



Användargränssnitt för mobiltelefoner

av

Karolina Sjöberg och Elisabeth Söderberg

Abstrakt

Vi har i vår undersökning inriktat oss på att ta reda på om riktlinjer för traditionell användargränssnittsdesign går att applicera på gränssnitt för mobiltelefoners små skärmar. Vi har även undersökt vad man bör tänka på för att kunna utforma ett användbart gränssnitt för mobiltelefoner. Undersökningens resultat bygger på litteraturstudier och utveckling av en applikation som genomgått användbarhetstestning. Vi har kommit fram till att det i stor utsträckning går att tillämpa de traditionella riktlinjerna även för mobiltelefongränssnitt, men att de bör anpassas efter det begränsade utrymmet på mobiltelefonens skärm. Vi har även sett att de traditionella riktlinjerna bör kompletteras med ytterligare riktlinjer som är mer specifika för att passa telefonernas små skärmar. Mobiltelefonens skärmstorlek begränsar gränssnittets användbarhet, men vi tror att användare kan ha överseende med detta eftersom det är just mobiltelefonens storlek som gör den till ett användbart verktyg. Att telefonen är liten medför att användaren alltid kan ha den med sig och ständigt ha tillgång till information, oberoende av tid och rum.

Magisteruppsats 20 poäng
Vårterminen 2000
Handledare: Wera Tegner Johansson

*Vi vill tacka **Anna-Karin Arvidsson**, vår handledare på Semcon IT Consulting. Hon har varit till stor hjälp under vårt arbete och har ställt upp med ovärderliga faktakunskaper. Mest av allt vill vi tacka henne för alla idéer, inspiration och entusiasm.*

*Vi vill även tacka **Semcon IT Consulting** som gjorde det möjligt för oss att prova ny teknik och undersöka hur en applikation till en mobiltelefon kan utformas.*

***Wera Tegner Johansson**, vår handledare på Institutionen för informatik har också varit till hjälp och uppmuntrat oss att göra denna magisteruppsats.*

Vi är också tacksamma för de insatser som de personer som medverkade vid användbarhetstesterna har bidragit med.

INNEHÅLLSFÖRTECKNING

1. INLEDNING.....	5
1.1 Mobil IT	5
1.2 Bakgrund och frågeställningar.....	5
1.3 Avgränsningar.....	6
1.4 Disposition	6
2. PSYKOLOGIS INFLYTANDE VID UTFORMNING AV ANVÄNDARGRÄNSSNITT.....	8
2.1 Människa-datorinteraktion.....	8
2.2 Kognitiv psykologi	8
2.2.1 Användarens kognitiva förmåga	9
2.2.2 Minnet.....	9
3. ANVÄNDBARHET	14
3.1 Användbarhet på webben	15
4. UTFORMNING AV ANVÄNDARGRÄNSSNITT.....	16
4.1 Användargränssnitt.....	16
4.2 Allmänt om principer och riktlinjer.....	17
4.3 Åtta principer för gränssnittsdesign	17
4.4 Riktlinjer för inmatning av data	20
4.5 Utformning av menyer	21
4.5.1 Rubrik	21
4.5.2 Ordval	21
4.5.3 Menyers utseende.....	22
4.6 Att fylla i formulär.....	22
4.6.1 Riktlinjer vid design av formulär	22
5. UTFORMNING AV ANVÄNDARGRÄNSSNITT PÅ WEBBEN	24
5.1 Skillnad mellan traditionella gränssnitt och webbgränssnitt.....	24
5.2 Riktlinjer för webbgränssnitt	24
5.3 Kompromisser om användbarhet.....	24
5.4 Navigering	25

