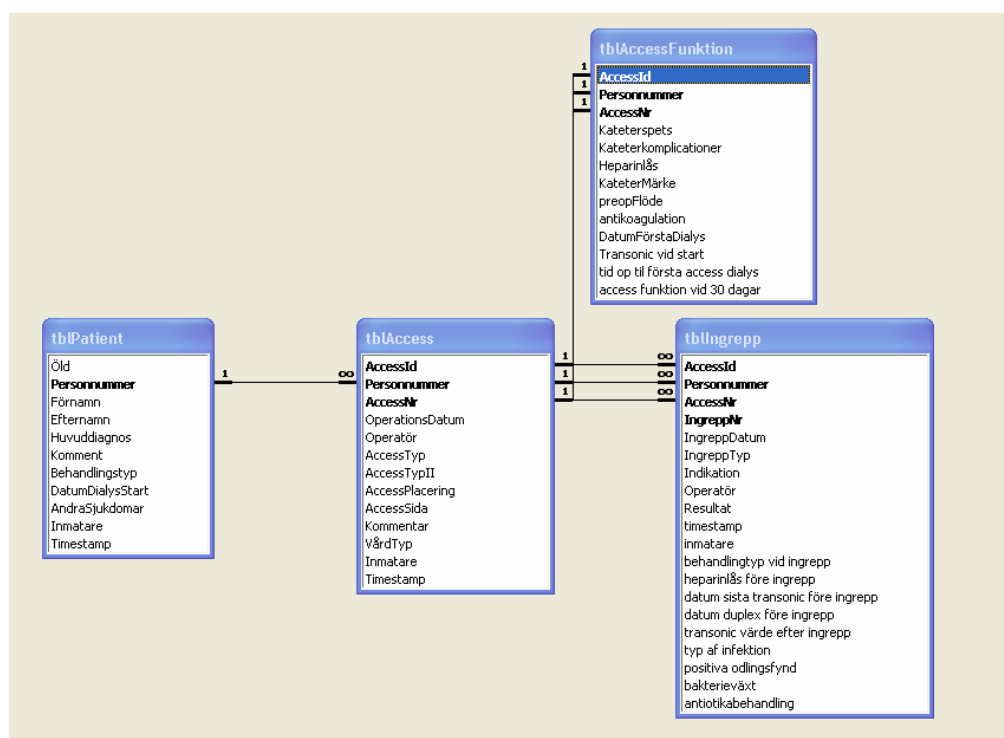
Det var ett problem med valet av designmetod, eftersom användbarhetsdesign fokuserar på utformning av gränssnitt. Då databasen var central i projektet hade en utförligare beskrivning av hur man designar databas i samarbete med användaren välkomnats. Projektgruppen började med att granska det nuvarande systemet och de data som samlades in. Sen gjordes en preliminär bedömning av vilka data som var nödvändiga och hur och om det var realistiskt att samla in dem. Det förra systemet hade en mängd olika data som skulle samlas in men i själva verket blev endast ett fåtal fält ifyllda av inmataren. Projektgruppen normaliserade databasen och gjorde en preliminär ER modell. De skapade tabellerna, utifrån entiteterna, i MS Access. Användaren var med hela tiden och de diskuterade vartenda fält noga så att inget blev placerat i en viss tabell utan föregående diskussion om begreppets egenskaper och kopplingar till andra begrepp. Utvecklarna upplevde att de lärde sig mycket om verksamheten på detta sätt medan användaren tyckte att processen förtydligade hans idéer om vilken information som var viktigast. Projektgruppen upplevde designprocessen som mycket positiv. ”Det var verkligen en bra start på projektet” sa Danisa.

De fick dock vissa problem med verktyget eftersom MS Access inte är ett professionellt utvecklingsverktyg utan mer anpassat för lekmän. Till exempel har Access inte förmåga att låta

många fält tillsammans bilda en unik nyckel, om de inte utgör en primär nyckel. Detta var ett problem när det visade sig nödvändigt att lägga till ID fält i några tabeller som hade stora primärnycklar. Det var dock nödvändigt att behålla alla fält i primärnyckeln unika tillsammans för att säkerställa databasens integritet. De löste problemet genom att behålla primärnycklarna men använde IDfälten för att koppla tabellerna i databasfrågor. Denna lösning gav ökad prestanda och bevarade dataintegriteten.

Användaren ville att man skulle kunna se vem har matat in informationen och när den var inmatad. De utsåg alla tabeller medfälten "timestamp" och "inmatare" för detta ändamål. Det var dock inte möjligt att se om någon hade gått in i databasen efteråt och ändrat uppgifter. De bestämde att vänta med denna funktionalitet och istället koncentrera sig på att göra systemet användbart i verksamheten. Detta för att det skulle ha krävt att man skapade en eller flera ändringstabeller och försåg varje fält i databasen med händelsefunktioner som tog hand om att registrera ändringar och markera dem. Dessutom skulle de behöva skapa ett fönster för att visa ändringar. Utvecklarna förklarade för användaren att denna funktion skulle kunna ta mycket tid och resurser i anspråk och begränsa deras möjligheter att göra andra funktioner färdiga i tid. Användaren tyckte då att det var bättre att vänta med denna funktion.

I tblPatient finns fältet "huvuddiagnos" som är här en beskrivande parameter som hjälper läkaren att kategorisera patienten. I den gamla versionen fanns det också ett fält "andra sjukdomar", där man ibland skrev en eller flera diagnoser. Utvecklarna tyckte att man borde ta bort dessa fält och skapa en speciell tabell för diagnoser. "Vi tyckte att det bröt mot god design att ha två fält med information av samma slag" sa Vin. "Och värden ska vara atomiska, det var de inte i det ena fältet." Användaren förklarade att denna information endast var avsedd för att hjälpa läkaren att kategorisera patienten. Den ska inte användas för att ta fram statistik. "Mycket fullständigare information finns redan i Melior" sa Daniel. "Varför krångla till det med en till tabell när det ändå inte ska användas." "Han har rätt" sa Danisa. "Eftersom det är många till många kopplingar mellan patient och diagnos skulle vi behöva två tabeller som behöver anslutas varje gång man byter patient. Det kommer bara att försämra prestandan ännu mer!". De bestämde därför att behållafälten som de var.



**Figur 4.3: Tabeller och relationer** Primärnycklar är i fet stil. Om det finns flera fält i fet stil betyder det en sammansatt primärnyckel<sup>2</sup>.

Resultatet av denna fas visas i Figur 4.3. Några av entiteterna har så många attribut att det inte är möjligt att visa dem alla i ER modellen. De bestämde sig därför för att använda relationsvyn i MS Access istället för ER modell. Relationsvyn liknar en förenklad UML modell och används för att sätta default relationer mellan tabeller. Det är så enkelt att skapa tabeller och relationer i Access att det går bra att designa databasen genom att börja skapa tabeller direkt. I designprocessen tar man enkelt bort oönskade tabeller och egenskaper.

Efter att databasdesignen var färdig, flyttades alla data från den gamla databasen till den nya. Många data var inkonsekventa och behövde lagas, eller ”tvättas”. Innan flyttningen kunde utföras var det därför ganska mycket arbete med att göra SQL satser för tvättningen. Detta arbete tog ungefär två dagar.

Databasdesignen gick mycket bra. Projektdeltagarna var mycket nöjda med designprocessen och det nära samarbetet med användaren gjorde det möjligt att rätta till många missförstånd i processen. Deltagarna tyckte att genom att designa databasen tillsammans med användaren, fick de en förbättrad förståelse för arbetsprocesserna i verksamheten.

#### 4.7.2 Design av gränssnitt

Projektgruppen använde principer från Användbarhet i praktiken av Ingrid Ottersten & Johan Berndtsson (Ottersten & Berndtsson, 2002) vid utformning av gränssnittet. Gruppen tyckte inte att Gulliksens (2002) riktlinjer var tillräckligt tydliga eller praktiska i detta avseende.

Vid utformning av gränssnittet upplevde utvecklarna att det var en utmaning att dela in sig i olika roller såsom interaktionsdesigner och programmerare. ”Ofta ändrar jag en sak i gränssnittet och

<sup>2</sup> Mellan tabellerna Access, Ingrepp och Accessfunktion är relationen tredubbel, men den borde vara tvådubbel dvs. bara mellan personnummer och accessnummer i alla tabellerna. AccessId skulle bara vara unik och indexerad som nämnt ovan. Den tredubbla kopplingen är ett MS Access knep som utvecklarna använde för att få AccessId att uppdateras automatiskt i gränssnittet.

går sen direkt och programmerar funktionaliteten. Jag kan inte välja om jag är interaktionsdesigner eller programmerare. Jag är både och!” tyckte Vin. Det blev så att dessa roller blev förenade, dock fungerade Danisa mer som interaktionsdesigner medan Vin gjorde större delen av programmeringen.

Två metoder blev provade för gränssnittsutvecklingen: Under de första tre veckorna av designtiden fick användaren själv designa gränssnittet under vägledning av interaktionsdesignern och programmera underliggande funktioner med programmeraren. Under de senaste tre veckorna ändrade de till iterativ utveckling där prototyper designades av systemutvecklarna och testades sedan av användaren. Detta gjorde man för att undersöka om ytterligare användarmedverkan skulle vara nyttig eller till och med skadlig för utvecklingsprocessen och användbarheten.

Båda metoderna hade sina fördelar och nackdelar men den första gav ett betydligt bättre resultat. Först och främst gick designprocessen mycket snabbare och smidigare när användaren designade själv. Användaren var nöjdare och positivare när han själv designade med hjälp av interaktionsdesignern. Han behövde ingen hjälp med designverktygen i MS Access eftersom han redan hade erfarenhet av att designa gränssnitt i MS Access sedan tidigare. Interaktionsdesignern hjälpte honom dock med att organisera gränssnittet enligt riktlinjerna för gränssnittsdesign och med andra designtips. Den iterativa processen med design och användartest gick fort och smidigt. Programmeringsdelen var dock ett problem. Användaren var programmeringskunnig och behövde inte mycket hjälp, men programmeraren tyckte att det var trist och tidskrävande att sitta och hjälpa användaren. Dessutom tog det användaren längre tid att programmera än det hade tagit programmeraren, så att tidsmässigt var det inte lönt för projektet. Användaren var dock nöjd, eftersom på detta sätt programmerade han snabbare och lärde sig snabbare än han skulle ha gjort på egen hand.

