



Informationssystem 120p  
Institutionen för Programvaruteknik och Datavetenskap  
DVC001 – Kandidatarbete – våren 2003

## Kommunikation med hjälp av mock-uper

– En studie av fastighetssystemet för Svenska kyrkan i Ronneby –

Av:  
Maria Alsén, is00mal@student.bth.se  
Nils Järgenstedt, is00nja@student.bth.se  
Handledare: Hans Kyhlbäck  
Examinator: Guohua Bai

## Abstract

In several cases, systems that have been developed have been very time consuming and cost a lot of money, but they still do not fulfil the users requirements and requests. To make new systems better, you have to find a way to communicate that allows the developers to understand the needs of the user.

The aim for our thesis is to highlight the importance of communication in system development. To investigate this we have chosen to do a study of the real-estate system. The work methods that have been used include mock-ups and informal conversations with the user, who is employed by the Church of Sweden in Ronneby. The purpose of this thesis is, among other things, to provide the Church of Sweden in Ronneby with a report, which can be of help for further development of the system. The system was developed by a work group at Blekinge Institute of Technology. Our question of issue is: *Would the Church of Sweden in Ronneby obtain a more useful system if the developers applied the guidelines and experience that exists within the area of HCI? Further more we have looked in to if the communication has improved with the use of mock-ups to increase the interaction?*

Human-computer interaction as a term was adopted in the mid-1980s as a means of describing this new field of study. The focus of interest described the new way of looking at the interaction between computers and people. In this field it is important to involve the user in the developing process from the beginning to the end. Together with the user, we have made some propositions based on previous documentation from the project. Mock-ups are a prototype of paper that shows the user what the system will look like. This method was applied together with the user, to find a more logical structure for the system.

After gathering the result of the case study, we can honestly say that communication do increase because of mock-ups. Good communication is not something that you can learn from books; you have to adjust to the situation. It is important when you develop a system to work with a model that allows you to go back and change things in previous phases that are incorrect, and also to include the user in the whole process.

---

## Sammanfattning

Idag finns utvecklade system som kostat såväl tid som stora mängder pengar att utveckla, men som ändå inte uppfyller användarens krav och önskemål. För att göra nya system bättre måste man hitta en bra kommunikation som gör att utvecklarna ser användarnas behov.

Målet med vårt arbete är att belysa hur viktig kommunikation är vid utveckling av system. För att undersöka detta har vi valt att göra det som en studie av ett fastighets-system. De metoder som vi har använts oss av är mock-uper (förenklad pappersmodell) och informella samtal tillsammans med användaren för Svenska kyrkan i Ronnebys fastighetssystem. Syftet med arbetet är bland annat att Svenska kyrkan i Ronneby ska få en rapport som kan vara till hjälp vid vidareutveckling av systemet, som utvecklats av en projektgrupp vid Blekinge Tekniska Högskola. Vi har arbetat utifrån frågeställningen: *Skulle Svenska kyrkan i Ronneby få ett mer användbart informationssystem om systemutvecklarna hade använt sig av riktlinjer och erfarenheter som bearbetas inom området HCI?* Mer specifikt har vi också undersökt frågan: *Hade kommunikationen förbättras med hjälp av mock-uper för att öka interaktionen?*

Termen HCI, Human-computer interaction, togs i bruk i mitten av 1980-talet och beskrev det nya tankesättet att se på interaktionen mellan datorer och människor. Inom HCI betonas vikten av att arbeta användarcentrerat vid utveckling av system. Därför ska man låta användaren vara involverad under hela utvecklingsprocessen. De förslag som vi arbetat fram tillsammans med användaren för systemet har gjorts med utgångspunkt från tidigare projektdokumentation. Mock-uper som vi använt oss av är en förenklad pappersmodell för hur system kommer att se ut. Vi har gjort mock-uper för att hitta en mer logisk struktur för sidor i fastighetssystemet. Dessa mock-uper har vi gjort tillsammans med användaren.

Efter att vi sammanställt vårt resultat har vi kommit fram till att kommunikationen ökar med hjälp av mock-uper. Bra kommunikation är inte något man kan läsa sig till utan måste anpassas efter situationen. Vid utveckling är det också viktigt att jobba med iterativa systemutvecklingsmodeller och att kunden finns med i hela processen.

---



## Innehållsförteckning

<b>INLEDNING .....</b>	<b>1</b>
MÅL OCH SYFTE .....	1
PROBLEMFÖRMULERING.....	1
<i>Avgränsningar</i> .....	2
METODBESKRIVNING.....	2
<b>BAKGRUND.....</b>	<b>4</b>
HISTORIA.....	4
<i>Systemutveckling i Skandinavien</i> .....	5
HCI .....	6
UTVECKLINGSMODELLER .....	7
<i>Vattenfallsmodellen</i> .....	7
<i>Livscykelmodellen</i> .....	9
MOCK-UP .....	10
FASTIGHETSSYSTEMETS KRAVSPECIFIKATION .....	11
FUNKTIONELLA KRAV.....	11
<b>RESULTATBESKRIVNING .....</b>	<b>13</b>
FÖRSTA MÖTET MED ANVÄNDAREN AV FASTIGHETSSYSTEMET .....	13
STRUKTUREN ÄNDRAS INFÖR ANDRA MÖTET .....	14
ANDRA MÖTET MED ANVÄNDAREN .....	16
MER STRUKTUR PÅ UNDERSIDORNA INFÖR TREDJE MÖTET .....	17
HUR ANVÄNDAREN AGERADE.....	22
<b>DISKUSSION AV RESULTATET .....</b>	<b>24</b>
FÖRSLAG PÅ FÖRÄNDRINGAR .....	25
SLUTORD .....	25
FRAMTIDA FORSKNINGSOMRÅDEN .....	26
<b>LITTERATURFÖRTECKNING.....</b>	<b>27</b>
LITTERATUR .....	27
OTRYCKTA KÄLLOR.....	27
SAMMANFATTNING AV KRAVSPECIFIKATION.....	BILAGA 1
SAMMANSTÄLLNING AV INTERVJUFRÅGOR TILL ANVÄNDAREN .....	BILAGA 2

## Inledning

Ett grundligt förarbete anser vi är viktigt när datoriserade system ska utvecklas. Med ett genomtänkt arbete kan företag spara både pengar och tid. Systemutvecklare som slarvar med förarbete är inte medvetna om vilka risker det innebär för slutresultatet. Idag finns många system som kostat såväl tid som stora mängder pengar, men som ändå inte uppfyller användarens krav och önskemål. Kunskaper och erfarenheter inom interaktionsdesign är relativt nytt inom systemutveckling. För att göra system bättre måste utvecklarna sätta sig in i och se mera till användarnas behov och önskemål. Man måste hitta ett bra sätt att kommunicera på.

Vi har valt att utgå från hur litteraturen i detta ämne beskriver principer för hur system ska utvecklas. Som utgångspunkt för vår undersökning har vi valt ett fastighetssystem som utförts som projekt vid Blekinge Tekniska Högskola åt Svenska kyrkan i Ronneby. Detta har vi gjort för att det är lättare att analysera ett existerande system och binda sina kunskaper till detta.

Med hjälp av arbetet önskar vi bredda det datavetenskapliga ämnet och öka förståelsen av att kommunikation mellan användare och utvecklare är viktig.

## Mål och syfte

Vårt mål med arbetet är att belysa hur viktig kommunikation är vid utveckling av system. Med hjälp av god kommunikation kan misstag undvikas. Målet är även att se hur projektet vid Blekinge Tekniska Högskola, som utförts åt Svenska kyrkan i Ronneby, hade kunnat utvecklas på ett annat sätt om utvecklarna hade använt sig mer av tips och utvecklingsprinciper som bearbetas inom ämnet HCI. Syftet med detta arbete är följaktligen, dels att Svenska kyrkan i Ronneby ska få en rapport som kan vara till hjälp vid vidareutveckling av systemet och dels att utvidga vårt och andras lärande om systemutveckling. Innehållet i rapportern kommer att beröra delarna förändringsanalysen och analysen i Erling S Andersens Livscykelmodell<sup>1</sup> som vi kommer att jämföra med vattenfallsmodellen<sup>2</sup>. Den sistnämnda har utvecklarna av Svenska kyrkan i Ronnebys fastighetssystem använt sig av. Modellerna kommer att förklaras senare i arbetet.

## Problemformulering

Vi tror att med kunskaper inom HCI kan utvecklare producera mer användbara system för användaren.

Vår frågeformulering lyder som följande:

---

<sup>1</sup> Andersen, Erling S, (1994) s 41

<sup>2</sup> Sommerville, Ian, (2001) s 45

*”Skulle Svenska kyrkan i Ronneby få ett mer användbart informationssystem om systemutvecklarna hade använt sig av riktlinjer och erfarenheter som bearbetas inom området HCI?” Mer specifikt har vi undersökt frågan: ”Hade kommunikationen förbättras med hjälp av mock-uper för att öka interaktionen mellan användare och utvecklare?”*

## **Avgränsningar**

Vi begränsade oss till systemet som utvecklats som projekt av Blekinge Tekniska Högskola för Svenska kyrkan i Ronneby, som vi i arbetet kommer att benämna ”Fastighetssystemet”. Alla system kan inte utvecklas på samma sätt då alla situationer är olika. Det finns inte någon mall för hur man utvecklar det perfekta systemet. Istället får man anpassa sig till situationen. Vi har valt detta system därför att det är utvecklat av personer med ungefär samma kunskaper som oss själva. Vi har haft möjlighet att intervjua både beställare och utvecklare. Vår studie kommer att utgå från förändringsanalysen och analysen i Erling S Andersens livscykelmodell. Utformning, realisering, implementering, förvaltning och drift eller avveckling kommer vi inte att ta upp. Projektgruppens dokumentation och våra möten med användaren har varit vår grund för att analysera, utveckla och bygga vidare på det befintliga systemet. Vi har begränsat oss till att bara titta på strukturen och inte på buggar<sup>3</sup> som finns i systemet. Vi behandlar inte heller färgsättning, såvida den inte är en viktig del för att markera olika val som görs i applikationen.