5.5 Sammanfattning av riktlinjer för webbdesign	26
6. UTFORMNING AV ANVÄNDARGRÄNSSNITT FÖR MOBILTELEFONER 28	
6.1 WAP.....	28
6.1.1 WML.....	29
6.1.2 WAP-strukturen	30
6.2 Applikationer för mobil användning.....	30
6.2.1 Stora skärmar kontra små skärmar	31
6.2.2 Navigeringsprinciper.....	32
6.2.3 Text på små skärmar	33
6.2.4 Rubriker	33
6.2.5 Användning av länkar	33
6.2.6 Inmatning av text.....	33
6.2.7 Konsekvent terminologi.....	34
7. METOD.....	35
7.1 Prototyping.....	35
7.1.1 Vad är prototyping?	35
7.1.2 Användarmedverkan	37
7.1.3 Sammanfattningsvis	38
7.2 Användbarhetstestning.....	38
7.2.1 Vad är användbarhetstestning?.....	38
7.2.2 När är ett användbarhetstest lämpligt?	39
7.2.3 Användbarhetstestning jämfört med forskningsstudier.....	40
7.2.4 Planering	40
7.2.5 Att mäta användbarhet	42
7.2.6 Användbarhetstest av webbsidor.....	43
8. GENOMFÖRANDE AV STUDIE	44
8.1 Användbarhetstest av prototypens gränssnitt.....	44
9. RESULTAT OCH DISKUSSION	46
9.1 Utformning av applikationen.....	46
9.2 Resultat och diskussion av användbarhetstesterna.....	47
9.2.1 Mobiltelefonvana	47
9.2.2 Anpassa efter typ av mobiltelefon.....	47
9.2.3 Rubriker	48
9.2.4 Mycket information på skärmen	49
9.2.5 Inmatning av text.....	50
9.3 Hur traditionella principer går att applicera på mobiltelefongränssnitt.....	50
9.3.1 Konsekvent utformning.....	50
9.3.2 Kortkommandon	51
9.3.3 Feedback	51
9.3.4 Gruppering	52
9.3.5 Felhantering	52
9.3.6 Gör det möjligt för användaren att ångra val.....	53
9.3.7 Användarkontroll	53

9.3.8 Minimalt krav på användarens minne	53
10. SLUTSATSER.....	54
10.1 Går det att på mobiltelefonernas små skärmar behålla de egenskaper som ett traditionellt användargränssnitt bör ha för att vara användbart?	54
10.2 Vad bör man tänka på när man utformar ett användargränssnitt för mobiltelefoner så att det, ur användarnas synpunkt, blir överskådligt och lätt att navigera i?	55
10.3 Reflektioner	56
11. REFERENSER	58

1. Inledning

1.1 Mobil IT

Dagens samhälle domineras i allt högre grad av informationsteknologi. Tekniken har gjort det möjligt för oss att kunna vara mer mobila, både i arbetet och i privatlivet. Ökad användning av mobil teknik gör att människor kan vara tillgängliga oberoende av tid och plats. Många människor lever ett rörligt liv och har därför behov av att kunna kommunicera och utföra vissa uppgifter oavsett var de råkar befinna sig.¹ Mobil informationsteknik har visat sig öka räckvidden för många verksamhetskritiska applikationer och gör dem tillgängliga för användarna när som helst och var som helst.²

De stora kanalerna för informationsinhämtning respektive kommunikation, dvs Internet och mobiltelefoner har slagits ihop och mobilt Internet har blivit ett allmänt känt begrepp.³ Att göra Internet tillgängligt via mobiltelefonen är en relativt ny företeelse, men det förutspås att en tredjedel av Europas befolkning, dvs mer än 219 miljoner personer, år 2004 regelbundet kommer att använda mobiltelefoner för att utnyttja Internettjänster. Dessutom planerar 90% av de företag som bedriver e-handel att optimera sina Internetsidor så att de kan användas i mobiltelefoner.⁴ Med andra ord kommer mobiltelefoner och deras användning om några år att bli ännu mer intressanta ur designsynpunkt.

Traditionell användning av Internet via PC:n är fortfarande det bästa sättet att utbyta information eftersom skärmen är stor och tangentbordet är lättanvänt. Ett bra komplement är att ha tillgång till Internet på mobiltelefonen. Användare har nästan alltid mobiltelefonen inom räckhåll vilket gör att de, oberoende av var de befinner sig, har möjlighet att snabbt och smidigt få tillgång till önskad information. Denna egenskap gör mobiltelefonen till ett användbart verktyg. Mobiltelefonen har visat sig utgöra en viktig del i människors liv och har nästan blivit en del av folks personlighet. Få personer lämnar sina hem utan att ta med sig plånbok, nycklar och mobiltelefon.

Mobila applikationer kommer att användas av en varierande grupp användare och inom många olika områden. Därför är det viktigt att mobiltelefonens användargränssnitt utformas på ett sätt som underlättar och hjälper användarna att ta tillvara information eller utföra en viss uppgift.

1.2 Bakgrund och frågeställningar

Det är relativt enkelt att utforma ett användargränssnitt. Utmaningen ligger i att skapa ett gränssnitt som tar hänsyn till människans förmåga och beteende – egenskaper vi varken kan påverka eller förändra.⁵ Det handlar i grund och botten om att utforma gränssnitt som är anpassade till människan, så att arbetsuppgifter kan underlättas, bli säkrare, och korrekt utförda. En utmaning vad gäller utformning av användargränssnitt

¹ Dahlbom, B., Ljungberg, F. (2000). *Mobile Informatics*

² Wiråker, P., Aspiro. (2000). *Life in motion*

³ Stedt, P. (1999). *WAP pratar alla om?*

⁴ wap.com. (2000) *What is WAP ?*

⁵ Eriksson, M. (1997). *Människa-dator interaktion*

till mobiltelefoner är att utforma ett användbart gränssnitt att navigera i, trots att displayen har begränsat utrymme för att visa information.