I den andra metoden designade utvecklarna prototyper som användaren sedan testade. Designen ändrades efteråt av utvecklarna enligt användarens önskemål. Här gick själva arbetet med designen fortare men användartesten visade sämre resultat än tidigare och dessutom krävdes många fler iterationer. Det uppstod missnöje eftersom användaren ofta var mycket kritisk till designen och utvecklarna tyckte att de hade designat enligt användarens önskemål. Detta bidrog till irritation mellan utvecklarna och användaren som gjorde att utvecklarna började känna att användaren var besvärlig och användaren tappade intresset för systemutvecklingen. Diskussionen som ägde rum i samband med denna analys visade att detta berodde på missförstånd mellan användaren och utvecklarna. Eftersom inga riktiga processanalyser hade utförts hade utvecklarna inte tillräcklig grund att bygga på för att designa utan användaren. Denna metod upplevdes som besvärlig av alla deltagarna. ”Det var ohållbart att fortsätta missförstå varandra” menade Vin. ”Antingen borde vi ha designat med användaren hela tiden eller gjort utförliga processanalyser, designat gränssnittet på papper, gjort en kravspecifikation för det och fått honom att skriva på att så här ska det vara.”

Daniel menade att den kritiska faktorn var hans tidsbrist, han hann inte sätta in utvecklarna i arbetsprocesserna tillräckligt bra, därför blev det så mycket missförstånd när de fortsatte designen utan honom. Danisa var inte enig: ”Det hade inte varit lönt att sätta in oss i arbetsprocesserna bättre, du borde ha varit med och designat hela tiden. Om du hade varit med i november så hade systemet varit färdigt nu”. ”Det är sant” sa Daniel, ”men det var inte möjligt, tyvärr”. ”Ja, ” sa Vin, ”vi skulle ändå ha missförstått en massa. Man tror att man förstår, men så har man missat någonting och man upptäcker det inte förrän man får prototypen i huvudet.”



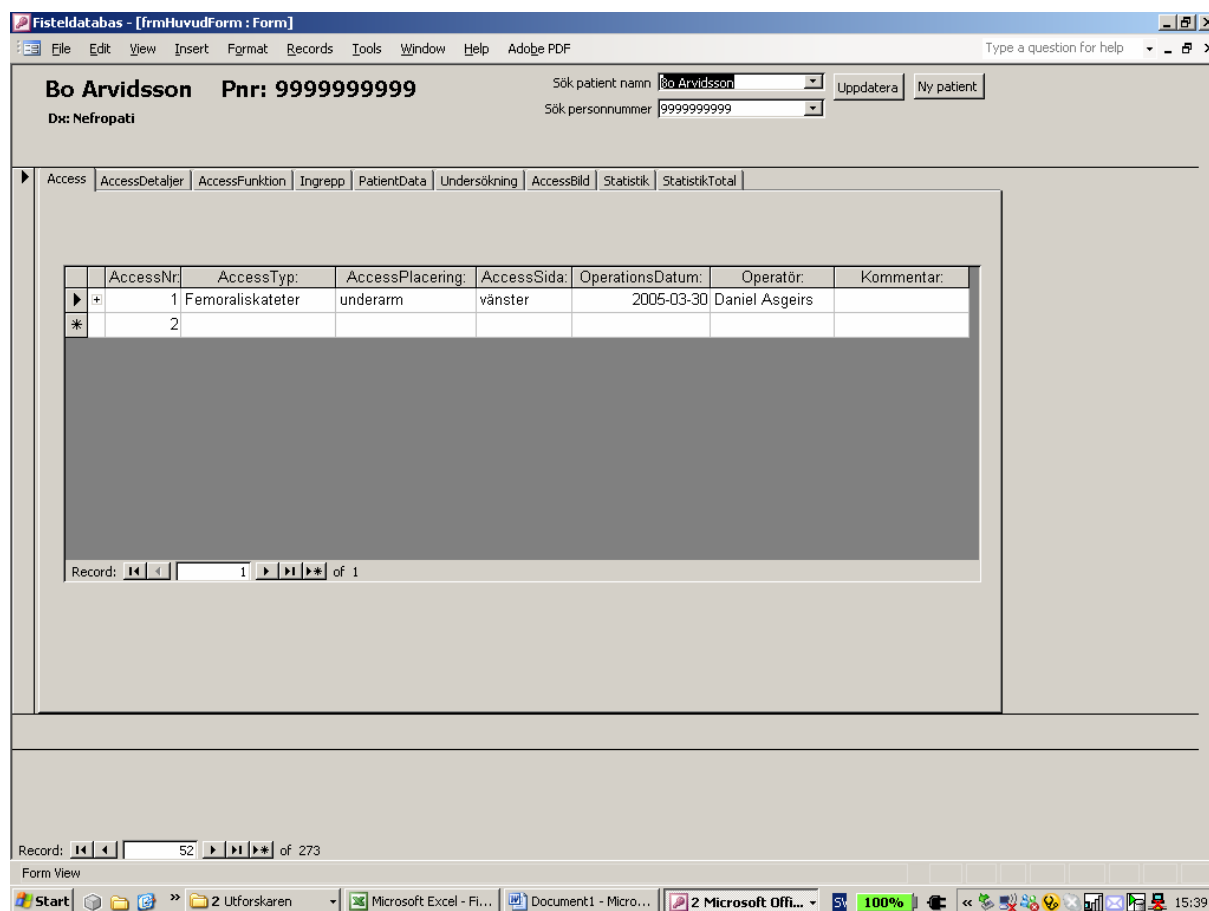
Designprocessen med utökad användarmedverkan ägde rum i början av gränssnittsdesigndelen av praktiska skäl, men också för att ett bra resultat inte skulle kunna bortförklaras av att projektgruppen hade fått mer erfarenhet och var effektivare eller bättre av det skäl enbart.

Under första delen av gränssnittsdesignen var det användaren som satt vid tangentbordet och använde de visuella verktyg för gränssnittsdesign som följer med MS Access för att designa gränssnittet. Interaktionsdesignern satt bredvid och hjälpte till med att organisera gränssnittet enligt riktlinjerna.

Programmeringen i projektet utfördes i viss mån parallellt med att prototyperna genomgick cykler av utvärdering och utveckling. Detta innebar att saker som redan programmerats på ett sätt ibland var tvungna att ändras. Många gånger kändes det som att arbete blev utfört i onödan och en viss motvilja mot att ändra i det som redan var gjort uppstod.

Projektgruppen använde principer från Användbarhet i praktiken av Ingrid Ottersten & Johan Berndtsson (Ottersten & Berndtsson, 2002) vid utformning av gränssnittet, som skapades i MS Access. De tyckte dock att de var ganska självklara och inte till stor hjälp. De bestämde sig för att använda flikar, de tyckte alla att detta var ett naturligt sätt att organisera informationen.

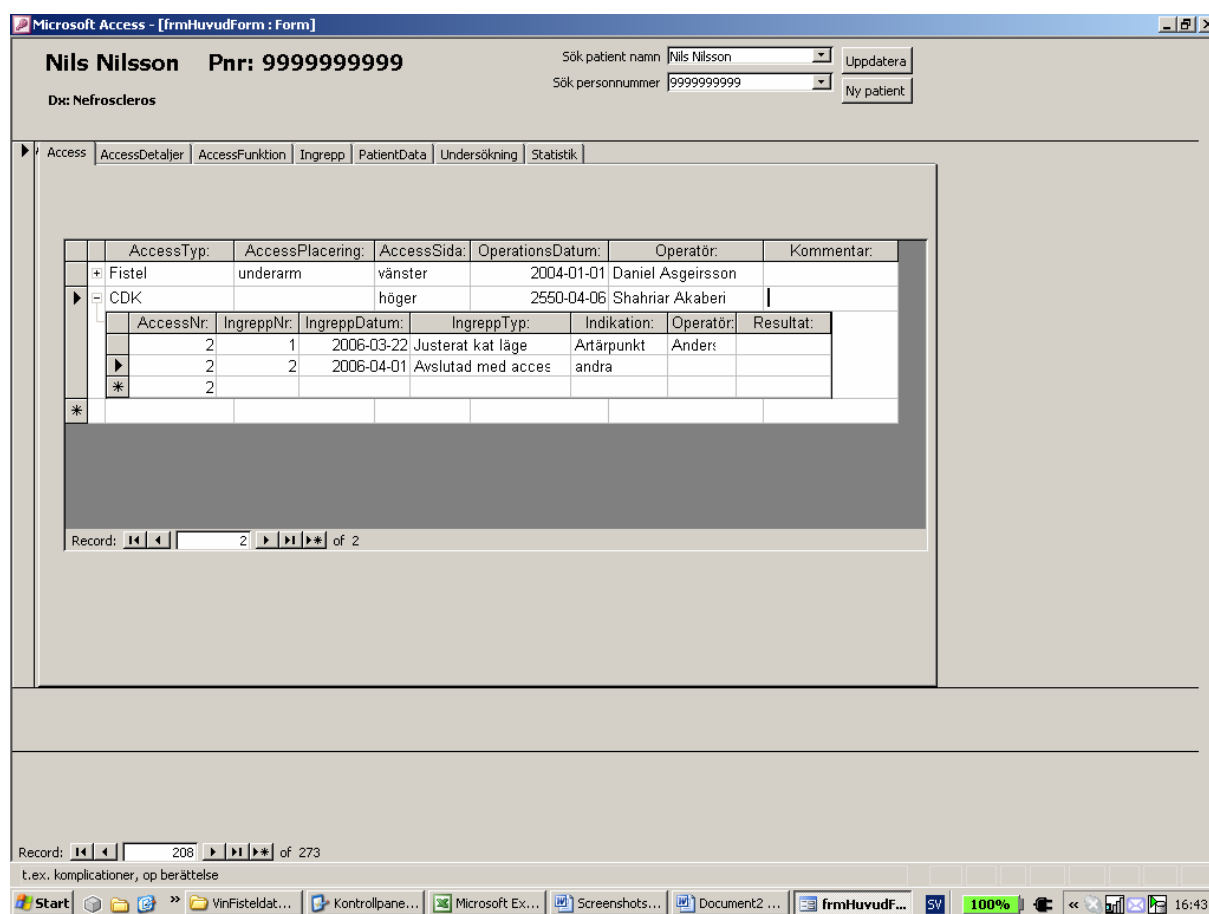
När applikationen öppnas kommer upp ett fönster. Till höger är det två fält där man kan skriva in patientens namn eller personnummer (se Figur 4.4).



Figur 4.4: Access fönstret Detta fönster öppnas när man öppnar applikationen.

Man kan då söka efter en patient efter personnummer eller namn. Det var en debatt om man skulle kunna söka efter namn, eftersom detta ökar chansen att man matar in data hos fel patient. Användaren tyckte att det var smidigt, men de bestämde sig för att ta bort denna möjlighet innan någon annan började använda systemet.

När man har valt en patient kommer hans namn och personnummer upp i stora bokstäver, uppe i fönstrets vänstra hörna. De tyckte att det var viktigt att man tydligt såg vilken patient man jobbade med för att minska risken för att mata in data hos fel patient. Patientens blodaccesser visas i en tabell. De kan vara många och det är viktigt att få en överblick över dem, tyckte deltagarna. Man kan mata in en ny access direkt i tabellen och om man trycker på plustecknet vid en access kan man se de ingrepp som har gjorts på accessen och även mata in ett ingrepp på den (se Figur 4.5). Detta tyckte användaren att var smidigt eftersom han inte behövde hålla på att byta flikar, utan kunde göra allt på samma ställe. ”Det är jättebra att kunna göra allt i samma flik, men ändå känns det inte stökigt!” sa Daniel.