## **Metodbeskrivning**

För utveckling har vi strukturerat vårt arbete i olika delar och ordningen för dessa presenteras i det följande:

1. För att få mer inblick i tankesättet bakom utvecklingen av detta fastighets-system har vi studerat den litteratur som finns inom HCI-ämnet. Hur kommunicerar utvecklare på bästa sätt med kunden/användaren av systemet?
2. Vi satte oss in i projektdokumentationen för fastighetssystemet för att se hur det gamla systemet fungerade. Vi har även kompletterat med mer informella samtal och intervjuer med användaren. Information har vi också fått ur projektdokumentationen om hur fastighetssystemet fungerar och används.
3. Kundens alla önskemål, enligt projektgruppens kravspecifikation, tillsammans med alla ”nya” krav som kommit upp vid möten med användaren har legat till grund för våra mock-uper. Vi valde att göra mock-uper av pappersdelar, som var flyttbara, istället för statiska dataprototyper. Detta för att kunna integrera användaren i utvecklingen. Metoden valde vi för att det är lättare att få användaren engagerad när det finns möjlighet att själv flytta på de olika

---

<sup>3</sup> Buggar, delar som är felimplementerade vilket gör att delar eller hela system är obrukbara



komponenterna.

4. Slutligen genomförde vi en mindre intervju med användaren för att få veta om vi som utvecklare har uppfattat allt korrekt och om han som användare kände att han, utifrån vårt arbete, hade ett bra underlag för framtida projekt samt utveckling av systemet.

Med dessa fyra delar som underlag genomförde vi en analys vars resultat vi redovisar i denna rapport.

## Bakgrund

### Historia

Området HCI<sup>4</sup> härstammar från det som kallades mjukvarupsykologin<sup>5</sup> i slutet av 1970. Mjukvarupsykologin som vetenskap hade en stadig förankring i experimentell psykologi. Målet med de kontrollerade experimenten var att studera och samla kunskaper om människans interaktion med datorer. Dessa undersökningar var sedan grunden för mer allmänna teorier om människans tänkande och hur människan handlar framför en datorskärm. Teoriernas resultat förklarade inte enbart de observerade delarna utan även i viss utsträckning kunde de förutsäga vad som skulle ske i nya situationer.

Under år 1983 presenterade Card, Moran, Newell en av de mest välkända teorierna inom HCI, *the modell human processor*<sup>6</sup>. De ansåg att området som intresserade oss var hur människor interagerar med datorer. Målet var att den vetenskapliga psykologin skulle hjälpa oss att utforma gränssnitt så att det blev enkelt, effektivt, felfritt och till och med njutbart. I dessa generella teorier såg de användbarhet som en maximalt god anpassning mellan systemets användningsegenskaper och det allmänna vetandet om mänskligt tänkande och handlande. Forskning av detta slag visade sig sedan inte påverka den praktiska systemutvecklingen i den utsträckning man hade hoppats på. De krav som den praktiska systemutvecklingen ställde på kostnadseffektivitet och projektstyrning var början till vad som kallas användbarhetskonstruktion<sup>7</sup>. I detta fall hämtas inspiration från de experimentalpsykologiska delarna, men inriktar sig på tillräckligt bra information istället för statistisk sanning. Good<sup>8</sup> säger att användbarhetskonstruktion går ut på att man på förhand kvantitativt specificerar vilka bruksegenskaper det färdiga systemet ska ha. Därefter byggs produkten och man visar sedan att de planerade egenskaperna faktiskt finns med. Denna metod går inte ut på att bygga det perfekta systemet med obegränsade resurser utan att bygga ett fungerande system som fyller behoven inom de ekonomiska ramarna. Good säger även att utan användbarhetsspecifikationer går det inte att avgöra en produkts användbarhetsbehov eller att mäta om den färdiga produkten uppfyller dess behov. Användbarhetskonstruktion kan inte användas om inte användbarhet kan mätas. Av detta kan vi förstå att det är av stor vikt att inom användbarhetskonstruktion specificera det blivande systemets användbarhet i mätbara termer och under utvecklingsprocessens gång mäta användbarheten för att se om specifikationen är uppnådd. För att kunna mäta användbarheten började man se:

- hur bra testanvändarna presterade vid lösande av olika testuppgifter. Detta mättes i delen lösta uppgifter, hur lång tid det tog eller hur många fel som begicks.
- hur flexibelt systemets utformning var mätt i andel testanvändare ur en heterogen grupp som lyckades genomföra de angivna testuppgifterna.

<sup>4</sup> Human Computer Interaction, sv. Människa-datorinteraktion

<sup>5</sup> Software psychology

<sup>6</sup> Löwgren, J, Stolterman, E, (1998) s 148

<sup>7</sup> Usability engineering

<sup>8</sup> Löwgren, J, Stolterman, E, (1998) s 149



- hur lätt systemet är att lära sig uttryckt i andel lösta uppgifter, prestationstid och antal fel över tiden, hur väl testanvändarna minns det de lärt sig och hur ofta de använde sig av hjälpfunktioner.
- vad testanvändarna tycker om systemet. Detta mättes i subjektiva skattningar av exempelvis systemets hjälpsamhet och effektivitet.

Användbarhetskonstruktion visade sig vara praktisk och vetenskaplig samtidigt som den har kritiserats på grund av mätbarhetskravet. Det finns risk för att utvecklingsprocessen inriktas på sådant som är lätt att mäta som användargränssnittets lättillgänglighet och tidseffektiviteten för testuppgifter som kan sakna relevans för den verkliga användningssituationen. Två av användbarhetskonstruktionens föregångsmän, John Whiteside och Dennis Wixon var också några av dem som först kritiserade detta. År 1987 sa de följande

*För oss handlar mjukvarans användbarhet om i vilken utsträckning mjukvaran stödjer och berikar den pågående upplevelsen hos människor som använder den. Detta fokus på upplevelsen avviker från standarddefinitionerna av användbarhet som primärt inriktar sig på mätbara tecken på användbarhet såsom produktivitet, preferenser, lärbarhet eller prestanda. Även om sådana tecken kan tolkas som viktiga aspekter på användbarhet är de inte primära för oss. Det primära för oss är i stället personens upplevelse i det upplevda ögonblicket.*

*Om [användbarhets-] målen inte utgår från något verkligt meningsfullt för användarna, så kommer den resulterande produkten att vara obrukbar för dem.<sup>9</sup>*

Beyer och Holtzblatt tittade närmare på detta och denna syn på användbarhet och skapade något som kallas kontextuell design. Vid denna teknik genomförs kravgenerering, design, implementation och utvärdering så många gånger som projektresurserna tillåter. Genom fältstudier baserade på etnografiska tekniker försöker man säkerställa att utvecklarnas förståelse av den blivande användningssituationen stämmer överens med användarens och kundens uppfattningar. Som hjälp för att säkerställa denna förståelse används diagram, modelleringstekniker och enkla prototyper som beskriver det nya systemet. Efter hand som idéerna och förståelsen växer fram utvärderas dessa om och om igen för att nå datorimplementerade prototyper och slutligen driftklara system. Under 90-talet har tekniker främst setts som ett sätt att öka kund- och användaracceptansen samt att utöka utvecklarnas förståelse för användningssituationen.<sup>10</sup>

## Systemutveckling i Skandinavien

I Skandinavien präglades den tidiga historien inom systemutvecklingsområdet mycket av Herberts Simons (1969) tankar om en designvetenskap. Han ansåg att många yrken kretsar runt design av konst och detta ställer andra krav än det som han ansåg vara de traditionella grunddisciplinerna, som han sa var bland annat matematik och fysik, för

<sup>9</sup> Whiteside, J, Wixon, D, (1987) s 17 f

<sup>10</sup> Löwgren, J, Stolterman, E, (1998) s 148 f

till exempel ingenjörer. Han var kritisk till den befintliga designkunskapen som han ansåg vara informell och ovetenskaplig. Hans lösning på detta problem var att designa systemen så att det är möjligt att behandla dem logiskt och matematiskt. Simon menade att i systemutveckling ska komplexa system delas upp i sina beståndsdelar och definiera funktionerna separat för varje del. I slutet av 60-talet sa man inom området för programvaruteknik att datorprogram sågs som matematiska delar som kunde förklaras som tänkta specifikationer. Utifrån dessa tänkta specifikationer kunde man sedan bygga ett system där korrektheten för systemet kunde garanteras. Inom de första utvecklingsmetoderna i systemutveckling var det en central tanke att de blivande användarna skulle kunna ge fullständiga och tydliga beskrivningar av sina behov och önskemål.

Under 70-talet växte filosofin om participativ design<sup>11</sup> fram som fokuserade sig på de mänskliga och sociala sammanhang där informationssystemen utvecklades och användes. En önskan var att använda systemutvecklingsprocessen som ett sätt att öka de anställdas inflytande och medbestämmande. Participativ design användes även under 80-talet och fortfarande under 90-talet. Idag ser man participativ design som en process av ömsesidigt lärande där designer och användare lär av varandra. Desto mer bakgrund dessa personer har att delge varandra ju mer deltagande blir designprocessen. Participativ design kräver alltså inte enbart att användaren deltar i designarbetet utan även att designern deltar i användnings- och arbetsituationen. Olika designarbetsformer skapar olika möjligheter för användaren att uttrycka sin praktiska kompetens i designprocessen. Mock-upper ger deltagarna en bild och utifrån dessa kan de se hur det system som ska utvecklas kommer att se ut. Detta gör det möjligt att utforska konsekvenserna av konstruktionen i interaktionen mellan användare och system. Det är viktigt att alla deltagarna upplever sin närvaro som meningsfull och engagerande för att det participativa designprocessens arbetsformer ska vara bra.<sup>12</sup>

## HCI

Termen HCI, human-computer interaction togs i bruk i mitten av 1980-talet, den beskrev det nya tankesättet att se på interaktionen mellan datorer och människor. Utvecklingen inom HCI har varit stor under de senaste tjugo åren dock väldigt långsam. I boken Human-computer interaction av Jenny Preece beskrivs termens uppkomst på följande sätt:

*This term acknowledged that the focus of interest was broader than just the design of the interface and was concerned with all those aspects that relate to the interaction between the users and computers.*<sup>13</sup>

---

<sup>11</sup> Participativ design, en designprocess där det är viktigt att ha med *deltagarna*, dessa måste *delta* i utvecklingen

<sup>12</sup> Löwgren, J, Stolterman, E, (1998) s 152 f

<sup>13</sup> Preece, J, (1994) s 7

Målet med god HCI är att producera användbara, säkra och funktionella system. Vare sig du är erfaren användare eller inte ska det inte vara omöjligt att tillämpa systemet. Ett nyckelord som används är ”användbarhet” och innebär att man hämtar förslag och idéer från ”användarens sida”. Som systemutvecklare måste man ha känsla för vad som är betydelsefullt och tänka på följande saker när man utvecklar ett system:

- att analysera från annat perspektiv än sitt eget
- att ta hänsyn till omkringliggande faktorer
- för att kunna lösa problemet måste man ha förståelse för omgivningen
- att alla människor är olika i tänkandet, fysiskt och psykiskt. Det har betydelse för designen när man ska anpassa systemet efter individen/slutanvändaren

För att ett system ska uppfylla kraven för användbarhet ska den utföra uppgiften som den är ämnad för på ett effektivt sätt (effectiveness), systemet ska hjälpa användaren att utföra sina uppgifter på ett enkelt och snabbt sätt (efficiency), det ska vara säkert att använda utan att något kan förstöras (safety), det ska uppfylla de krav som användaren har på systemet (utility), det ska vara lätt att lära sig hur det används (learnability) och man ska komma ihåg hur applikationen fungerar även om man inte använder den på ett tag (memorability).<sup>14</sup> Som användare vill man inte lägga ner för mycket tid på att fundera ut hur ett system fungerar. Därför bör systemutvecklarna ta reda på hur lång tid användaren är beredd att lägga ner för att lära sig ett program. 10-minuters-regeln<sup>15</sup> är en bra utgångspunkt. Inom dessa tio minuter ska användaren kunna hitta det man letar efter och hinna få en övergripande översikt av systemet, vilket ska vara lätt om det finns en logisk uppbyggnad.