Vi har i vår magisteruppsats undersökt hur användargränssnittet till en mobiltelefon bör designas för att, trots den primitiva terminalen, passa in i användarens mobila sammanhang. Vi har valt att inrikta oss på följande frågor:

- Går det att på mobiltelefonernas små skärmar behålla de egenskaper som ett traditionellt användargränssnitt bör ha för att vara användbart?
- Vad bör man tänka på när man utformar ett användargränssnitt för mobiltelefoner så att det, ur användarnas synpunkt, blir överskådligt och lätt att navigera i?

För att kunna undersöka ovanstående frågor tyckte vi det var viktigt att själva utveckla och testa en mobil applikation. Vi fick möjlighet att i samarbete med Semcon IT Consulting utveckla en mobil klient till deras tidsrapporteringsystem och fick i och med detta möjlighet att praktiskt testa användargränssnittet till en mobil applikation.

1.3 Avgränsningar

Vi har valt att endast undersöka användargränssnitt för mobiltelefoner och har valt att inte behandla hur användargränssnitt ska utformas för andra typer av terminaler som har små skärmar. Vi har enbart inriktat oss på hur den teknik som finns idag ska användas för att utveckla grafiska gränssnitt och har inte undersökt andra alternativa gränssnitt såsom t.ex. ljud och röststyrd interaktion.

1.4 Disposition

Kapitel 1, *Inledning*, beskriver uppsatsens bakgrund och frågeställning, samt de avgränsningar vi valt att göra.

Kapitel 2, *Psykologins inflytande vid utformning av användargränssnitt*, behandlar människa-datorinteraktion och kognitionspsykologin där bland annat människans minnesprocesser och förmåga att behandla och organisera information tas upp.

Kapitel 3, *Användbarhet*, beskriver begreppet användbarhet och varför det är viktigt att system utformas på ett användbart sätt.

Kapitel 4-6, redogör för några principer och riktlinjer som kan tillämpas vid utformning av användargränssnitt. Det behandlar traditionella gränssnitt, gränssnitt för webbdesign och gränssnitt för mobiltelefoner.

Kapitel 7, ger en beskrivning av de metoder vi har använt oss av för att undersöka hur man utformar ett användbart gränssnitt.

Kapitel 8, *Genomförande av studie*, ger en inblick i hur vi gick tillväga när vi utvecklade och testade en mobiltelefon applikation och dess gränssnitt.

Kapitel 9, *Resultat och diskussion*, redogör för resultat av användartesterna och diskuterar vilka traditionella kriterier som går att applicera och vad man bör tänka på vid utformning av användargränssnitt till mobiltelefoner.

Kapitel 10, innehåller våra slutsatser och eventuella ämnen för fortsatt forskning som framkommit under arbetets gång, men som inte har avhandlats i denna uppsats.

2. Psykologins inflytande vid utformning av användargränssnitt

Följande kapitel handlar om människa-datorinteraktion och undersöker vilka kognitionspsykologiska faktorer som påverkar människans sätt att hantera information.

2.1 Människa-datorinteraktion

För att förstå vad som sker när människor använder datorer är det nödvändigt med en viss grundläggande förståelse av användaren själv. Det är viktigt att förstå hur användare tänker och tar emot information, hur människor samverkar i sitt arbete, hur mjuk- och hårdvara ska skapas för att stödja användare, hur informationssystem påverkar människor, hur metoder ska skapas för att bygga mjukvara, hur man utvärderar applikationer osv.⁶ Människa-datorinteraktion (MDI) är ett forskningsområde som syftar till att framställa programvara med en hög acceptans hos användaren.⁷ MDI innefattar många olika vetenskapliga discipliner men de fyra viktigaste anses vara: ergonomi, datavetenskap, kognitionspsykologi och organisations- och beteendevetenskap.⁸