**Figur 4.5 Accessflik med underdatablad öppet** Här kan man klicka på en access för att lägga till ett ingrepp. Då öppnas ett underdatablad som är kopplad till den aktuella accessen. Detta ger fokus på den valda accessen samt bättre överblick.

När man skriver in en patients personnummer eller namn, kommer det upp en lista under inmatningsrutan där man kan välja en av redan inmatade patienter. Denna lista blir kortare eftersom man skriver in fler siffror/bokstäver eftersom det finns allt färre patienter som har denna kombination i början på sitt personnummer/namn. Ibland behöver man mata in en ny patient. Det händer ofta att användaren börjar skriva in patientens personnummer eller namn och får upp en lista som sedan försvinner eftersom ingen patient i databasen har denna kombination. Då klickar man med musen till höger om inmatningsrutan på knappen ”Ny patient”. Då öppnas ett formulär där man kan mata in en ny patient, se Figur 4.6.

The screenshot shows a Microsoft Access form titled "tblPatient" for a patient named Nils Nilsson (Pnr: 9999999). The form is divided into several sections:

- Header:** Patient name "Nils Nilsson" and ID "Pnr: 9999999".
- Diagnosis:** "Dx: Nefroscleros".
- Access Table:** A table with columns "AccessTyp:" and "AccessPlacering:". It contains two rows: "Fistel" (underarm) and "CDK".
- Form Fields:** Input boxes for "Århundrade:" (value: 19), "Personnummer", "Förnamn", "Efternamn", "Huvuddiagnos", "Behandlingstyp", "DatumDialysStart", "Komment", "AndraSjukdomar", and "Inmatare".
- Navigation:** Record navigation controls at the bottom of the form, showing "Record: 1 of 2".

The Windows taskbar at the bottom shows the Start button, several open applications (Uppsatsen, Microsoft Access, Screenshots.doc, anvandningsanalys\_P...), system tray icons (69% battery, network, volume), and the time 15:59.

Figur 4.6 Fönstret "ny patient"

Projektgruppen diskuterade om det skulle vara pop-up fönster eller en till flik. ”Jag tyckte först att pop-up fönstret störde helhetsupplevelsen, och bröt stilen med flikar, men nu tror jag att det var den bästa lösningen” sa Danisa. ”Vi kan inte ha hur många flikar som helst.”

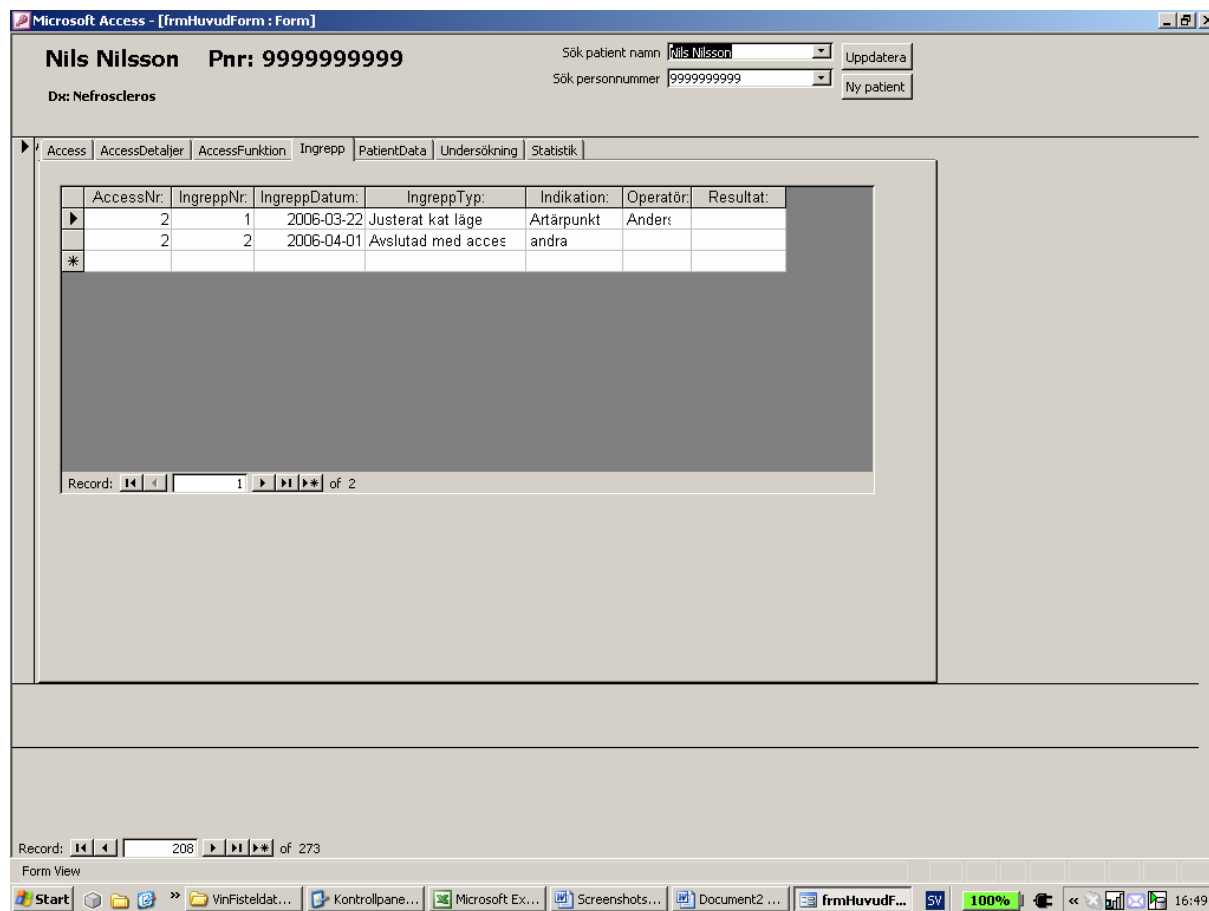
Här matar man in information om en ny patient. Här finns ett fält som visar det århundrade patienten är född på. Detta var nödvändigt för att kunna räkna ut personens ålder utifrån personnumret. Man skulle också kunna använda ett långt personnummer som börjar med 19 eller 20. De diskuterade detta och kom fram till att eftersom det var så pass få patienter födda på 2000 talet, skulle man spara tid på att ha 19 som default värde, men i de fall en patient var född på tvåtusentalet, kunde man ändra 19 till 20.

Fältet ”Andra sjukdomar” var en legacy från den gamla databasen. De bestämde att inte ta bort den för att ibland räcker det inte att skriva huvuddiagnos, det finns andra sjukdomar som är viktigt att kommer fram. Utvecklarna tyckte att mest korrekta vore att ha en tabell för sjukdomar, men eftersom det redan fanns ett journalsystem kändes det onödigt. ”Vi ville inte göra en tabell för sjukdomar, den finns ju i Melior” sa Vin. ”När systemet kopplas till Melior kan man använda den för att hitta intressanta samband.” Därför blev det så att om det behövs kommer andra sjukdomar att matas in i detta fält, men det kommer inte att användas vid dataanalys.

The screenshot shows a web-based application window titled "Fisteldatabas - [frmHuvudForm : Form]". The patient information at the top is "Bo Arvidsson Pnr: 9999999999" with a diagnosis of "Nefropati". The "Access" tab is selected, showing details for "AccessTyp Femoraliskateter" and "AccessNr 1". The form includes fields for "OperationsDatum" (2007-03-31), "VårdTyp", "Operatör" (Daniel Asgerisson), "AccessPlacering" (underarm), "AccessSida" (vänster), "AccessTypII", "Kommentar", "Inmatare", and "Timestamp" (2007-02-13 15:26:34). The bottom of the window shows a record navigation bar and a Windows taskbar.

**Figur 4.7: Detaljfönstret** Under fliken AccessDetaljer kan man mata in ytterligare information om den valda accessen om nödvändigt. Denna information behövs bara i undantagsfall och är därför samlad i en separat flik. Det är tanken att man går genom flikarna från vänster till höger, men man kan hoppa över vissa flikar om det inte finns någon extra information som man behöver mata in under fliken. Här visas information som redan matats in under accessfliken för att man ska slippa bläddra fram och tillbaka om man glömmer.

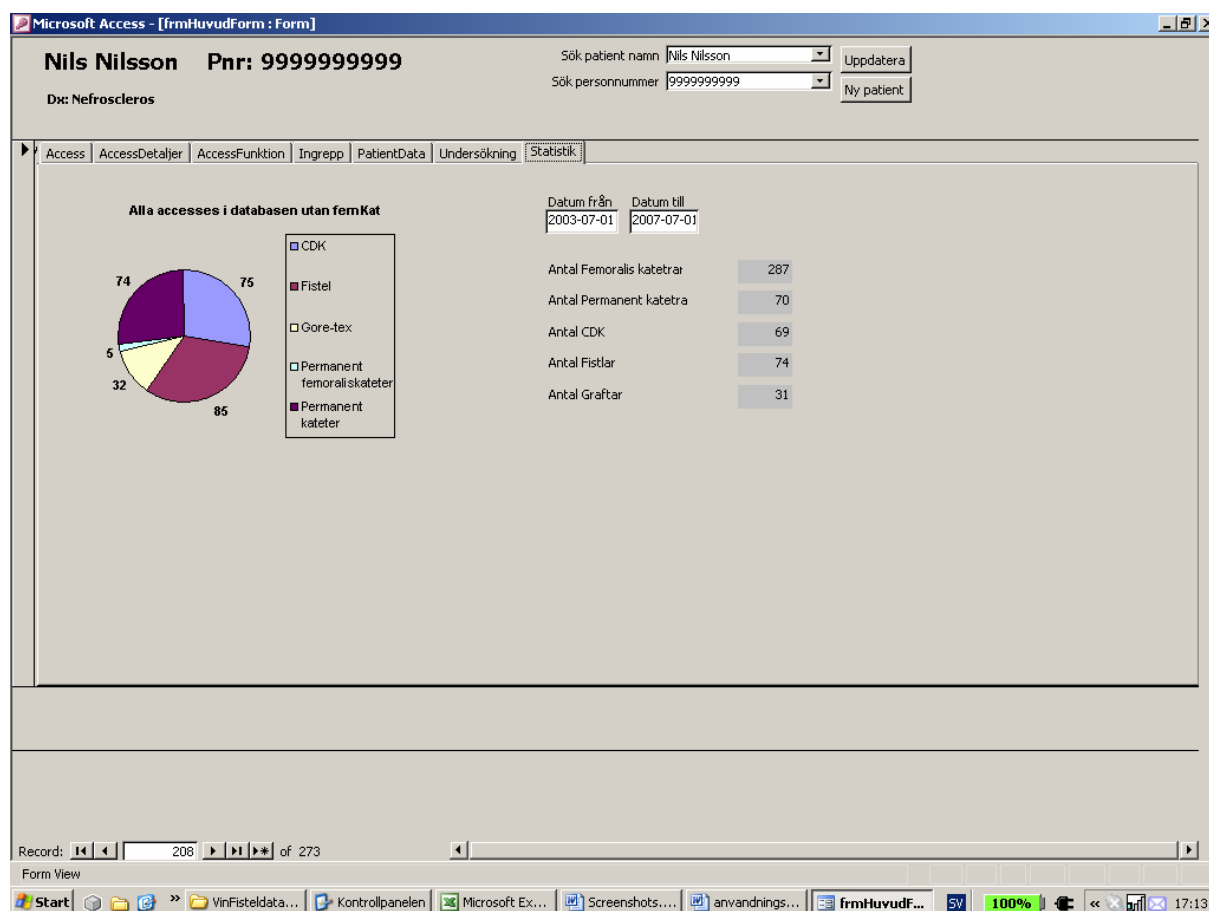
Det var några fält i tblAccess som nästan aldrig fylldes i. Några av dem var viktiga för den statistiska analysen men hade inte fyllts i eftersom det var krångligt att hitta informationen. I dialysavdelningen hade man tänkt anställa senare en person för att hitta dessa data och fylla i fälten. Man behövde därför ett inmatningsformulär för detta ändamål i gränssnittet men som inte fick störa inmatningen av ordinarie data. De löste detta problem genom att göra en speciell flik, Accessdetaljer. Under en annan flik "Accessfunktion" matas det in detaljer som har att göra med accessens funktion. Användartester visade att dessa data matas oftast in separat från de andra. Detta är också för att göra informationen mer överskådlig. Överst på fliken står vilken access man arbetar med för att undvika att man matar in under fel access.