Det är viktigt att vara användarcentrerad när man utvecklar ett system. Därför ska man låta användaren vara involverad under hela utvecklingen. Redan i ett tidigt stadie kan användaren förstå vad produkten är kapabel till och inte. De får även en bättre förståelse för hur den slutgiltiga produkten kommer att påverka deras arbete och vad som kan förväntas av den. Risker för besvikelser vid levererad slutprodukt minskar på så vis. Dåligt designade system kan vara väldigt irriterande för en användare vilket kan resultera i att man inte alls använder sig av systemet.<sup>16</sup>

## Utvecklingsmodeller

Skapandet av ett system är en process som innehåller många olika aktiviteter. För att se hur de olika aktiviteterna är relaterade till varandra, använder man sig av modeller. Alla utvecklingsmodeller har sina fördelar och nackdelar men för att kunna bestämma vilken som är mest effektiv måste man se till vilket sammanhang den ska användas i.

## Vattenfallsmodellen

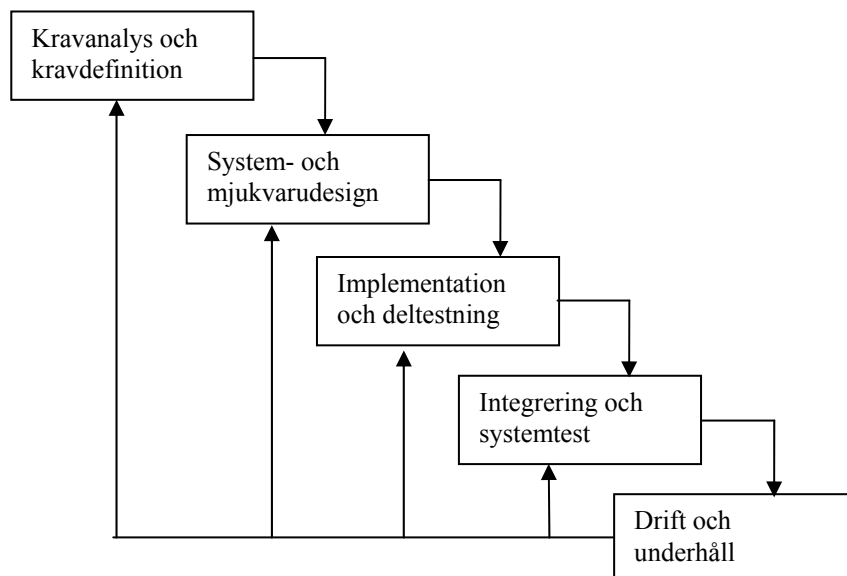
Vattenfallsmodellen kallas ofta för traditionell livscykelmodell. Den ger en bild av systemets hela livscykel, se Figur 1. I modellen kan inte en ny fas påbörjas förrän den

<sup>14</sup> Preece, J, (1994) s 14 f

<sup>15</sup> Rubinstein, R, Hersh, H, (1984) s 9

<sup>16</sup> Preece, J, Rogers, Y and Sharp, H, (2002) s 280 f

gamla är helt färdig. När en fas i vattenfallsmodellen är klar kan man inte gå tillbaka och ändra i efterhand.<sup>17</sup> Problem som uppstår skjuts längre fram i tiden, ignoreras eller kodas runt. Det finns dock en viss återkoppling mellan faserna, men det kan bli kostsamt och problematiskt att åtgärda.



Figur 1 Vattenfallsmodellen (Sommerville 2001)

Utseendet och namnen på varje fas varierar men vattenfallsmodellen börjar med ”kravanalys och kravdefinition” som innebär att man specificerar vad systemet ska kunna göra och inte göra utifrån användarens mål och önskemål, samt vilka personer som kommer att vara berörda av systemet och i vilken miljö som den ska användas i. När den fasen är färdig går man vidare till nästa fas: ”system och mjukvarudesign”. Där görs en struktur på hela systemet och det utförs en utförlig beskrivning av systemet och dess funktioner. När denna fas är färdig görs ”implementation och deltestning” som innebär att man testar de delar av systemet som är färdiga och försäkras om att de uppfyller kraven som finns i kravspecifikationen. Med ”Integrering och systemtest” menas att systemdelarna sätts in i sin helhet och testas för att vara säkra på att systemet uppfyller sina krav i sin helhet. Efter att testningen är färdig levereras systemet till kunden. Den sista fasen är ”drift och underhåll” som är den aktivitet som tar längst tid och kräver mest resurser av alla faser. Systemet installeras och sätts i användning. Underhållsbiten innebär att ”fel” som inte upptäckts i tidigare faser rättas till. Detta kan vara tidsödande och kostsamt.<sup>18</sup>

Vattenfallsmodellen ska endast användas när kraven från användaren är helt tydliga och helt förstådda av utvecklarna. Allt eftersom utvecklingen har gått framåt och kunden blivit en del av systemutvecklingsprocessen har vattenfallsmodellen visat sig vara otillräcklig. Kundens krav och önskemål kan lätt misstolkas och/eller vara vagt definierade, vilket kan göra att de behöver ändras mitt i processen. Om man inte har möjlighet att gå tillbaka en fas och ändra i kravspecifikationen kan det innebära att kunden i slutändan får ett system som man inte kan använda. Utvecklingen försenas ofta

<sup>17</sup> Preece, J, Rogers, Y and Sharp, H, (2002) s 187 f

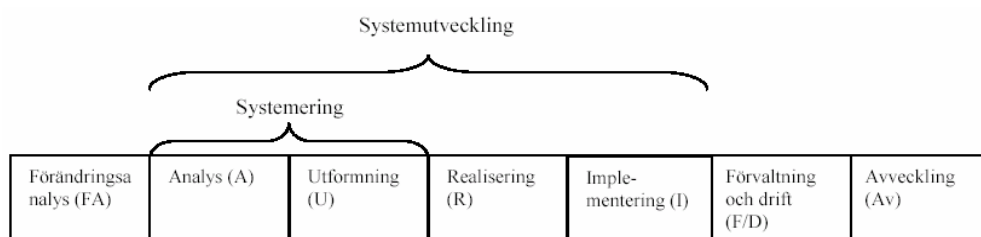
<sup>18</sup> Sommerville, I, (2002) s 44 ff

på grund av att man måste vänta medan de andra utvecklingarna färdigställer sina deluppgifter. En liten del av iteration är nu inbyggd även i vattenfallsmodellen men för delen att kunna utvärdera systemet med användaren är ännu inte inbyggd i denna modell.<sup>19</sup>

## Livscykelmodellen

För att kompensera vattenfallsmodellens brister har man utvecklat olika typer av iterativa modeller. Största skillnaderna ligger i att man försöker få med kunden så mycket som möjligt i utvecklingen. Livscykelmodellen uppfyller detta kravet med att vara användarecentrerad. Den är iterativ vilket menas att man kan gå tillbaka mellan olika faser och kontrollera att man jobbar mot och möter användarens krav och önskemål. De aktiviteter som finns är i stort sett samma som i vattenfallsmodellen, men här finns en medvetenhet om att kraven som kunden har på systemet, kan behöva ändras av olika anledningar under utvecklingen.<sup>20</sup> I livscykelmodellen kan man gå tillbaka mellan faserna och ändra om något skulle visa sig vara ”fel”. Ibland kan en fas överlappa en annan och ibland kan man behöva gå tillbaka till en fas och komplettera den. Livscykelmodellen visar hela systemet från att det föds till dess att det avvecklas. I livscykelmodellen, se Figur 2 ingår faserna:

- Förändringsanalys
- Analys
- Utformning
- Realisering
- Implementering
- Förvaltning och drift
- Avveckling



Figur 2 Livscykelmodellen (1994) Andersen

Erling S Andersens livscykelmodell<sup>21</sup> börjar med ”förändringsanalysen”. Denna del anser Andersen inte ens höra till den traditionella systemutvecklingen. Det första som görs är att beskriva nuläget och sätta sig in i den nuvarande verksamheten. Här analyseras de problem och möjligheter som finns och man beskriver den önskade

<sup>19</sup> Preece, J, Rogers, Y and Sharp, H, (2001) s 185 ff

<sup>20</sup> Sommerville, I, (2001) s 46

<sup>21</sup> Andersen, Erling S, (1994) s 41

situationen. Det är viktigt att användaren är med redan i denna fas av utvecklingen. Tillsammans tittar man på hur det nuvarande läget och den önskade situation ser ut samt förändringsbehovet. Den andra fasen kallas ”analys” och är uppdelad i två delar. I verksamhetsanalysen analyserar man användarens arbetsuppgifter och ser på vilket sätt ett informationssystem kommer att underlätta i verksamheten. I andra delen av analysen som kallas informationssystemanalys ska man bedöma vad som ska finnas med i systemet. Resultatet av denna fas är en kravspecifikation. Den tredje fasen, ”utformning” delas också upp i två delar. Den första kallas ”Principiell utformning av teknisk lösning” och här väljs den tekniska lösningen som önskas. Det är viktigt att se den lösningen som är mest ändamålsenlig för verksamheten. Även om man är van och känner sig säker på sitt gamla system så får man inte vara rädd att testa något nytt. I andra delen, ”Utformning av utrustningsanpassad teknisk lösning”, väljer man vilken utrustning man ska använda sig av till exempel vilket operativsystem, hur data ska lagras och vilket programspråk som ska användas. Fjärde fasen är ”Realiseringen” där själva informationssystemet utarbetas och programmeras. ”Implementeringen” är sista fasen som ingår i systemutvecklingen. Systemet implementeras i verksamheten. I fasen ”Förvaltning och drift” ska systemet underhållas och förbättringar ska göras. Sista fasen kallas ”Avveckling”. Här är det viktigt att säkra den information som finns lagrad i systemet, när systemet ska läggas ner. För att kunna jobba på detta sätt krävs en aktiv kund som är villig att lägga ner tid och resurser på att testa och godkänna eller förkasta förslagen som utvecklarna kommer fram till.