**Figur 4.8 Ingreppsfliken** Här kan man se en lista över alla ingrepp på alla accesser. Detta ger överblick över patientens ingreppshistoria och ger underlag för beslut om vilket slags ingrepps ska göras.

Under fliken Ingrepp finns en överblickstabell som visar alla ingrepp som gjorts på alla accesser. Denna flik gjordes efter att användartest visade att användaren behövde kunna få överblick över alla ingrepp utan att behöva klicka på + tecknet framför alla accesser i accessfönstret.

Fliken Patientdata blev inte färdigställd. Tanken var att detta fönster skulle ge en överblick över patienten. I framtiden när systemet används som ett beslutstödsverktyg i verksamheten behöver läkaren ett överblicksfönster där viktig och relevant information om patienten är samlad på ett ställe och presenterad på ett sätt som läkaren upplever som bekväm. Fliken undersökning blev inte heller färdigställd. Slutligen hann projektteamet bara börja på fliken statistik men inte avsluta innan projekttiden var slut, se Figur 4.9. Där ska man kunna se beskrivande statistik men även ställa egna frågor till databasen med SQL.



**Figur 4.9 Statistik** Här kan man välja att se statistik över en viss period. Denna flik är inte färdig. Det ska vara möjligt att specificera olika avgränsningar (så **Figure 4-1** som kön och ålder) och visningsmetoder (så som scatterplot och histogram).

Först använde de två filer, en för data och en för gränssnittet. Detta för att förenkla utvecklingen av gränssnittet, skilt från databasen. Med denna metod var det möjligt att enkelt prova olika prototyper av gränssnittet utan att påverka databasen. Denna metod var rekommenderad av Balter (2002). De använde länkade tabeller för att koppla de två filerna. Men det visade sig att prestanda försämrades avsevärt av detta, dessutom var det krångligt att behöva ändra sökväg till filerna varje gång man flyttade databasen och gränssnittet mellan arbetsplatser. Utvecklarna tvingades ändra tillbaka så att gränssnitt och databas var i samma fil. Denna motgång, som orsakades helt av tekniska svårigheter och i grund och botten felaktig information i Balter (2002) var svår för samarbetet. Det hade varit heta diskussioner i projektgruppen om denna metod skulle användas. Utvecklarna hade rekommenderat denna metod men användaren tyckte att det var enklare och säkrare att ha allt i samma fil. Gruppen bestämde till slut att använda delade filer. När det visade sig att detta var olämpligt, skadades användarens förtroende för utvecklarna. ”De använde oerhört mycket tid för att länka allting på rätt sätt, det var jättekrångligt att jobba med denna metod. Sedan blev databasen så seg att jag fick vänta 20 sekunder bara för att växla mellan flikarna. Då sa jag stop, då räckte det!” sa Daniel. ”Det blev jättejobbigt, projektet var noggrant planerat, men vi hade inte tid för sånt krångel. Vi är jättearga på Balter!” sa Vin.

Trots motgångarna tyckte hela gruppen att samarbetet gick bra och var produktivt. Vid halva projekttiden ändrade utvecklarna designmetoden från att låta användaren själv designa till en iterativ metod där utvecklarna designade prototyper som användaren sedan testade på arbetsplatsen. Utvecklarna var entusiastiska i början. ”Det blir skönt att få frihet att designa själv”

tyckte Vin. Arbetet gick snabbare och de hade programmerat in bildhantering innan nästa användartest. Men användaren var inte särskilt imponerad. ”Det viktigaste är inte att kunna visa bilderna, jag behöver först kunna lagra bilderna på ett enkelt sätt. Vi måste bestämma det först.” ”Vi började med bildhanteringen för att jag tyckte att det var ett av det svåraste” sa Vin. ”Enligt min erfarenhet i programmering ska man alltid börja med det svåraste för att kunna bli färdig i tid. Jag förstår inte varför han är så negativ. Vad spelar det för roll om vi börjar med bildvisningen? Vi tänkte göra det ändå.” Daniel sa: ”Jag vet vad som är viktigast, det är mycket bättre om ni talar med mig innan ni bestämmer att använda en massa tid på att göra något onödigt”. Dessa problem fortsatte och det verkade som kommunikationen hade brutits ner, de var inte längre inställda på samma frekvens, tyckte Danisa.

Ett annat problem var att användaren ville vara insatt i hela koden. ”Jag måste förstå vad som händer i programmet, annars kan jag inte rätta till fel. Jag kan inte alltid ringa Vin och Danisa när jag får problem” sa Daniel. Användaren ville ha en full förståelse för hur koden fungerade men det gick inte att sätta in honom i koden. Dels för att den blev alltmer komplicerad i och med att de började utveckla bildhanteringen, dels för att utvecklarna hade svårt att förklara koden för användaren. Utvecklarna upplevde att detta berodde på deras begränsade erfarenhet som programmerare. Detta gjorde att arbetet gick mycket trögare nu och var därför inte lika roligt. Användaren blev irriterad: ”De kan inte förklara någonting. De vet inte varför de gör saker. De borde ha haft det klart för sig innan vi började”. Vin sa: ”Det verkar som användaren förväntat sig att vi skulle vara lika säkra i vår roll som utvecklare som han var i sin roll som läkare. Men det var vi inte alls.”

Resultatet blev att användaren tappade intresset för designprocessen, användartesten blev svårare att utföra och han gav mindre feedback till utvecklarna. De fortsatte dock tills projekttiden var slut. Gränssnittet blev inte färdigt men kunde dock (med varsamhet) användas för inmatning. De hade inte hunnit färdigställa felhantering och valideringsfunktioner, så de ansåg inte att gränssnittet kunde användas av någon annan än den initiala användaren.

#### 4.8 Evaluering av användbarhet

Enligt definitionen av användbarhet (avsnitt 3.2.1) är användbarhetens tre viktigaste beståndsdelar ändamålsenlighet, effektivitet och tillfredsställelse.

För att mäta ändamålsenlighet använde de den grad, till vilken systemet uppfyller kravspecifikationen enligt Sommerville's definition av kvalitet (1989, i Hägerfors, 1995). Enligt det hårda systemtänkandet räcker det att systemet uppfyller kravspecifikationen för att det anses ha tillräcklig kvalitet (Hägerfors, 1995; s.52) så att vår definition av ändamålsenlighet är synonym med kvalitet från det perspektivet.

För att bedöma ändamålsenligheten har utvecklarna alltså jämfört systemet med de behov som de utgick från i början att systemet skulle uppfylla dvs.

1. insamling av data om blodaccesser för statistisk analys
2. förbättra kvalitet på patientbehandling och uppföljning
3. ge underlag för förbättringar i behandlingsstrategier

Det första syftet har endast uppfyllts till hälften. Systemet används redan för att samla in all data som specificerades i kravanalysen. De har inte hunnit göra några statistiska funktioner, så att den

statistik man behöver är fortfarande gjord i Excel eller andra program (som Matlab och SPSS). Andra och tredje syftet har inte uppfyllts alls. (se bilaga 9.2.1)

Den andra komponenten av användbarhet är effektivitet. De mätte hur länge inmatningen tog i det nya systemet och jämförde med det gamla systemet. I det gamla systemet tog inmatningen längre tid än i det nya systemet:

Uppgift	Gamla systemet	Nya systemet
Att mata in en ny patient	12 sek	13 sek
Att lägga till en blodaccess	35 sek	24 sek
Att lägga till ett ingrepp	36 sek	18 sek

Medeltidsförbättringen var närmare 40 % när man viktar talen med respektive frekvensandel (10 % för nya patienter, 30 % inlägg av blodaccesser och 60 % för ingrepp).

Effektiviteten tycktes vara tillfredsställande enligt användaren. Dock tyckte han att det var tidskrävande att växla mellan flikarna hela tiden. Förutom direkt förbättring i tidsåtgång har det förenklade gränssnittet lett till att användaren känner att det tar mindre resurser och är mindre stressigt att mata in, som i sin tur leder till att han har mer energi och är effektivare i sina andra sysslor (se bilaga 9.1.1).

Användartesten visade att användaren var tillfreds med vissa aspekter av systemet. Han var nöjd med systemets databas och att systemet har underlättat processen att mata in data. Det är en klar förbättring från det tidigare systemet. Han tycker dock att systemet är långt från att uppfylla andra krav. Han är inte nöjd med att bara han kan mata in data eftersom systemet är så ostabilt. Han är inte heller nöjd med att varken statistiken eller bildhanteringen blev färdigställda.

Enligt deltagarnas uppskattning, gick designarbetet ungefär 30 % snabbare när utvecklarna designade, medan själva projektet framskred ungefär 90 % snabbare när användaren designade med hjälp av interaktionsdesignern. Utvärderingarna påvisade konsekvent brister i användbarhet och missnöje hos användaren. Detta försenade projektet, förutom att skapa missnöje hos deltagarna och minska deras motivation.

#### 4.9 Projektets slut

Projektet blev inte fullbordat. Det avslutades sommaren 2005 eftersom tidsgränserna var överstigna och det fanns inte mer tid eller resurser för projektet. Då var själva databasen färdig. Denna användes redan i verksamheten och var till belåtenhet. En prototyp av gränssnittet som tillhandahöll de mest grundläggande funktionerna var färdigt, men eftersom det saknades felhantering och säkerhet var det inte möjligt att ta projektet till nästa nivå; det kunde inte användas av andra än denna enda användare som deltog i utvecklingen och då bara med stor varsamhet. Kravspecifikationen uppskattades ha uppfyllts till ca 25 %.