Andersen beskriver utvecklingsarbetet så här: ”Användarnas uppgift är att fastställa vad man önskar uppnå. Experternas uppgift är att finna de tekniska lösningar som ger användarna vad de önskar.”<sup>22</sup>

## Mock-up

Att omsätta en vision i en operativ bild innebär att åskådliggöra, precisera och detaljera. För att gestalta denna bild krävs det någon teknik för att själv kunna tolka den och att kunna kommunicera den till andra. En operativ bild innehåller tankar om dellösningar och delfunktioner. Genom att bygga upp denna bild kan konflikter och motsägelser som inte syns i en mer övergripande vision förebyggas. Genom olika sätt kan en operativ bild gestaltas, bland annat genom en så kallad mock-up. En mock-up är en dynamisk pappersprototyp som syftar till att illustrera det blivande systemets interaktivitet. Denna teknik är speciellt användbar då kraven från användaren är svåra att specificera direkt. Med en testmodell visar man systemet ur olika perspektiv för att åskådliggöra den tänkta byggnaden.

Mock-up ses som en förenklad version av systemet. Idén bygger på att man med hjälp av bilder förbereder ett antal dialogtillstånd som användaren kan hamna i. Vid demonstration eller testning av prototypen agerar man själv ”fönsterhanterare”. Genom att låtas klicka på olika knappar eller skriva något i ett fält visas olika sidor beroende på vilken inmatning som gavs. Fördelarna med en mock-up är att den kan ge en bra känsla för systemets interaktiva beteende men även att de är enkla och billiga

---

<sup>22</sup> Andersen, Erling S, (1994) s 39 ff



att bygga samt lätta att använda. En nackdel kan vara att det är svårt att gestalta vissa saker till exempel då man har för avsikt att använda sig av direkt manipulation.<sup>23</sup> Direkt manipulation är när man märker det användaren utför direkt som output, exempelvis ändrar volymen.<sup>24</sup>

Mock-uper är ett stort stöd i utvecklingen och i kommunikationen. Man låter kunden och utvecklaren vara jämspelta. Denna typ av tillvägagångssätt kräver en engagerad kund som är beredd att lägga ner tid för att sätta sig in i, testa och godkänna mock-upen. Det är viktigt att hitta ett gemensamt språk mellan slutanvändare och utvecklare, annars är det lätt att missuppfatta varandra. Trots att teamet och slutanvändaren använder samma termer kan de ha olika betydelse för de båda. Detta sätt att arbeta på, underlättar då en kravspecifikation ska skrivas. Genom arbetet kan användare och utvecklare se att man är på samma nivå och förstår varandra.

Alla människor har olika bakgrund som ger olika erfarenheter och perspektiv. Vid utveckling i team är det svårt att tillgodose allas viljor men tankarna kan även ses som en fördel, då flera infallsvinklar på problem kan belysas. När utveckling sker i team är det viktigt att projektledaren plockar ut bra och dåliga idéer för slutanvändarens behov. Den grafiska designen på en programvara utvecklad för vana datoranvändare kan enkelt förstås för dessa, men ge komplikationer för en användare som aldrig förut arbetat med en dator.

## Fastighetssystemets kravspecifikation

Ur projektgruppens dokumentation har vi sammanställt kravspecifikationen som finns för fastighetssystemet. Vi har valt att dela upp dem i funktionella och icke-funktionella krav. Nedan beskrivs utvalda krav som har varit relevanta för oss och vårt resultat. En sammanfattning av alla krav finns i bilaga 1.

Huvudsyftet med projektet är att erbjuda en intern webbplats<sup>25</sup> med nödvändig information om alla fastigheter och installationer med såväl bilder som text. Den ska vara användbar och lätt att använda för personer som saknar datavana. Strukturen för systemet ska göra att det är lätt och logiskt uppbyggt. Behörig personal ska enkelt kunna lägga till och ta bort information från webbplatsen, medan all personal ska kunna läsa informationen.

### ***Funktionella krav***

Användare

- Vissa personer ska kunna logga in och ändra uppgifter.
- Annan personal ska inte kunna logga in utan bara ta del av information.

Användaren ska kunna:

---

<sup>23</sup> Löwgren, J, Stolterman, E, (1998) s 123

<sup>24</sup> Preece, J, (1994) s 270 f

<sup>25</sup> Försvenskning av det engelska ordet website, i detta sammanhang syftar vi till lite större hemsidor på Internet som innehåller flera sammanlänkande sidor.

- lägga till en bild när nya fastigheter köpts eller när nya installationer skett
- lägga till och skapa nya checklistor för olika byggnader
- skriva checklista
- skriva och lägga till text till bilder, byggnader och orter
- skriva och lägga till text för en viss sorts installation
- skriva och lägga till text för ett system

Användaren ska kunna:

- ta bort bilder
- ta bort checklista
- ta bort text för bilder, byggnader och orter
- ta bort text för en viss sorts installation
- ta bort text för ett system

Användaren ska kunna:

- ändra text och checklistor
- ändra text för bilder, byggnader och orter
- ändra text för en viss sorts installation
- ändra text för ett specifikt system

Uppdatering:

- Sista uppdateringsdatum ska alltid synas och vara placerat uppe i höger hörn
- Checklistor ska kunna skrivas ut

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

Sidans utseende

- Länkar ska vara blå
- Länkar ska vara understrukna när pekdonet förs över dem
- Menyn ska alltid vara synlig till vänster på sidan
- I menyn ska de fem orterna vara listade enligt följande: Ronneby, Kallinge, Bredåkra, Saxemara, Möljeryd. Därefter allmän information, checklistor, logga in och felanmälning.
- ”Tillbaka”-knappar ska finnas i applikationen där det behövs
- Svenska kyrkans logotyp ska alltid finnas i övre vänstra hörnet. Under logotypen ska det stå ”Svenska kyrkan i Ronneby”

Tydligt användargränssnitt

- Lätt att navigera
- Ingen onödig text eller bilder på sidorna
- Innehållet på sidan ska visas strukturerat och logiskt

Det var bland annat dessa krav som låg till grund för vårt arbete och vår studie.



## Resultatbeskrivning

Momenten bestod av mer informella samtal och intervjuer med användaren av fastighetssystemet samt uppbyggnad av mock-uper. I avsnittet har vi sammanställt det material vi fått fram av dessa moment och utifrån den dokumentation som projektgruppen som utvecklat systemet har gjort.

### **Första mötet med användaren av fastighetssystemet**

Vid första samtalet med kunden fick vi titta på fastighetssystemet som projektgruppen gjort. Kunden är den person som har beställt fastighetssystemet, han är även en av slutanvändarna. I fortsättningen av arbetet kommer vi att benämna kunden som "han" eller "användaren". Han berättade för oss vad systemet var till för och vilka funktioner som önskades av fastighetssystemet. Allt som fanns med i avtalen fanns i systemet, men vissa sidor var ologiskt uppbyggda och inte så väl genomtänkta. Vidare blev vi upplysta om att det inte fanns någon klar bild över hur systemet önskades vara uppbyggt när det beställdes, utan enbart vilka funktioner som önskades finnas i det. Systemet innehöll en mängd buggar som gjorde att det inte kunde användas för det ändamål det var byggt för.

Menyn, se Figur 3 för en inloggad användare i systemet ansågs vara för lång och rörig. Det krävs för många musklick för att ändra egenskaper för till exempel en byggnad. För att ändra egenskaper som text och bild för en byggnad krävs att användaren väljer *Lägg till/Ändra/Ta bort text* i steg 1, i steg 2 väljer användaren vilken byggnad texten ska ändras för och sedan spara. Efter det väljer användaren i steg 3 *Lägg till bild*, i steg 4 väljs för vilken byggnad bilden ska läggas till för, i steg 5 väljs bild och slutligen trycker användaren på "Spara". Hans önskemål visade sig vara, att det skulle vara möjligt att ändra alla dessa uppgifter för en specifik byggnad på en och samma sida, istället för att behöva klicka sin in i olika menyer. På samma sätt önskas att det ska vara möjligt att ändra alla uppgifter för en installation för en byggnad på en specifik sida.

Angående länkarna som behandlar checklistor ansåg han också att det fanns vissa brister. När länken för *Lägg till i checklista* väljs och användaren navigerat sig fram till en viss checklista, visas inte de rader som användaren tidigare lagt in. För att få veta vilka rader som redan tidigare finns i systemet är han, för att undvika dubbelinskrivning, tvungen att först skriva ut en checklista för aktuellt system. Det ansågs även var mycket klick för att skriva ut olika checklistor och han önskade att det ur en lista gick att välja vad som skall skrivas ut.