När projektgruppen diskuterade projektet efteråt var de inte säkra om det hade lyckats eller misslyckats. ”Vi har tydligen misslyckats i planeringen, det tog mycket längre tid än vi hade planerat” sa Vin. De tyckte att den användarcentrerade metoden hade fungerat väl, men att det senare projektskedet, där användaren enbart testade prototyperna men inte var med i designmomenten, gick mycket sämre än det förra. ”I det första skedet var vi produktiva” sa Vin. ”I det senare gick vi i cirklar.” ”Det gick mycket bättre när jag var med och designade systemet, det blev lite jobbigt att behöva kritisera efteråt när det inte blev riktigt som jag hade tänkt. Ni



hade lagt mycket jobb på det och så...” sa Daniel. ”Ja, det var så jobbigt att hela tiden få kritik men nästan aldrig beröm”, tyckte Vin. ”Det är synd att vi inte kunde hantera problemen bättre, då hade vi hunnit längre” sa Daniel. ”Men vi har i alla fall lagt grund till ett bra system” sa Danisa.

## 5 Diskussion

Det visade sig att användarmedverkan gick mycket bra och var avgörande för design av databasen. Vid designen av databasen var samarbetet optimalt. Deltagarna var produktiva, utvecklarna lärde sig mycket om användarens minivärld och hade kul samtidigt var användaren nöjd med den hjälp och han fick av utvecklarna i designarbetet. Där rådde ett ömsesidigt lärande mellan användaren och utvecklarna. Användaren var mycket nöjd med den resulterande databasen och de ansåg alla att den var bra utformad från databasteoretisk synpunkt och användningen av prototypen visade att även praktiskt var den väl genomtänkt, eftersom det uppkom mycket få problem med inmatning av data, även om alternativen bestod i de flesta fall av ett bundet val och få möjligheter till fritext. Detta för att värna om dataintegritet och undvika behov av framtida datatvättning (vilket hade redan inneburit mycket arbete i början av projektet). Det nära och informella samarbetet med expertanvändaren, där han deltog i designen av databasen tillsammans med utvecklarna, gav ett mycket bra utfall vilket stödjer att denna metod är lämplig för design av databaser.

I detta avseende kan också utvecklingsverktyget Microsoft Access ha hjälpt. Även om få systemutvecklare anser det vara ett seriöst utvecklingsverktyg (personlig kommunikation med lärare på institutionen), har Access dock den fördel att databasdesignen går snabbt och enkelt för sig vilket är en viktig egenskap hos ett databasverktyg (se avsnitt 3.4.2).

Även under första delen av gränssnittsdesignen upplevde utvecklarna användarens medverkan som enbart positiv och kände att den var av stor betydelse för designarbetet. Däremot tyckte de inte samma sak om den senare delen. Samarbetet fungerade inte bra och utvecklarna tyckte inte att användarmedverkan gav något positivt utbyte. Samtidigt upplevde de att användaren tappade intresset för gränssnittsdesignen.

Denna förändring kan bero på graden av hur mycket och i vilken form användaren engagerade sig i de olika aktiviteterna. Under den första delen av arbetet med gränssnittsdesignen var användaren aktiv i designarbetet, men i den senare delen var han passiv. Istället för att vara med och designa fick han testa och tycka till om de olika lösningarna utvecklarna designade.

Under denna del av designarbetet rådde det mycket klagande av samtliga projektdeltagare, klagomålen var över missförstånd mellan utvecklare och användare. Detta påverkade det fortsatta samarbetet i utvecklingsprocessen på ett negativt sätt. Dels bidrog detta till att arbetet fortskred långsamt och att utvecklarna kände sig kränkta av användarens ständiga kritik. Under denna period upplevde båda parter, användaren och utvecklare, att det fördes en otydlig och osamstämmig kommunikation dem emellan.

Vår analys resulterar i att problemen i det senare skedet bestod av 1) ihållande missförstånd mellan användare och utvecklare, 2) samarbetsproblem och 3) motivationsbrist.

### 5.1 Missförstånd

Missförståndet tycks ha orsakats av en brist i kommunikationen som uppstod när inte användaren var där för kontinuerligt samarbete. Detta kan bero på att gruppen aldrig gjorde en grundlig systemtäckande processanalys/uppgiftsanalys utan den växte fram iterativt så som användbarhetsdesignmetoden föreskriver. De hade vant sig vid att analysen var informell och gjorde inte en separat fas för denna. Denna gjordes istället löpande under själva designarbetet genom att införa en stunds analys för att ställa frågor till användaren, efter varje arbetspass av

designarbete. Effektivt utgjorde prototypen den egentliga kravspecifikationen. Detta gick bra i det första designskedet men i det senare tycks det ha blivit för lite analys och den kommer dessutom för sent, dvs under användbarhetstesten, efter att designarbetet har ägt rum. Analysen ska enligt Gulliksen vara en separat fas och man ska i början bestämma hur användaren kommer att medverka i den, så som i alla andra faser. Kravanalysen är grunden som systemdesign bygger på (Gulliksen, 2002), och brister den, lider systemets användbarhet.

När kommunikationsförloppet analyseras enligt Gallivan och Keils (2003) modell (se avsnitt 3.4.4), är det tydligt att användaren blev medveten om de meddelanden han ville förmedla (steg 1 i deras modell) under användbarhetstesten. Men i andra sammanhang blev användaren ofta inte medveten om nödvändiga meddelanden (steg 1) eftersom underlag saknades när han inte deltog i designprocessen. Under användbarhetstesten kunde användaren tycka till om prototyperna men hans attityd var negativare än i det första designskedet eftersom han upplevde att utvecklarna inte hade tolkat rätt hans tidigare kommentarer i utvärderingen av designlösningen. Utvecklarna å sin sida tyckte att användaren var otydlig i sin kommunikation. Det handlade alltså om möjlig otydlighet i skickade meddelanden (brist i steg 2) och feltolkning å utvecklarnas sida dvs. brist i steg 3. Utvecklarna prioriterade och agerade på de meddelanden de fick från användaren (steg 4) men användaren tyckte dock att de i vissa fall prioriterade fel, till exempel när de började utveckla bildhanteringen innan det fanns ett system för att lagra bilderna, se ovan. Utvecklarna hade dock anledning att börja med det som var svårast och skulle antagligen vara mest tidskrävande. Det är därför svårt att säga att steg 4 har brutit men det är klart att användaren inte var nöjd med några av utvecklarnas prioriteringar.

Detta missnöje kan möjligtvis spåras till användarens roll som beställare. Som beställare har han mycket större makt än en vanlig användare och han tyckte att han skulle vara med och bestämma hur man planerade utvecklingen. I kraft av sin roll som beställare verkar användaren ha haft mer makt över utvecklingsprocessen än systemutvecklarna när han deltog i hela utvecklingen. Som skildrat i avsnitt 3.4.3 ökar en jämn maktfördelning probabiliteten för konflikter, medan en hierarkisk maktfördelning inte gör det. Det är möjligt att användaren kände att han mist något av sin makt över designprocessen när han inte längre deltog i den. Samtidigt kände utvecklarna att deras makt över designen ökade. Detta gjorde maktfördelningen jämnare och ledde samtidigt till konflikter i enlighet med Gallivan och Keils teori (se avsnitt 3.4.3).

## 5.2 Samarbetsproblem

Projektdeltagarna tyckte att samarbetsproblemen i det senare skedet av gränssnittsdesignen orsakades av missförstånd relaterat till design av systemet och olika uppfattningar om vilka delar som skulle prioriteras. Episoden där utvecklarna gör den bedömning att bildhanteringen behöver utvecklas först, men användaren vill först bestämma hur och var bilderna ska lagras, visar att användaren tyckte att han skulle ta del av att bestämma projektplaneringen på detaljnivå. Utvecklarna samtycker inte, de anser att användaren överskrider på deras område.

Det tyder möjligtvis på att bytet av designmetod inte har varit tillräckligt förberett. De har inte diskuterat hur bytet påverkar deras roller och ansvarsområden. En av utvecklarna säger att hon ser fram emot att ha frihet att designa själv. Denna attityd visar att hon antar att användarens roll i designen, samt hans makt över den, kommer att minska. I den förra projektfasen hade användaren större makt än utvecklarna, både på grund av att det var han som satt vid tangentbordet och designade med systemutvecklarnas hjälp och på grund av sin roll som beställare. Användaren var nöjd, utvecklarna var inte lika nöjda, men arbetet gick smidigt och utan konflikter. När de ändrar metodik i den senare projektfasen får utvecklarna större makt över

designen och direkt uppstår det konflikter. Enligt Gallivan och Keil ska ökad användarmakt leda till konflikter (se avsnitt 3.4.3). I detta fall var det ökad utvecklaremaktsom ledde till konflikt men detta fall är ovanligt eftersom här är användaren också beställare och hade från början mer makt än utvecklarna. Han blev därför självvald projektledare. Därför var maktfördelningen hierarkisk med användaren på toppen i projektets första fas. När andra fasen inleds verkar utvecklarna ha den uppfattning att användaren kommer att släppa taget något av designprocessen och endast agera som användare vid användartesten. Detta verkar ha varit önsketänkande och inte ett gemensamt beslut. Under användbarhetstesten börjar konflikterna, om prioritering, insättning av användaren i koden och missförstånd rörande kravspecifikationen.

### 5.3 Motivationsbrist

I den senare fasen i projektet tappade användaren så småningom intresset för systemdesignen. Enligt Gulliksen och kollegor är en av tre viktigaste framgångsfaktorerna att användare direkt ser hur deras förslag påverkar designen (se avsnitt 3.4.5). När användaren inte var närvarande under designfasen blev detta förskjutet i tiden till nästa användbarhetstest. Användaren tyckte också att utvecklarna missförstod hans förslag/krav och att hans ämnade konstruktiva kritik av detta inte ledde till någon förändring. Användarens kritik mottogs inte på ett konstruktivt sätt av utvecklarna. Detta bidrog möjligen till att användaren tappade intresset. Utvecklarna tyckte också att det var jobbigt att fortsätta på grund av användarens kritik och motivationsbrist.

Utvecklarna själva gav uttryck för motivationsbrist under denna fas. De fortsatte att arbeta med gränssnittet men de ändrade ytterst lite, trots användarens invändningar. Efter den initiala spurten i gränssnittsutvecklingen och användarens kritik av dess resultat, händer det inte mycket. Utvecklarna verkar ha givit upp på designarbetet efter det första användbarhetstestet i den senare projektfasen. Detta kan vara till följd av utvecklarnas relativt ringa erfarenhet av systemutveckling. Studier visar att oerfarna designers tar liten hänsyn till användarens kritik och modifierar produkten minimalt (Sugar, 2001). De analyserar inte heller användarens kritik på djupet, utan tar den bokstavligen och applicerar s.k. plåster (till exempel skriver ett hjälpavsnitt om den problematiska biten) eller helt enkelt tar bort den problematiska funktionen. Här spelar det möjligen in att utvecklarna inte får ersättning för sina insatser. Eftersom de inte hade en ekonomisk motivation att fortsätta och var dessutom upprörda efter användarens kritik, var det relativt enkelt för dem att så småningom minska och till slut upphöra med designarbetet.

### 5.4 Nyttan av användarmedverkan

Som sagt ovan, uppskattades nyttan av användarmedverkan i denna studie genom att analysera resultaten från de integrerade användbarhetstesterna och genom projektdeltagarnas subjektiva uppskattningar.

Deltagarna upplevde att användarmedverkan var av stor vikt i alla delar av projektet, störst var nyttan enligt deltagarna i analysfasen och design av databasen. Det visade sig att nyttan påverkades negativt av att nästan utesluta användaren ur designfasen (förutom utvärdering av prototyper).

Användbarhetstesten visade på mycket större användbarhet av de prototyper som designades innan användaren slutade vara med att designa, men det är svårt att tolka denna skillnad, eftersom användaren då själv designade prototyperna – med hjälp av utvecklarna. Det är möjligt att han därför varit mindre kritisk mot designen, än om någon annan hade designat den. Detta

resultat tyder dock på att uteslutandet av användare från en del av designfasen kan reducera nyttan av användarmedverkan.

Nyttan av användarmedverkan påverkas av användarnas motivation (se avsnitt 3.4.5) som i sin tur påverkas bland annat av hur snabbt och tydligt användare ser sina förslag förverkligas i designen. I det senare skedet av projektet förlorade användaren motivationen, enligt honom själv, som följd av att han inte upplevde att hans förslag påverkade designen. Hans attityd till systemet förändrades och han blev mycket kritisk och tappade till slut intresset för systemutvecklingen. Det har konstaterats ovan att orsaken till detta sannolikt är brister i kommunikationen mellan användare och utvecklare. I detta sammanhang är det intressant att i utveckling av EIS (Executive Information Systems) har man funnit att de två största framgångsfaktorerna för utvecklingsprojekt (som inte är relaterade till projektstyrning eller expertis i systemutveckling) är utvecklingsteamets kommunikationsförmåga och användarnas attityd till systemet, där den senare faktorn påverkas till stor grad av den förra (Srivihok, 1999). Detta verkar också vara fallet med användarcentrerad utveckling.

I vårt fall har kommunikationsfärdigheter inte uppskattats, men utvecklarnas oförmåga att hantera användarens kritik på ett konstruktivt sätt i situationen med nedsatt användarmedverkan, tyder på en brist i kommunikationsfärdigheter. Det finns möjligen också en brist på användarens sida, men samtidigt är det klart att systemutvecklare behöver utveckla färdigheter för att kunna arbeta med användare, oavsett användarnas kommunikationsförmåga. Vår studie tyder på att projektdeltagarnas kommunikationsfärdigheter påverkar starkt nyttan av användarmedverkan och att dessa blir viktigare när användaren inte kontinuerligt är med i designen.

## 5.5 Lärdomar

Denna fallstudie är inte representativ för användarcentrerade projekt. Här var beställaren och användaren en och samma person, det fanns bara en användare medan användarcentrerad metod rekommenderar minst 3-5 användare (se avsnitt 3.3). Systemutvecklarna var oerfarna och ingen av dem fick ekonomisk kompensation för sina insatser, vilket, enligt Gulliksen, kan påverka motivationen. Man kan därför anta att de fick större problem än många andra systemutvecklingsprojekt. Många av de problem som drabbade projektet har dock hänt i andra användarcentrerade projekt, ss. konflikter (se avsnitt 3.4.3), problem med kravanalys (Lindgaard m.fl., 2006) och motivationsproblem (se avsnitt 3.4.5) vilket stödjer studiens validitet.

Vår studie är en av få studier där man ändrar formen av användarmedverkan under projektets gång<sup>3</sup>. Detta gör att vi får en unik möjlighet att jämföra två metoder utförda på samma projekt.

Det är intressant att i projektets första fas, där användaren designade själv med hjälp av en interaktionsdesigner gick samarbetet mycket bättre och projektet framskred fortare än i projektets senare fas, där användaren inte medverkade i gränssnittsdesignen.

Enligt Gulliksen och Göransson's anvisningar för användbarhetsdesign (se avsnitt 3.5.1) ska användarna delta i alla projektets faser, så som analys, design, utvärdering och konstruktion vilka skall itereras genom hela projektet. I projektets senare skede deltog användaren inte i själva designarbetet, även om hans åsikter skulle guida designen. Detta ledde till stora problem vilket

---

<sup>3</sup> Vi kunde inte hitta någon annan.

styrker Gulliksens påstående att uteblivande av att låta användare medverka i alla faser i systemutvecklingsprojekt kan bidra till att användarcentrerade projekt misslyckas.

En del av problemen kan möjligen spåras till metoden för användbarhetsdesign. Projektdeltagarna tyckte att metoden inte var tillräckligt tydlig. Till exempel fanns det ingen vägledning till hur man skulle dokumentera och presentera resultaten från kravanalysen, så att den kunde bli översatt till gränssnittsdesignen på ett enkelt sätt. Det fanns inte heller några anvisningar till hur kravspecifikationen skulle byggas upp och användas. Detta är ett problem för många användarcentrerade metoder (Lindgaard m.fl., 2006) och det pågår arbete för att få användarcentrerade metoder att integrera med icke-användarcentrerade kravspecifikationstekniker (Requirements Engineering).

En tydligare och mer formell kravspecifikation skulle möjligtvis kunna förhindra kommunikationsproblem av det slag som kom upp i detta projekt. Den skulle också göra det möjligt för en större del av designarbetet att äga rum utan användaren. Deltagarna i denna studie tyckte dock att användarens medverkan bidrog till en smidigare designprocess och användbarhetstester visade på bättre användbarhet när användaren deltog i hela utvecklingsprocessen vilket tyder på att utvecklare borde sträva efter att inkludera användare i hela utvecklingsprocessen.

Konflikt i sig är inte av ondo, den kan till och med gynna användbarheten, förutsatt att den löses på ett tillfredsställande sätt (se avsnitt 3.4.3). I detta projekt blev det dock ingen lösning av problemen och projektet led för det.

Varför blev det ingen lösning, vad gjorde de fel? Utvecklarna reagerade på användarens kritik genom att bli defensiva. De tyckte att användaren var otydlig och jobbig. De analyserade inte orsakerna bakom användarens kritik och vidtog inte några speciella åtgärder för att säkerställa att deras tolkning av användarens meddelande var rätt. När användaren kände att hans kritik inte ledde till förändring tycks han ha givit upp.

Det verkar som utvecklarna skulle behöva en kurs i kommunikation! Det finns metoder för att minska risken för missförstånd. Inom psykologin finns det kommunikationstekniker (active listening) som bygger på att deltagarna i kommunikationen återger sin tolkning av meddelandet för att ge den som skickade meddelandet chans att rätta feltolkningar. Dessa tekniker har använts i organisationer med ett lyckat resultat (Helms Marilyn & Haynes Paula, 1992). De skulle också kunna användas i systemutveckling för att förebygga missförstånd och hantera konflikter. Enligt Hägerfors (1995, s.170) kan missförstånd förhindras genom att ofta fråga om man uppfattat rätt. Detta är samma teknik som active listening.

Denna studie visade därför att kommunikationsfärdigheter i projektgruppen är av stor vikt i användarcentrerad systemutveckling. Det skulle kunna vara fördelaktigt att få in en kommunikationscoach i början av och även under projektets gång för att förhindra brister i kommunikationen och hjälpa till med att lösa konflikter. Utbildningen av systemvetare skulle kunna innehålla praktisk kommunikationsteknik för att förbereda dem för framtida relationer med användare.

I de flesta systemutvecklingsmetoder finns det verktyg för att minska risken för missförstånd, till exempel UML diagram, processdiagram och andra verktyg som i viss utsträckning kan hjälpa användarna att förstå utvecklarnas föreställningar om hur systemet ska se ut och fungera. Enligt Gulliksen och kollegor är dessa dock ofta svårförståliga för användarna. Gulliksen föreslår en rad andra verktyg som kan hjälpa användaren att få en konkret bild och förståelse över hela den

framtida användningssituationen med det nya IT-systemet, se avsnitt 3.4.2. I detta projekt användes endast prototypen som analysverktyg och när användaren var där under designen gick detta bra. Men när användaren inte längre deltog i designen borde de ha övergått till enklare verktyg så som skisser på papper och mock-up:er. Om de hade gjort detta är det sannolikt att det hade blivit färre missförstånd.

Eftersom risken för missförstånd är så stor och kan skapa så många problem är det viktigt att den påpekas för projektdeltagarna redan från början. Detta skulle kunna göra dem mer medvetna i sin kommunikation och mer overseende med varandra när missförstånd dyker upp.

## 6 Slutsatser

Det ena målet med studien var att få kunskaper om hur användarmedverkan påverkar systemutveckling och systemkvalitet.

I denna studie var användarmedverkan av stor nytta i alla delar av utvecklingsprocessen som behandlades, dvs. initial kravanalys, iterativ analys och design av databas och gränssnitt samt utvärdering av prototyper. Tydligast var nyttan av användarmedverkan under den iterativa designen av gränssnittet, eftersom utvecklingsprocessen gick sämre och prototypernas användbarhet försämrades när användaren inte längre deltog i denna. Detta betyder dock inte att användarmedverkan var av mer nytta i denna fas, utan endast att studiens utformning var på så sätt att de faktorer som hade betydelse för denna fas lyftes fram, eftersom de kunde utrönas av den typ av problem som uppstod vid metodbytet.

Vi fann också att nyttan av användarmedverkan påverkades av vissa faktorer, vilket var det andra målet med studien. De faktorer som påverkade nyttan av användarmedverkan i detta projekt var följande:

*Utvecklingsmetoden.* Den faktor som var mest avgörande i denna studie var utvecklingsmetoden. I den förra delen av projektet deltog användaren i hela designfasen. I denna del var nyttan av användarmedverkan stor och ledde till smidig utveckling och bra användbarhet hos prototyperna. I den senare delen deltog användaren endast i utvärdering och analys men inte i design. I denna del gick systemutvecklingen sakta och prototyperna hade sämre användbarhet. I användarcentrerad utveckling är det därför viktigt att användaren deltar i hela designprocessen, inte bara i analys och utvärdering. Vi drar de slutsatser av denna studie att metoden användbarhetsdesign inte innehåller tillräckligt utförlig metodik för analys för att design utan användare ska vara lyckad. Design med användare går däremot hyfsat bra med denna metod.