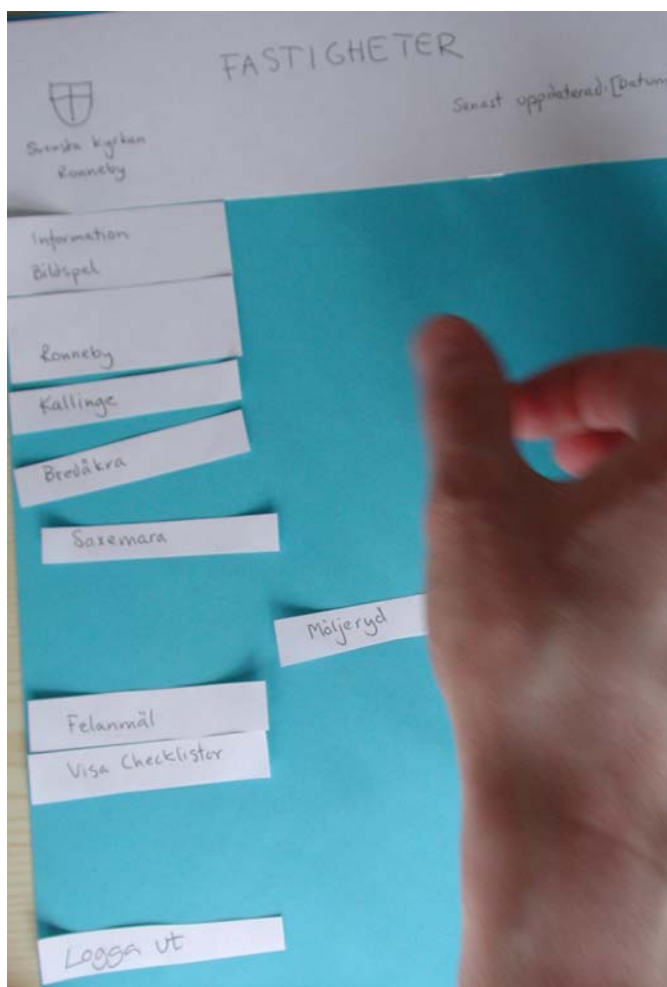


Figur 3 Meny som visas för inloggad användare

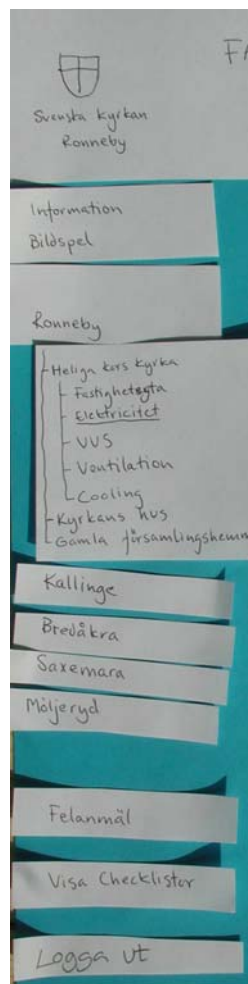
## Strukturen ändras inför andra mötet

Efter första mötet fick vi i uppdrag av användaren att skissa på ett annat förslag utifrån det vi sett och hört vid första mötet. För att kunna göra detta gick vi igenom projektgruppens dokumentation för att se vilka krav som han hade ställt på fastighets-systemet.

Efter genomgång av kravspecifikationen valde vi att ändra strukturen. Vi arbetade enligt designprincipen *consistency*<sup>26</sup> som bygger på att upprätthålla en enhetlighet för varje sida. Funktionerna kategoriserade vi och placerade på sidor med egenskaper för byggnader/installationer/system. För att nå dessa sidor väljer användaren först vilken ort som något ska ändras för. Menyn fick då ett helt annat utseende enligt följande ordning, *Information, Bildspel, Ronneby, Kallinge, Bredåkra, Saxemara, Møljerud, Felanmälan, Visa checklistor* och slutligen en länk för att *Logga ut*, se Figur 4. Enligt



Figur 4 Mock-up efter första mötet med kund för systemet



Figur 5 Mock-up på meny med trädstruktur efter första mötet med kund för fastighetssystemet

<sup>26</sup> Preece, J, (2002) s 24

detta förslag ska sedan alla funktioner som hör samman finnas under respektive sida. På installationsidan ska alla delar för installationen vara möjliga att redigera.

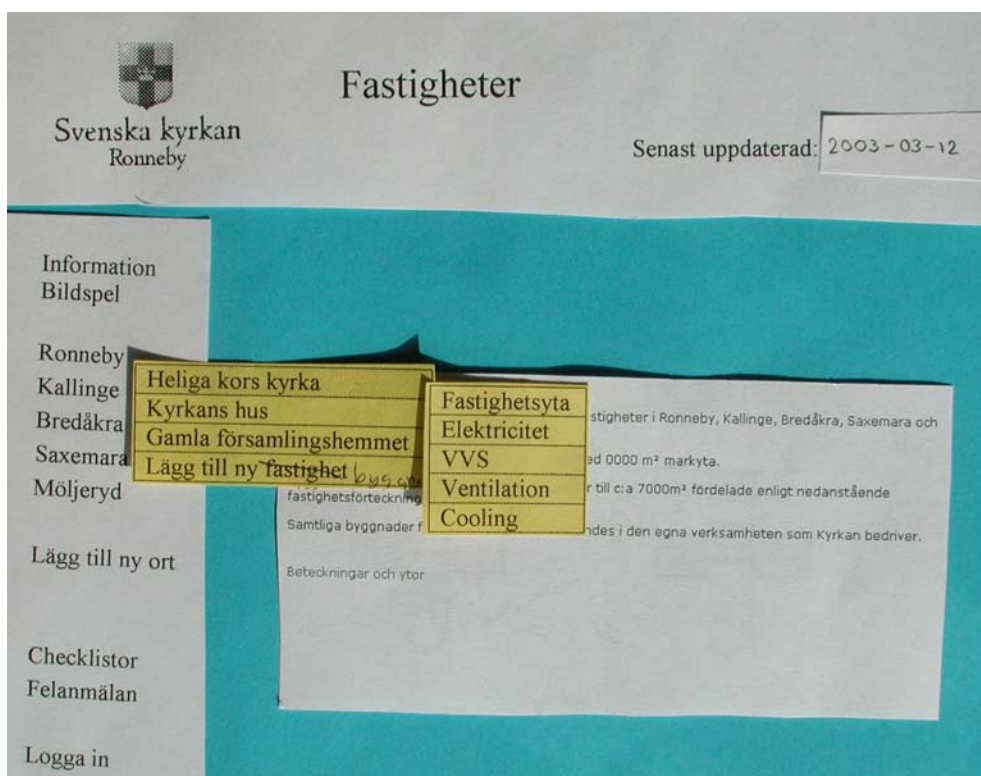
För att öka överskådligheten över var användaren befinner sig i applikationen, valde vi också att det ska fällas ner en trädstruktur under den ort användaren valt, se Figur 5. Alla undersidor till en viss ort ska finnas med i denna trädstruktur. Länken för sidan som användaren är inne på ska dessutom utmärka sig på något vis.

För att öka överskådligheten och minska klicken för användaren valde vi att vidareutveckla denna innan vi på nytt träffade användaren. I nästa mock-up skapade vi drop-downmenyer som visas när användaren drar pekmarkören över en viss ort. Det som visas är allt som finns under ortens första steget i trädstrukturen, det vill säga byggnaderna, se Figur 6. När sedan användaren flyttar pekmarkören över menyn för byggnaderna fälls ytterligare en meny ut för de speciella installationer som finns för den byggnad som markeras. Då användaren klickar på byggnadens namn visas en sida med information om byggnaden. Om en installation väljs visas information för denna.

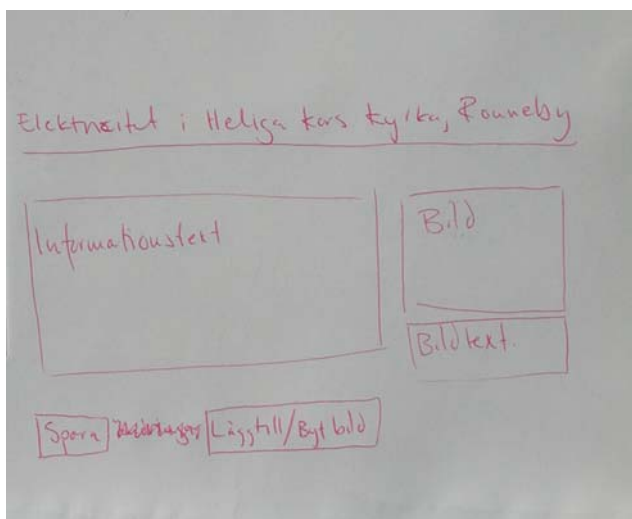
På varje informationssida för byggnader ska det finnas en rubriktext för vilken byggnad som är vald samt vilken ort byggnaden finns i. Under rubriken ska användaren kunna läsa texten för denna byggnad samt kunna editera den om så önskas. Editerbar text ska vara *Namn*, *Byggnadsyta*, *Fastighetsyta*, *Fastighetsbeteckning* samt informationstext benämnd *Text* om byggnaden. Vid sidan om informationstexten ska det även finnas möjlighet för användaren att lägga till samt byta bild för byggnaden. Om användaren klickar på en installation kommer en informationssida om denna installation att visas. I vårt först mock-upförslag visas det en rubrik, som talar om vilken installation som valts, samt vilken byggnad installationen finns i och slutligen på vilken ort. Under denna rubrik finns det en editerbar textruta där all information om installationen visas. På sidan med informationstexten ska det även finnas möjlighet för användaren att lägga till/byta en bild för installationen samt under bilden skriva och editera en bildtext, se Figur 7.

Först när användaren klickat på någon länk i dropdownmenyerna kommer trädstrukturen under den valda orten att visas i menyn till vänster. Alla undersidorna för vald ort i trädstrukturen visas när användaren valt orten.

Detta förslag visade vi för användaren vid vårt andra möte.



Figur 6 Mock-upförslag för ökad överskådlighet och minskning av antal klick för användaren genom dropdownmenyer



Figur 7 Mock-upförslag för hur informationssidan för olika installationer ska se ut, i detta fall elektricitet i Heliga kors kyrkan i Ronneby

## Andra mötet med användaren

Vid detta möte visade vi mock-uper enligt figur 6 och 7. Vi började med att förklara varför vi gjort modeller i papper och vilken hjälp användare och utvecklare kan ha av mock-uper. I stort tyckte användaren att systemet kändes mer strukturerat och överskådligt än tidigare. Han framförde önskemål om att alla informationssidor ska vara så lika varandra som möjligt. Oberoende av vilken installation som visas ska de vara upp-

byggda på samma sätt. På varje sida ska det finnas en knapp för att ta bort den byggnad eller installation som visas. Under vårt möte frågade han hur vi menade att det skulle vara möjligt att lägga till nya installationer för olika byggnader. Han ansåg att det enklaste skulle vara, om det under alla installationer i dropdownmenyerna fanns ett val för detta, på liknande sätt som det finns val för att lägga till byggnad på en ort.

Länken bör heta *Lägg till ny installation*. För varje installation kan det finnas behov av att kunna lägga till flera bilder. Varje bild ska kunna ha en tillhörande bildtext. För att få en tydlig sida för bilderna tyckte han, att det skulle vara bättre om det för varje installation fanns en speciell bildsida, där alla bilder och bildtexter för installationen visas.

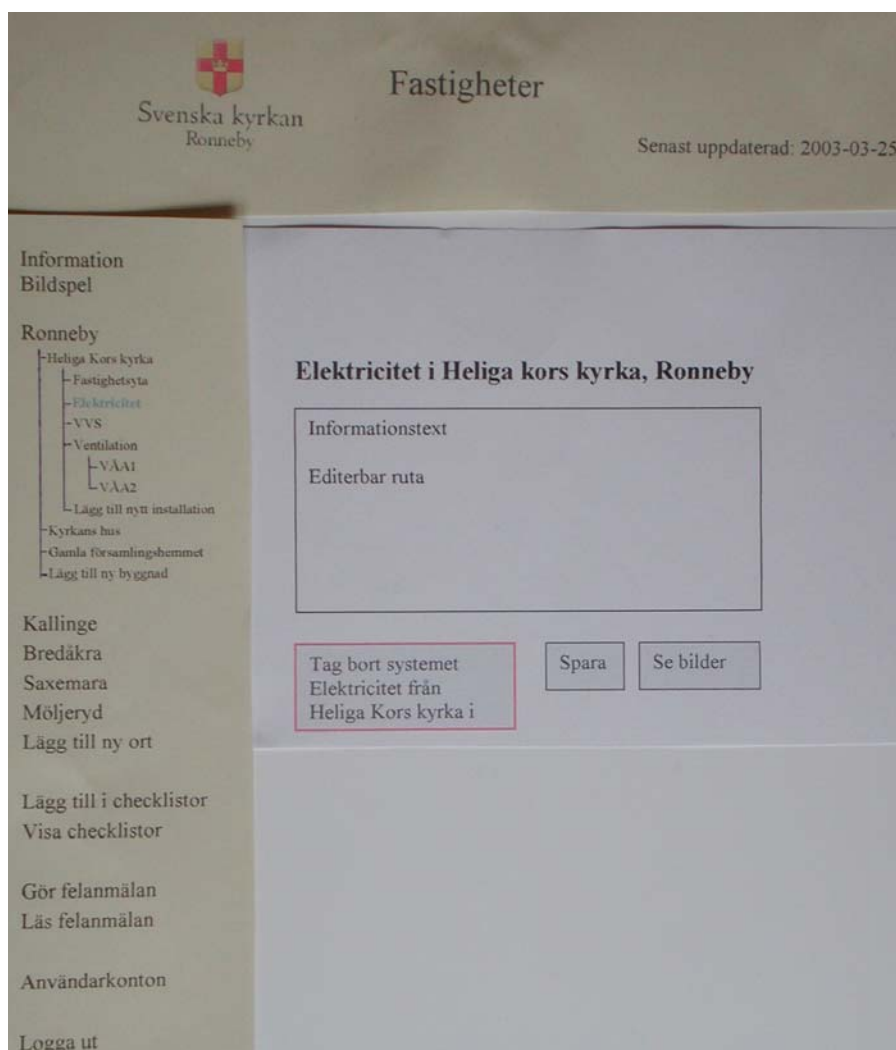
Under vårt möte med användaren begrundade och bytte vi mock-updelar, för att se olika sidor och hur han reflekterade över det vi gjort. Han kompletterade med sina förslag och önskemål. Han menade att det nu var lättare att förstå hur vi tänkte och hur det skulle se ut i slutet och såg också fördelen med att kunna ändra på saker innan implementationen börjar. Med hjälp av dessa mock-uper kunde han se strukturer samt möjligheter och begränsningar i det vi visat honom. Strukturen för informationssidorna önskas vara den struktur som vi byggt upp för installationen. Han önskar även att det ska finnas länkar för att komma till sidor för *ritningsförteckningar* och *bygg-händelser*.

Vid detta möte beskrev användaren mer i detalj vilken funktion checklistorna har. På dessa sidor ska det vara möjligt för användaren att lägga till delar i olika byggnaders system som behöver någon form av översyn. Den person som sedan beställer servicen, eller utför kontrollen, ska kunna skriva ut en checklista för att veta vilka delar som behöver översyn samt vilken service de ska genomgå.

### ***Mer struktur på undersidorna inför tredje mötet***

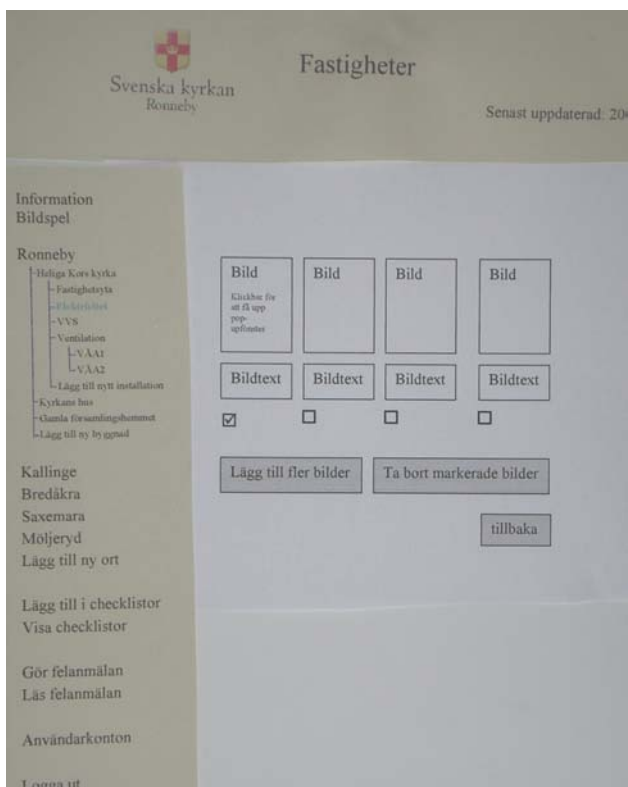
Efter andra mötet arbetade vi vidare med de samlade intryck som vi fått av användaren. Vi gjorde om strukturen för installationssidan. På sidan placerade vi en knapp, som skiljer sig i utseende, för att ta bort hela installationen från byggnaden. På knappen ska det stå exempelvis "Ta bort installationen Elektricitet från Heliga Kors kyrka i Ronneby". Knappen ska vara omsluten av en röd markering. Alla bilder för installationer valde vi att placera under en knapp där det finns möjlighet att visa flera bilder och bildtexter för installationen, se Figur 8.



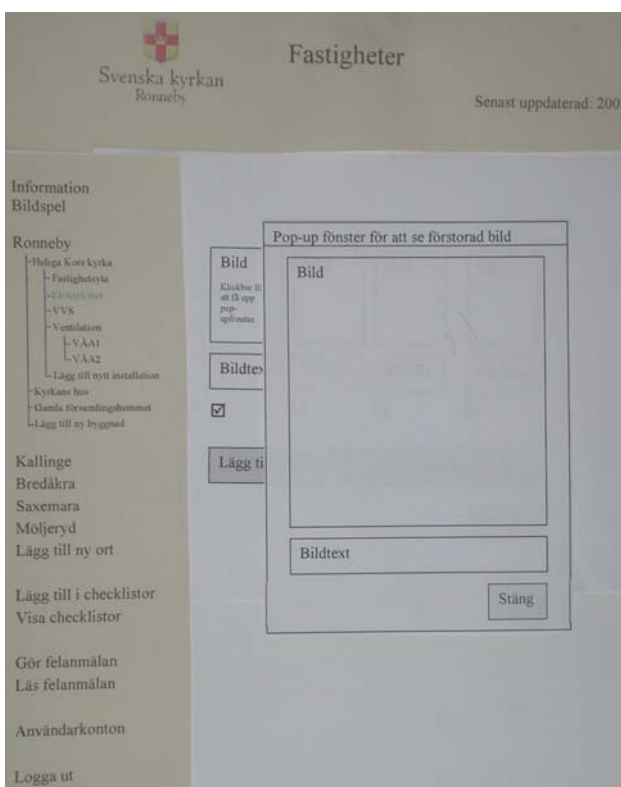


Figur 8 Mock-up för systemet som visar en sida för installation efter andra mötet med användaren

På bildsidan visas alla bilder för det valda systemet med editerbara bildtexter. På denna sida ska det vara möjligt att lägga till och ta bort bilder. Bilder som ska tas bort ska markeras med hjälp av en checkboxruta, se Figur 9. För att användaren ska kunna komma tillbaka till föregående sida väljs "Tillbaka"-knapp och då sparas eventuell text som ändrats. Om en bild i större format önskas ses klickar användaren på den önskade bilden. Då visas ett popup-fönster med vald bild och bildtext, se Figur 10. I menyns trädstruktur visas inte bildsidor utan endast ut till installationer samt eventuella undersystem för installationer om en installation består av flera olika undersystem.

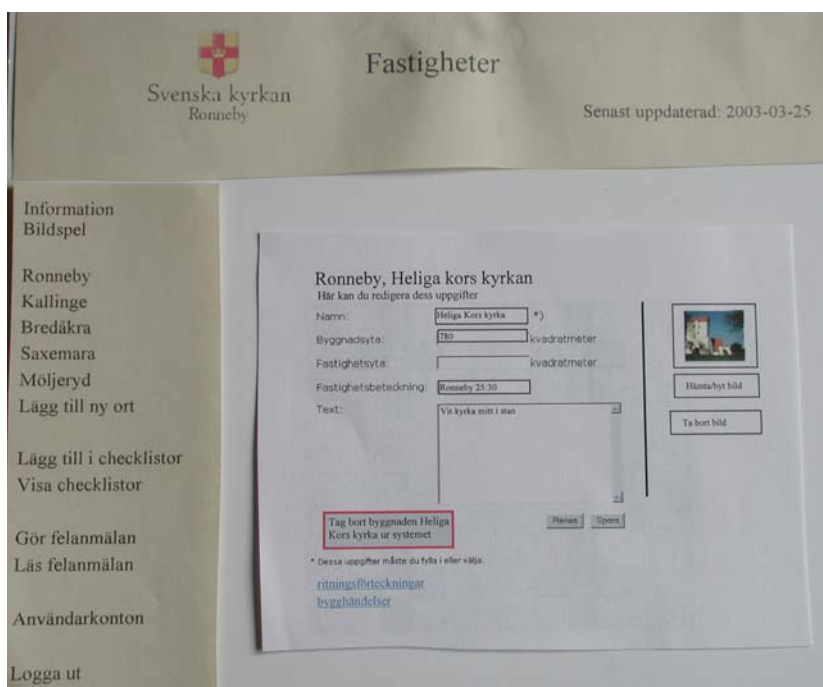


Figur 9 Sida som presenterar alla bilder med bildtexter för vald installation



Figur 10 Pop-upfönster med förstord bild med bildtext

På informationssidan för en byggnad ska rubriken vara vilken ort byggnaden finns i samt namn på byggnaden. På denna sida ska användaren ha möjlighet att ändra och lägga till information om *namn*, *byggnadsyta*, *fastighetsyta*, *fastigbeteckning* samt



Figur 11 Sida med egenskaper för Heliga kors kyrka

*lägga till, ändra och ta bort* bild för byggnaden. Sidan ska även innehålla länkar till sidor där information om *ritningsförteckningar* och *bygghändelser* presenteras. På sidan finns även en knapp för att ta bort hela byggnaden från orten. En knapp som skiljer sig i utseende. På knappen ska det stå exempelvis ”Ta bort Heliga Kors kyrka ur systemet”. Användaren ska även ha möjlighet att rensa alla fält genom att trycka på en knapp med texten ”Rensa”, namn på byggnaden måste vara ifyllt för att det ska vara möjligt att spara, se Figur 11.