*Utvecklarnas kommunikationsfärdigheter.* En bidragande orsak till de skildrade problemen var utvecklarnas bristfälliga kommunikationsfärdigheter. Utvecklarna verkade inte vara medvetna om den stora risken för missförstånd som muntlig kommunikation innebär. De använde inga metoder för att säkerställa att användarens meddelande tolkades rätt. Detta gjorde att användarens förslag inte nådde fram till prototypen vilket ledde till konflikter

*Maktfördelning* I förra delen av projektet, då nyttan av användbarheten var stor, hade användaren, som också var beställare av systemet, större makt än utvecklarna. I den senare delen minskade hans makt så att maktfördelningen var jämnare var nyttan av användarmedverkan liten. Vi vill dock inte gå så långt att säga att användare borde ha större makt i systemutveckling än utvecklare, även om det gick bra i detta fall. Men det är tydligt att en jämn maktfördelning kräver större kommunikationsfärdigheter och skicklighet i lösning av konflikter. Om dessa inte är bra är därför en hierarkisk maktfördelning att föredra. I vårt fall, där det rör sig om en expertanvändare med beställarkompetens gick det mycket bra att användaren hade större makt än utvecklarna.

*Motivation* Till följd av kommunikationsproblemen uppstod det motivationsbrist hos alla deltagare. Denna motivationsbrist verkar ha orsakat att de gav upp och la ner projektet. Ingen av dem var kompenserade på något sätt för deltagande i projektet. Det kan ha varit en bidragande orsak till att deltagarna inte lade ner tid på att utveckla strategier för att hantera kommunikationsproblemen.



Den slutgiltiga slutsatsen blir därför att studien stödjer nyttan av användarmedverkan i systemdesign men nyttan påverkas bland annat av utvecklingsmetoden, utvecklarnas motivation och kommunikationsfärdigheter samt maktfördelningen inom projektgruppen.

## 7 Framtida forskning

Studien antyder att brister i kravanalysen kan leda till missförstånd och samarbetsvårigheter mellan utvecklare och användare. Det skulle vara intressant att studera kravanalysen i projekt där samarbetet mellan utvecklare och användare inte går så bra.

- Finns det samband mellan dålig kravanalys och dåligt samarbete med användare?
- Borde samarbetsproblem i systemutvecklingsprojekt utredas med hänsyn till brister i kravspecifikationen (eller omvänt)?
- Hur går samarbetet i projekt i iterativ systemutveckling där användare utvärderar men inte designar?

Det vore intressant att se flera studier där man växlar mellan användarmedverkan i designfasen och endast i utvärderingsfasen.

Vi vill tacka vår handledare, Hans Lundin, för utomordentligt tålamod, hjälpsamhet och skicklig handledning. Det är din förtjänst att denna uppsats blev verklighet och vi är dig evigt tacknömliga.

Vin och Danisa

## 8 Referenser

- Allwood, C. M. (1998) *Människa-datorinteraktion: Ett psykologiskt perspektiv*, Lund, Studentlitteratur.
- Alter, S. (2002) *Information Systems. The Foundation of E-business*, New Jersey, Prentice Hall and Pearson Education International.
- Asgeirsson, D. (2004) muntlig kommunikation.
- Balter, A. (2003) *Alison Balter's Mastering Microsoft Access 2003* New York, Macmillan Computer Pub.
- Barki, H. & Hartwick, J. (1989) Rethinking The Concept Of User Involvement. *MIS Quarterly*, 13, 53-64.
- Barki, H. & Hartwick, J. (1994) Measuring User Participation, User Involvement, and User Attitude. *MIS Quarterly*, 18, 59-82.
- Bryman, A. (1997) *Kvantitet och kvalitet i samhällsvetenskaplig forskning*, Lund, Studentlitteratur.
- Davidson, J. E. (1999) Joint application design (JAD) in practice. *The Journal of Systems and Software*, 45, 215-223.
- Ehn, P., Meggerle, T., Steen, O. & Svedemar, M. (1995) What Kind of Car is this Sales Support System? *Proceedings of the Third Decennial Computers in Context Conference*. Århus, Denmark.
- Gallivan, M. J. & Keil, M. (2003) The user-developer communication process: a critical case study. *Information Systems Journal* 13, 37-68.
- Garrod, S. & Anderson, A. (1987) Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27, 181-218.
- Gould, J. D., Boies, S. J. & Ukelson, J. (1997) How to Design Usable Systems. i M. Helander & T. K. Landauer (Red.) *Handbook of Human-Computer Interaction*. Amsterdam, Elsevier Science B.V.
- Gulliksen, J. & Göransson, B. (2002) *Användarcentrerad systemdesign* Lund, Studentlitteratur.
- Gulliksen, J., Lantz, A. & Boivie, I. (1999) User Centered Design in Practice - Problems and Possibilities. *CID-40*. Stockholm, KTH.
- Göransson, B. & Gulliksen, J. (2000) *Användarcentrerad systemutveckling, version 1.0*. Stockholm, Kungliga Tekniska Högskolan.
- Helms Marilyn, M. & Haynes Paula, J. (1992) Are you really listening? The benefit of effective intra-organizational listening. *Journal of Managerial Psychology*, 7, 17-22.
- Hunton, J. E. & Beeler, J. D. (1997) Effects of User Participation in Systems Development: A Longitudinal Field Experiment. *MIS Quarterly*, 21, 359-388.
- Hägerfors, A. (1995) *Att samlära i systemdesign.*, Lund, Studentlitteratur.
- Ifudu, O. (1998) Care of Patients Undergoing Hemodialysis. *The New England Journal of Medicine*, 339, 1054-1062.
- ISO\_9241-11 (1998) Definition av begreppet användbarhet., <http://www.usabilitypartners.se/usability/standards.shtml>.
- Kalén, T. (1997) *Usability in context: A psychological and social-pragmatic perspective*, Vasastadens Bokbinderi AB.
- Kvale, S. (1996) *Interviews : an introduction to qualitative research interviewing*, Thousand Oaks, Calif, Sage.
- Lin, W. T. & Shao, B. B. M. (2000) The relationship between user participation and system success: A simultaneous contingency approach. *Information & Management*, 37, 283-295.

- Lindgaard, G., Dillon, R., Trbovich, P., White, R., Fernandes, G., Lundahl, S. & Pinnamaneni, A. (2006) User Needs Analysis and requirements engineering: Theory and practice. *Interacting with Computers*, 18, 47-70.
- Löwgren, J. (1993) *Human-computer interaction*, Lund, Studentlitteratur.
- Marengi, C. (1982) James Martin Talks About the Data Base Environment. *Computerworld*, 16, In Dept-12.
- Mayers, J., Markell, M., Cohen, L., Hong, J., Lundin, P. & Friedman, E. (1992) Vascular access surgery for maintenance hemodialysis: Variables in hospital stay. *ASAIO J* 38, 113-115.
- Merriam, H. B. (1994) *Fallstudie som forskningsmetod*, Lund, Studentlitteratur.
- National\_Kidney\_Foundation (2000) K/DOQI Clinical Practice Guidelines for Vascular Access. *Am J Kidney Dis*, 37, S137-S181.
- Nielsen, J. (1993) *Usability engineering*, San Francisco, Morgan Kaufmann.
- Ottersten, U. & Berndtsson, I. (2002) *Användbarhet i praktiken : praktiska handgrepp, grundbegrepp och tankemodeller*, Studentlitteratur (Danmark).
- Saleem, N. (1996) An empirical test of the contingency approach to user participation in information systems development. *Journal of Management Information Systems*, 13, 145-167.
- Schultz, U. (2007/06/05) lärare i HCI vid Hamburg University of Applied Sciences. <http://www.bui.fh-hamburg.de/pers/ursula.schulz/webusability/evaluation.html>.
- Srivihok, A. (1999) Understanding executive information systems implementation: an empirical study of EIS success factors. *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, 10 pp.-10 pp.
- Standish\_Group (1995) The CHAOS report. Tillgänglig på <http://www.standishgroup.com>.
- Standish\_Group (2003) The CHAOS report. Tillgänglig på <http://www.standishgroup.com>.
- Sugar, W. A. (2001) What Is So Good about User-Centered Design? Documenting the Effect of Usability Sessions on Novice Software Designers. *Journal of Research on Computing in Education*, 33, 235-251.

## 9 Bilagor

### 9.1 Observationer

#### 9.1.1 Observation 1

Datum: 2005-05-14 , kl: 13:15

Plats: Daniels kontor på njuravdelningen

Personer: Daniel (användare, läkare), Vin (observatör)

Verktyg: Skrivblock och penna

Uppgift: inmatning av blod accesser i VAD systemet.

(användarkommentarer inom parenteser)

Daniel sitter vid sitt skrivbord. Han använder sin bärbara dator och det finns inte mycket plats för honom vid skrivbordet. Han har dokumenten i en mapp som han har på höger sida. Det verkar lite jobbigt. Han öppnar applikationen och access inmatningsformen. Den öppnas på Access fliken. Han tittar på den, trycker på ”+” tecknet för att se ingrepp, detta tar i allt 30 sekunder. Öppnar fliken Access detaljer, tittar 10 sekunder. Öppnar alla fliker, det tar 5 sekunder. I allt tar detta 45 sekunder.

Han hittar dokumentet för första patienten i mappen, 10 sekunder. Han trycker direkt på ny patient (jag vet att detta är en ny patient). Inmatningsformuläret för ny patient öppnas. Han matar in uppgifter om patient, 1 minut. Han stänger formuläret för ny patient. Matar in under Access fliken, 10 sek. I allt tar detta 1 min, 10 sek.

Inmatning 1:

Vänder sida i mappen. Skriver i ”Sök patient” inmatningsruta, söker efter namn, patient finns inte i databasen. Trycker på ”Ny patient”, matar in, det tar 30 sek. Han lämnar ”Huvuddiagnos” fältet (jag vet inte huvuddiagnosen). Matar in i Access fliken, 10 sek. Öppnar Access Funktion flik (jag måste kolla spetsläget på denna). Vänder sig till höger mot sin stationära dator, öppnar Melior, hittar patientens journal, kollar en minut, hittar inte. Loggar in sig på Spektra system för labbresultat. Söker efter patientens personnummer, hittar labbresultat för datum efter operation. Vänder sig till vänster mot sin egen bärbara dator , skriver in spetsläget. Detta tar 1 minut. I allt tar detta 2 min 40 sek.

Inmatning 2:

Vänder sida i mappen. Skriver i ”Sök patient” inmatningsruta, söker efter namn. Hittar patient. (det är jobbigt att man måste trycka på nedåt pilen för att markera hela rutan, annars måste man radera förra namnet. Hela namnet borde markeras när man klickar i den). Matar in under Access fliken, 30 sek. (det är problem om man trycker på ”enter” efter sista fältet, då kommer man till en ny post och om man går till annan flik då, då tillhör den inte accessen som man matat in). Öppnar Access Funktion flik (nu måste jag kolla spetsläge igen). Vänder sig till höger mot sin stationära dator. Loggar in sig på Spektra system för labbresultat. Söker efter patientens personnummer, hittar labbresultat för datum efter operation. Vänder sig till vänster mot sin egen bärbara dator , skriver in spetsläget. Detta tar 20 sekunder. I allt tar detta 50 sek.