Figur 12 Sida för att lägga till ny byggnad

När användaren exempelvis väljer att lägga till en ny byggnad i Ronneby ska ett formulär visas som innehåller rubriken ”Lägg till ny byggnad i Ronneby”. Alla fält om byggnaden, som redan finns i systemet, ska vara synliga. Länkarna till ritningsförteckningar och bygghändelser ska finnas med men inte vara klickbara, se Figur 12. När byggnaden är sparad ska byggnadens namn visas i dropdownmenyn för vald ort samt i trädstrukturen när den visas.

Vid utformandet av sidan för checklistor valde vi att behålla mycket av strukturen från det redan befintliga systemet. När användaren väljer sidan för *Lägg till i checklistor*, se Figur 13, måste användaren välja för vilken byggnad det ska läggas till rader i checklistan. Detta val görs i en dropdownmeny. När byggnaden är vald visas ytterligare en dropdownmeny, där användaren väljer vilken installation som är aktuell. Om installationen består av undersystem finns ytterligare en valmöjlighet där systemet väljs. Efter det att alla val är gjorda visas en tabell med rubrikerna *nr*, *delutrustning*, *beteckning* och *typ av kontroll*. Under dessa rubriker visas de rader som finns lagrade för det valda systemet. Efter det att resultatet av sökningen i databasen visats, följer formulärrutor där användaren har möjlighet att lägga till fler poster. Under formulärrutorna har användaren möjlighet att trycka på en ”Rensa”-knapp för att radera all text i formulärrutorna samt ”Spara”-knapp för att lagra de nya ändringarna i databasen.





Figur 13 Sida för att lägga till rad/rader i checklistor

Inför mötet med användaren skapade vi två olika alternativ för att presentera och skriva ut checklistorna. För att titta på och skriva ut checklistor väljer användaren *Visa checklistor* i menyn till vänster.

I alternativ 1 fällt en trädstruktur ner som visar alla checklistor som finns för installationer i systemet, därefter klickar användaren på den checklista som han önskar visa. Checklistan skrivs ut genom att trycka på *Skriv ut*. I alternativ 2 fällt det inte ner någon trädstruktur. Alla checklistor som finns presenteras på sidan till höger om menyn, där de visas i en kolumn. Listan i kolumnen är uppdelade efter orter. För varje ort ordnas listorna efter byggnader och installationer, exempelvis *Heliga Kors kyrka*, *ventilation*. Önskar användaren titta på en checklista klickar man på länken för önskad lista och den visas. Från detta nya fönster kan användaren sedan skriva ut aktuell checklista.

Efter att han funderat och plockat lite själv med de olika delarna tyckte han att alternativ 2 var det bästa. Han önskade dock att det även skulle vara möjligt att skriva ut flera olika checklistor i samma process genom att göra flervalsmarkeringar med hjälp av checkboxrutor. Detta skulle minska klickerna i applikationen då användaren slipper välja en checklista i taget för utskrift. Vid dessa samtal valde vi att rita på de mockuper vi tillverkat, för att bygga på med nya intryck och förslag.

## ***Hur användaren agerade***

Under det första mötet sa användaren för fastighetssystemet, att han var positiv till att vi gjorde detta arbete. Medvetenhet fanns om att det inte skulle bli byggt något nytt system, men att vårt resultat kunde vara bra för framtida utveckling av systemet. Vi kunde även märka att det fanns stora brister i det befintliga systemet, men att användaren endast kunde peka på direkt synbara fel, som att en knapp var felplicerad eller att utskriften ur systemet var otydliga. Det ansågs även var för mycket ingångar i systemet genom menyn till vänster och det var önskvärt med ett annat upplägg för att få det mer strukturerat.

När vi vid det andra mötet hade med oss mock-updelar för att visa hur vi tänkt oss den nya strukturen för systemet kunde vi känna en viss negativ inställning från honom. Han sa, att det inte gav så mycket med att sitta och "leka" med pappersbitar utan det kunde vi göra själva och inte ta hans arbetstid till det. Vad han ville ha var ett system som fungerade i slutändan. Vi förklarade att om vi sitter hemma och inte får hans tankar för användandet av systemet, så finns risken att det på nytt blir ett oanvändbart system. När vi visade våra mock-uper frågade han hur vi hade tänkt vid olika händelser som sker i systemet. Vi förklarade hur vi menade att olika funktioner kan lösas och frågade om hans åsikter samt vad som önskades. I början var han ganska försiktigt med att flytta på olika mock-updelar men ju längre tid som vi satt där, ju mer pekande han och frågade han om olika saker.

Vid vårt tredje möte presenterade vi en vidareutveckling för ett nytt system, där hans synpunkter från föregående möte var med. Då kunde vi känna att intresset ökade. Han upplevde sig vara delaktig och det innebar en positiv förändring för systemet. Även denna gång pekade och frågade han om olika möjligheter och vad som händer vid olika val men nu flyttade han mer på olika mock-updelar i applikationen. Han insåg att slutresultatet kunde påverkas och att vi var intresserade av att utveckla det som är bra för honom som användare. Han hade börjat inse att pappersmodeller kan göra att icke-användbara system vidareutvecklas till något bättre. Vid detta tillfälle sades även, att det som ses i våra mock-uper är mer likt det system som han nu kan förstå att han önskat sig. Utan dessa mock-uper hade det förmodligen varit svårare att precisera sig, för att få oss att förstå. Under hela utvecklingen har han kunnat se om vi uppfattat honom och hans önskemål rätt samt och tvärt om.

Genom hela utvecklingsdelen har vi kommunicerat med hjälp av mock-uper, informella samtal och frågor, se bilaga 2. Vid vårt andra möte började vi använda oss av mock-uper. Vi styrde mycket av det som sades och vilka delar i den tänkta applikationen som borde flyttas på. Som vi tidigare sagt upplevde vi att han inte var helt positiv till detta arbetssätt. Vi märkte att han kände sig pressad och tyckte att vi höll på med något som var mindre viktigt för hans arbete. Han ansåg att vi inte hade rätt uppfattning om hur mycket tid som skulle avsättas för oss och vårt arbete. Vid tredje mötet med honom var han, som vi också sagt tidigare, mer positiv och såg att de delar som reflekterats över vid tidigare möten fanns med. Han var då mer positiv till att samtala, flytta och ge återkoppling på vårt arbete. Detta kändes mycket bättre. När vi gick därifrån kunde vi känna, att det vi gjort kunde vara en bra grund för vidareutveckling av det fastighetssystem som finns idag.



Vid våra olika möten med användaren fick vi även fram nya krav som han hade på systemet. Vissa av dessa var förtydliganden av krav som redan fanns men inte hade uppfyllts på det viset som önskats. Exempel på det är att det ska vara möjligt att skriva ut flera checklistor i en och samma funktion genom att använda sig av fler-valsmarkeringar.

## Diskussion av resultatet

Vi nämnde i inledning att det är viktigt med grundligt förarbete vid utveckling av system för att spara tid och pengar i olika verksamheter. För att göra system bättre är det viktigt att se till användarens behov och önskemål. Genom den snabba IT-utvecklingen kommer allt fler människor i kontakt med system av olika slag och påverkas av utvecklingen vare sig man önskar eller inte. Användaren måste acceptera att datoranvändningen ökar inom olika verksamheter.

När användaren inte var positiv till vårt arbete med hjälp av mock-uper kändes det tungt då risken fanns att han ännu en gång skulle få ett fastighetssystem, som inte uppfyllde hans krav. Vi valde ändå att gå vidare även om inte hans inställning till vår arbetsmetod var helt positiv, därför att vi trodde på idén med att mock-uper underlättar kommunikationen mellan två parter. När han märkte att vi uppfattat hans tankar om systemet ökade hans intresse och vi kunde se att mock-uper bidrog till bättre kommunikation. Vi tror att hans förmåga att uttrycka sig blev bättre eftersom kraven kunde preciseras och ges mer i detalj, som att användaren ska kunna skriva ut flera checklistor samtidigt. Vid utveckling av system är det svårt att läsa sig till hur system ska utformas. Litteraturen garanterar inte att ett system blir bra. Det är därför viktigt att kommunikationen är bra.

Med hjälp av mock-uper har vi kommit fram till att man kan få fram många idéer och konkreta förslag på hur system ska se ut. Dessa mock-uper kan bestå av allt från struktur till väldigt detaljerade bilder. Vi ser arbetssättet med mock-uper som ett bra sätt att arbeta på då de är positivt för kommunikation och verkar för god utvecklingspotential. Exempel på tillvägagångssätt för att samla information kan vara att göra intervjuer med den målgrupp produkten är riktad mot. Däremot vet inte användaren alltid vad han/hon vill ha. Det kan vara svårt att veta vilka de egna behoven är. Vi förespråkar därför att dessa intervjuer genomförs i form av informella samtal. Vid utveckling är det inte alltid som användaren har några givna svar på frågor som ställs. Svaren måste få växa fram och det anser vi att de gör bäst vid dessa samtal. Det är bra att låta slutanvändaren få testa en prototyp i form av mock-uper. Användaren kan, med förslaget framför sig i pappersform och inte bara uttryckt i ord, ge direkt återkoppling. Ett användbart gränssnitt kräver att man tar in information utifrån, det vill säga från användaren. För att nå goda resultat är det viktigt att utveckling sker i en process där användaren deltar.

Vi kan inte säga att vårt slutresultat för Svenska kyrkan i Ronnebys fastighetssystem hade blivit bättre, men vi kan konstatera att han nu har en ökad förståelse för hur slutprodukten kan komma att se ut. Samtidigt har vår förståelse ökat för hur användaren önskar ha systemet. Genom processen har han kommit underfund med nya krav och förslag som han har på systemet. Vårt arbete kan han ha som ett bra underlag för kommande utveckling av fastighetssystemet. Ju bättre kommunikationen är mellan användare och utvecklare, ju lättare har man att förstå den andres problem och därigenom komma med konstruktiva lösningar. Genom att arbeta med en iterativ modell som livscykelmodellen har vi kunnat beakta krav som uppkommit genom processen vi har haft med användaren i hela utvecklingen. Vid användning av vattenfallsmodellen har inte utvecklaren samma möjlighet. Detta då användaren inte finns med genom

hela processen. Det är viktigt för utvecklaren att man kan stämma av med användaren, att man uppfattat varandra på samma sätt och inte bygger sin utveckling på något som inte stämmer med kraven. Vid användning av Andersens livscykelmodell som är iterativ finns denna möjlighet under hela utvecklingsprocessen.

## Förslag på förändringar

Användarens befintliga fastighetssystem fungerar till stora delar som det ska men användaren känner ändå att det inte finns tillräckligt logisk struktur och uppbyggnad av det. För att få en klarare bild av detta har vi tagit fram konkreta förbättringsförslag som vi anser skulle göra det mer strukturerat och tydligt. Istället för att användaren ska välja en viss funktion i menyn till vänster ska denna funktion finnas på en sammanfattande sida där alla egenskaper för byggnader/installationer/system finns. Här ska det vara möjligt att redigera/ta bort/lägga till egenskaper för vald del. Detta gör att användaren till exempel får en tydlig överblick för vilka egenskaper som finns för en byggnad. För att öka överskådligheten för vilka sidor som finns under en viss ort har vi valt att även presentera dessa i en trädstruktur till vänster under vald ort. Vårt förslag för hur antalet klick i applikationen ska minska är, att det fälls ner dropdown-menyer med undermenyer, när användaren för pekdonet över respektive ort. Eftersom andra personer, som inte är helt vana vid applikationen också ska kunna ändra egenskaper för de olika delarna i systemet, är det viktigt att det är väl strukturerat. Det tycker vi, och även användaren efter våra samtal och kommunikationer, att det nu är.

Som svar på vår frågeställning om Svenska Kyrkan i Ronneby skulle få ett mer användbart informationssystem, om systemutvecklarna hade använt sig av riktlinjer och erfarenheter som bearbetas inom området HCI, anser vi att de hade fått det. I den dokumentation som projektgruppen har sammanställt, kan vi se alla delar som önskats finnas med i systemet. Vi kunde dock inte läsa något om hur det grafiska gränssnittet skulle se ut. Ju mer vi talade med honom, ju mer förstod vi hur han önskade att det skulle vara uppbyggt. Genom vår kommunikation med hjälp av mock-uper byggde vi sedan upp ett nytt förslag på system där allt var tydligt gestaltat för både användaren/kunden och utvecklaren.

Efter det arbete som vi genomfört anser vi att kommunikationen förbättrats med hjälp av mock-uper. Även om det kan kännas lite ”billigt” och ”enkelt” att göra modeller i papper menar vi att metoden innebär ett stort stöd. På ett lättföränderligt sätt kan man testa olika förslag för samma problem.

## Slutord

Att ha fått jobba med detta fastighetssystem har varit bra träning för oss som systemutvecklare. Vi fick sätta oss in i någon annans system och fått försöka bygga vidare på material som redan fanns. Vi märkte snabbt vikten av att dokumentera allt som görs inom ett projekt, speciellt när det gäller kommunikation med användaren. Det är svårt att sätta sig in i ett system som inte är väl förklarat. Detta kan då resultera i att utvecklarna får göra stora delar en gång till, vilket kan innebära så väl tid som pengar



för kunden. Ett bra sätt att dokumentera kommunikationen är att ta foton på gjorda mock-uper, som förslagsvis kompletteras med beskrivningstext och/eller ljudupptagning.

Vårt resultat tycker vi är en bra grund för kunden att bygga vidare på vid vidareutveckling av fastighetssystemet.

### **Framtida forskningsområden**

Efter att vi avslutat denna uppsats finns det obesvarade frågor kvar inom detta område. Vi skulle tycka det var intressant att se en eventuellt mer etablerad användning av mock-uper i systemutveckling och i samband med det, förändringar av systemutvecklingsmodeller och praktiker.

## Litteraturförteckning

### Litteratur

- Andersen, Erling S. (1994) *Systemutveckling – principer, metoder och tekniker*, Studentlitteratur, Lund.
- Beyer, H. Holtzblatt, K. (1997) *Contextual design: Defining customer-centered systems*. Morgan Kaufmann.
- Dawson C. W. (2000) *The Essence of Computing Projects*, Prentice hall, London.
- Löwgren, Jonas, Stolterman, Erik. (1998) *Design av informationsteknik – materialet utan egenskaper*, Studentlitteratur, Lund.
- Patel, R, Davidsson, B. (1994) *Forskningsmetodikens grunder Att planera, genomföra och rapportera en undersökning*, Studentlitteratur, Lund.
- Preece, Jenny. (1994) *Human-Computer Interaction*, Addison-Wesley Publishing company, New York.
- Preece, Jenny, Roger, Yvonne, Sharp, Helen. (2002) *Interaction Design*, John Wiley & Sons Inc, Crawfordsville.
- Rubinstein, R, Hersh, H (1984) *The Human Factor: Designing Computer Systems for People*, MA: Digital Press, Woburn.
- Sommerville, I. (2001) *Software engineering*, Pearson Education Ltd, Harlow.
- Whiteside, J., Wixon, D. (1987) *The dialectic of usability engineering*. Elsevier, Amsterdam.

### Otryckta källor

- Föreläsningar i Människa-datorinteraktion, dvc002
- Intervjuer med personal på Svenska kyrkan i Ronneby
- Projektdokumentation för Svenska kyrkan i Ronnebys fastighetssystem, SALCCA, Blekinge Tekniska Högskola, ht 2002.

# Sammanfattning av kravspecifikation

## Funktionella krav

### Användare

- Det ska finnas en huvudadministratör
- Vissa personer ska kunna logga in och ändra uppgifter
- Annan personal ska inte kunna logga in utan bara ta del av information.

### Skapa felanmälan

- Användaren ska kunna skriva in en felanmälan i ett formulär och skriva i valfri rubrik.
- Felanmälan ska skickas till den ansvarige via e-post.

### Ta bort felanmälan

- Den ansvarige som fått en felanmälan ska kunna ta bort en specifik felanmälan.

### Lägga till

#### Användaren ska kunna

- lägga till en bild när nya fastigheter köpts eller när nya installationer skett
- lägga till och skapa nya checklistor för olika byggnader
- skriva checklista
- skriva och lägga till text till bilder, byggnader och orter
- skriva och lägga till text för en viss sorts installation
- skriva och lägga till text för ett system
- lägga till och ändra sitt lösenord för att få tillgång till fastighetssystemet, mellan 6 och 10 tecken
- lägga till användarnamn, mellan 6 och 10 tecken – detta ska enbart huvudadministratören kunna göra.

### Ta bort

#### Användaren ska kunna

- ta bort bilder
- ta bort checklista
- ta bort text för bilder, byggnader och orter
- ta bort text för en viss sorts installation
- ta bort text för ett system
- ta bort lösenord för att få tillgång till systemet
- ta bort användarnamn - detta ska enbart huvudadministratören kunna göra.

### Ändra

#### Användaren ska kunna

- ändra text och checklistor
- ändra text för bilder, byggnader och orter
- ändra text för en viss sorts installation
- ändra text för ett specifikt system
- ändra lösenord



- ändra användarnamn

#### Varningsmeddelanden

##### Användaren ska få information

- när något läggs till i systemet
- när något ändras i systemet
- när något tas bort ur systemet

#### Inloggad - utloggad

- För att kunna ändra i systemet måste användaren vara inloggad
- En användare som är inloggad ska även kunna logga ut
- En användare som varit inaktiv i mer än 10 min loggas ut automatiskt
- En användare som inte är inloggad ska kunna läsa all information på hemsidan men inte redigera den
- När användaren loggar ut ska det visas ett meddelande som talar om att utloggningen är gjord

#### Uppdatering

- Sista uppdateringsdatum ska alltid synas och vara placerat uppe i höger hörn
- Alla delar som sparas ska katalogiseras på ett logiskt sätt, till exempel ska alla bilder som sparas i systemet ligga under samma mapp.

#### Skriv ut

- Checklistor ska kunna skrivas ut

### **Icke-funktionella krav**

#### Sidans utseende

- Länkar ska vara blå
- Länkar ska vara understrukna när pekdonet förs över dem
- Menyn ska alltid vara synlig till vänster på sidan
- I menyn ska de fem orterna vara listade enligt följande: Ronneby, Kallinge, Bredåkra, Saxemara, Møljerud. Därefter allmän information, checklistor, logga in och felanmälning.
- När användaren trycker på en länk ska den valda sidan visas
- ”Tillbaka”-knappar ska finnas i applikationen där det behövs
- Systemet ska innehålla ett bildspel som visar bilder med tillhörande text för dem som inte har fulla rättigheter till systemet
- Svenska kyrkans logo ska alltid finnas i övre vänstra hörnet. Under logon ska det stå ”Svenska kyrkan i Ronneby”
- Teckensnitt ska vara ”Veranda” eller ”Arial”

#### Tydligt användargränssnitt

- Lätt att navigera
- Ingen onödig text eller bilder på sidorna
- Innehållet på sidan ska visas strukturerat och logiskt
- Språket som användas på hemsidan ska vara svenska

## **System krav**

- Databas ska vara MySQL
- Programmeringsspråk är JDK (Java Development Kit) version 1.4
- Operativsystem ska vara Windows 98 eller senare
- Webbbrowser ska vara Microsoft Internet Explorer 5.5 eller senare
- Webbservern måste kunna köra Java-servlets

## **Dokumentation**

- Användarmanualen ska vara på svenska
- Kontrakt och kravspecifikation skrivs på engelska

## **Sammanställning av intervjufrågor till användaren**

Vad är bra och vad är dåligt i systemet?

Finns det några saker i systemet som du skulle önska ändra på?

Vad är det som inte fungerar i systemet som du önskar?

Finns det fler funktioner som du skulle vilja ha med i systemet och var skulle dessa placeras?

Har du själv några idéer på förbättringar?

Har du själv några skisser på förslag på förbättringar efter att du fått projektgruppens system?

Vad tycker du om detta sättet att jobba på?

Vad tycker du om att vara med i utvecklingen?

Tycker du att arbetssättet med mock-uper är bra för dig som kund?

Känner du dig mer delaktig i utvecklingen vid detta sätt att arbeta?

Är det lättare att som kund påverka utvecklingen?

Är det lättare att förstå hur systemet kommer se ut om man som kund får se och flytta innan utvecklaren börjar implementera systemet?