Inmatning 3:

Vänder sida i mappen. Skriver i ”Sök patient” inmatningsruta, söker efter namn. Hittar inte. Trycker på ”Ny patient”, (vad står där egentligen), Vänder sig till höger mot sin stationära dator, öppnar Melior, hittar patienten, matar in, det tar 10 sek. Öppnar Access Funktion flik. Vänder

sig till höger mot sin stationära dator, öppnar Melior, hittar patientens journal, hittar spetsläget, det tar 20 sek (Spektra är bekvämare). Matar in i Access Funktion, 10 sek. I allt tar detta 40 sek.

#### Inmatning 4:

Vänder sida i mappen. Skriver i "Sök patient" inmatningsruta, söker efter personnummer. Hittar patient. (jag söker oftast efter namn, för det är lättare men man måste passa att alltid kolla att personnumret stämmer). Matar in under Access fliken.. Trycker på "+" tecknet, ingrepp post öppnas. Matar in. (Det är problem att under ingrepp är operatör inte kopplad till namnlistan i access fliken). I allt 40 sekunder.

#### Inmatning 5:

Vänder sida i mappen. Skriver i "Sök patient" inmatningsruta, söker efter personnummer. Hittar patient. Matar in under Access flik. (Denna patient har fått ny fistel. Det betyder att hans äldre fistel är nerlagd, men vi vet inte när.). I allt 60 sek.

#### Inmatning 6:

Vänder sida i mappen. Skriver i "Sök patient" inmatningsruta, söker efter namn, patient finns inte i databasen. Trycker på "Ny patient", (jag hatar att mata in ny patient), matar in, det tar 30 sek. Matar in i Access fliken, 10 sek. (man borde inte behöva klicka med musen för att komma till första fältet i Access fliken). (Här har vi igen gammal fistel som vi vet inte när borttagen). Öppnar Access Funktion flik. Vänder sig till höger mot sin stationära dator, öppnar Melior, hittar patientens journal, kollar en minut, hittar inte. Loggar in sig på Spektra system för labbresultat. Söker efter patientens personnummer, hittar labbresultat för datum efter operation, 1 minut. Vänder sig till vänster mot sin egen bärbara dator, skriver in spetsläget. 20 sek. I allt tar detta 2 min.

#### Inmatning 7:

Vänder sida i mappen. Skriver i "Sök patient" inmatningsruta, söker efter namn, patient finns inte i databasen. Trycker på "Ny patient", matar in, (jag vet inte huvuddiagnosen på denna heller) det tar 30 sek. Han lämnar "Huvuddiagnos" fältet tomt. Matar in i Access fliken, 10 sek (jag kommer inte ihåg namnet på denna accesstyp, måste kolla). Vänder sig till höger mot sin stationära dator, öppnar Internet, skriver in adressen för UpToDate hemsidan, söker efter det som han tror att den heter, hittar inte. Skriver in adressen för Yahoo, söker efter det som han tror att den heter, hittar den. 1 min. Vänder sig till vänster mot sin egen bärbara dator, skriver in spetsläget, öppnar Access Funktion flik, skriver in spetsläge. I allt 1min 50 sek.

#### Inmatning 8:

Vänder sida i mappen. Skriver i "Sök patient" inmatningsruta, söker efter namn, patient finns inte i databasen. Trycker på "Ny patient", matar in. Stänger fönstret. (Jag vet att det sparas men det känns konstigt att bara stänga fönstret utan att spara. Kanske vi borde ha en spara och stäng knapp?). Matar in i Access flik. Stannar i AccessTypII, (det borde vara inmatad på plats). Lämnar den tom.(Nu är jag trött) Öppnar Access Funktion flik, hittar inte Access sida. Lämnar den tom. I allt 2 min 30 sek.

Denna observation har tagit 30 minuter.

#### Noteringar:

1. Han använder mest flikarna Access och Access funktion, ibland ingrepp. Aldrig access detaljer. Han har förklarat att någon måste gå igenom alla poster, och hitta Access detaljer i patientens Vaskulär Access mapp.
2. Det är problem att ibland får patienter nya vaskulära accesser men man vet inte när den gamla var nerlagd. Man måste söka i:
  - a. Melior
  - b. Patientens fisteljornal
  - c. Röntgen, Spektra

Detta måste göras i efterhand.
3. Ibland är spetsläget på katetern i venen skriven i dokumentet, men oftast inte, då behöver användaren kolla i Spectra eller Melior.
4. Han söker efter namn i 6 av 8 fall.
5. accessTypII är inte bra namn, han har namngett den själv. Han vet själv vad han menar men ingen annan kan. Den ska innehålla en mer detaljerat typ av access än vad som står i accessTyp om det behövs.
6. accessTypII är också problematisk därför att den är svår att få reda på. Det borde vara skrivit på dokumentet i samband med operationen. Han hade gjort en blankett för att fylla i , i samband med operationen, men den används inte, utan de skriver på nedersta delen av dokumentet . (operations förfrågan?)
7. Det är problem att under ingrepp är operatör inte kopplad till namnlistan i access fliken. Detta är för att han ville först inte ha operatör under ingrepp fliken. Men ändrade sig och la till den själv.
8. Access Detaljer fliken la han till själv och den ser inte bra ut. Se heuristik analys.

## 9.2 Enkät:

### 9.2.1 Frågor om systemet

#### *Kvalitet (effektivitet):*

Vad var enligt din uppfattning syftet med projektet ”System för registrering av blodaccesser”? Att samla statistik över blodaccesser som kan användas till att förbättra access överlevnad och vård. Också att ha lättillgänglig information för vårdpersonal för sitt arbete, jmf elektronisk medical journal.

Hur tycker du att det har gått att uppnå vårt syfte med projektet?

Sådär. Tyvärr hade jag inte lika mycket tid avlagt som behövde. Mer tid behövdes för att sätta er in i komplexa vårdprocesserna. Det gjorde jag visserligen i början men självklart behöver en kunnig att följa efter att alla ”subalties” kommer fram i systemet. Om denna iterativa feedback inte kommer är det nästan klart att det inte kommer att vara optimalt. Blir mer som ett bokföringssystem i en målningssaffär men fångar inte de komplexa ”business processer”.

Vad hade enligt din mening bidragit till ett bättre resultat?

Att jag hade haft mer tid. Självklart kunde vi ha detaljerad krav specifikation men det verkade så svårt att göra ett. Sannolikt fel av oss att inte prova. Vi vet till nästa gång. Men samtidigt är så många möjligheter som man inte ser förr än i projektets utveckling, därför är enl. min uppfattning nödvändigt att förutom en "ram kravspecifikation" med allt som man kommer ihåg från början att ha en "iterativ kravspecifikation" kontinuerligt under projektets gång.

Vad tycker du om det nya systemet?

Jag tycker faktiskt det är mycket bra. Det har dock mycket kvar för att innehålla allt som var planerat från början. Som det är nu då har det "served its purpose" i att titta närmare på användarmedverkan vid systemutveckling. Jag har faktiskt lärt mig mycket av det. Både om var användare ska komma in mer och var han ska inte komma in. Men användare är olika och vissa ska hålla sig helt borta men med mer närvaro av vissa blir systemet bättre. Bäst är självklart om användaren är systemutvecklare som sällan är.

*Efficiency:*

Hur har det nya systemet påverkat tidsåtgången i registrering av blodaccesser?

Svårt att säga, systemet "ligger på is" för närvarande. Sjuksköterskor matade in i det under ett tag. Det var inget problem med inmatning men själva projektet behövde mer förankring i själva organisationen. Vi var bara 2 läkare som drev detta men andra hade inte kännedom om. Meningen var att sprida detta vidare när systemet var mer invecklad och inkörd men till det stadium kom vi aldrig och därför inmatande sjuksköterskor slutade göra detta när vi båda läkare lämnade dialysen under några månader.

Hur har systemet påverkat din prestation på arbetet?

Oklart vad menas med frågan... Med ändringar av denna natur är alla emot i början och ser inte möjligheterna för en det har givit resultat. Tyvärr har vi inte kommit dit ännu men det är planerat att göra det och har jag sökt anslag för att kunna lägga tid åt att sätta samman statistik över det som redan är insamlat och fortsätta utvecklingen. Oklart hur det blir med allt annat som konkurrerar med mans tid.

### 9.2.2 Frågor om metoden

Kan du beskriva metoden som vi använde i projektet?

Så koncentrerad användarmedverkan som kan vara. Iterative, oplanerad, ostrukturerad, tyvärr "messy"

Vad tycker du om den metod vi använde?

Bra om vi hade haft mer tid. Skulle lägga när mer tid på kravspecifikation, både i början och iterativt under utvecklingen då man kommer på nya idéer/lösningar.

Vad tyckte du var bra med metoden?

Att jag som användare såg de möjligheter och begränsningar som programmering och databas och gränssnitt har. På det sättet kommer fram mer realistiska lösningar på många processer som funktions önskemål/krav från layman användare kan vara för "höga". Också kan jag se



möjligheter att göra något på ett annat och enklare eller effektivare sätt än layman användare kan som hade varit nöjd med en suboptimal lösning.

Vad tycker du var mindre bra?

Att den icke var tillräckligt strukturerad.

Hur tycker du att metoden har påverkat systemets kvalitet?

Netto sätt dåligt. Men den hade potential att ge mycket positiv synergism.

### 9.2.3 Frågor om kommunikation

Hur tycker du att samarbetet har gått?

Samarbetet gick bra när jag var involverad och hade tid. När det minskade, blev flera missuppfattningar om hur något skulle vara som ledde till irritationer när mycket tid lagdes när på något som inte spelade roll eller vise versa. Min stora lärdom i detta är att släppa tagit och låt andra göra det som de är bra på. Jag hade för mycket ambition att lära mig

Hur upplevde du maktförhållandet mellan dig och andra i gruppen?

Då jag var ”beställaren” och användaren och till stor del utvecklaren låg nästan all makt på min sida. Jag är övertygad om att hade jag haft mer tid hade vi haft ett väldigt fint system idag. Jag vet också att om jag hade lämnat ifrån mig min ”makt”, gjort en bra kravspecifikation och inte varit så obsessiv om att kunna göra allt själv, hade systemet varit längre kommit idag.

Hur upplevde du din insats i utvecklingen?

Jag tror att min insats har varit stor. Det är alltid svårt att säga vem gör vad när flera sitter framför en datorskärm och pekar och säger typ ”nej, det skulle vara bättre så ....” etc.

Hur tycker du att samarbetets kvalitet har påverkat systemets kvalitet?

I början var mycket spänning och entusiasm om systemets möjligheter och denna utveckling, då var allt mycket trevligt. Då jag kopplades ifrån och flera missförstånd kom, kom fram irritation och frustration på bägge parter som sannolikt har ofta hindrat bättre lösningar eller att utvecklarna var bättre insatta i någon process som de skulle designa.