Att förstå användarens interaktion med datorn har varit det centrala målet med MDI. Ett program måste anpassas till den person som slutligen ska använda produkten, men det kan till exempel vara svårt att definiera hur en representativ användare av en ordbehandlare ser ut. Detta kan jämföras med att försöka beskriva en typisk användare av en hammare. Då framstår användarbegreppet som verklighetsfrånvänt. En snickare är inte expert på att använda en hammare utan på att utforma strukturer i trä. På samma sätt är de personer som sitter framför datorerna inte i första hand professionella datoranvändare utan snarare experter på ett arbete som kräver datorstöd och det viktiga är att programmet blir ett användbart redskap. På samma sätt som en bra hammare måste utformas olika beroende på vad den ska användas till, bör program också utformas olika beroende på vilken arbetsuppgift det ska understödja.⁹

Hur information presenteras på dataskärmen påverkar de beslut som användaren fattar. Lämplig information och funktionalitet måste finnas tillgänglig på ett bra sätt för att användaren ska kunna fatta riktiga beslut. Syftet med ett system är att stödja utförandet av en uppgift och inte att göra det svårare att utföra den. Om det används för säkerhetskritiska uppgifter, kan felaktigt utformade gränssnitt leda till direkt fara, för egendom eller för liv.¹⁰

2.2 Kognitiv psykologi

Informationsteknikens spridning i samhället har lett till att personer med högst varierande bakgrund, kultur och kunskap använder sig av datorer och datorsystem. Detta för bland annat med sig att datorernas gränssnitt måste vara lätta att använda. En förutsättning för att utveckla lättanvända system är att man förstår hur människan

⁶ Lif, M. (1998). *Adding Usability*

⁷ Oestreicher, L. (1994). *Vad är MDI?*

⁸ Eriksson, M. (1997). *Människa-Dator Interaktion*

⁹ Oestreicher, L. (1994). *Vad är MDI?*

¹⁰ Eriksson, M. (1997). *Människa-Dator Interaktion*

tänker, uppfattar sin omgivning och hur hon tar till sig kunskap och information. Kognitionsteorin beskriver hur människan uppfattar sin omvärld och använder sin kunskap om denna. Detta är avgörande för design av användbara gränssnitt.¹¹

2.2.1 Användarens kognitiva förmåga¹²

Psykologi handlar till stora delar om att förstå mänskligt beteende och vad som ligger bakom det. Kognitiv psykologi kan beskrivas som studier av mänskliga mentala processer och deras roll vid tänkande, känsel och uppträdande. Bland de processer som studeras kan nämnas uppfattning, minne, uppmärksamhet, inläring, problemlösning, kreativitet, beslutsfattande och resonemang.¹³ Det är ofta inte nödvändigt att ge en fullständig bild av hur människans kognitiva processer fungerar för att förklara deras viktiga betydelse vid förståelse av gränssnitt.¹⁴ Det finns två olika slags kognitiva processer, den automatiska och den sekventiella. Den automatiska, även kallad den undermedvetna nivån, tar emot och behandlar flera miljoner sinnesimpulser varje sekund.¹⁵ Detta innebär att vi utan att medvetet tänka på det kan behandla ett stort antal informationsprocesser samtidigt. Exempelvis kan en bilförare vid ett och samma tillfälle registrera vägkrökning, motorljud, ratt rörelser m.m. utan att tänka aktivt på det. Process nummer två är den sekventiella, den medvetna nivån, och den gör bl.a. att vi kan göra bedömningar och hantera osäker information. Denna analytiska och logiska förmåga är stor, men kapaciteten är starkt begränsad, vilket innebär att vi på den sekventiella nivån endast kan hantera en informationsprocess i taget. Vi är alltså bara medvetna om en liten del av alla de informationsprocesser som ständigt behandlas i våra hjärnor.

Med hjälp av kognitiv psykologi kan man alltså förbättra ett gränssnitts utformning genom att exempelvis förutsäga vad användare kan, inte kan, vad de förväntas göra och förklara hur människan uppnår uppsatta mål. Man kan även identifiera och förklara det som orsakar de problem som användaren stöter på, samt erbjuda modellverktyg och metoder för att bygga gränssnitt som är lätta att använda.

2.2.2 Minnet

Minnet är en egenskap som gör det möjligt att lagra information från ett tillfälle till ett annat och vi förstår omvärlden i termer av den information som finns i minnet. Människans minne och minnesprocesser har en central betydelse för samspelet med datorer. Människans förmåga att använda datorn är helt beroende av egenskaperna hos det mänskliga minnessystemet. För att förstå hur vårt minne fungerar och för att kunna lära oss använda det på ett bra sätt är det viktigt att försöka förstå de mekanismer som styr minnet.

Människans minne kan delas in i tre olika delar: sensoriskt minne (SM), korttidsminne (KTM) och långtidsminne (LTM).¹⁶ Figur 2.1 beskriver sambandet mellan de olika minnesdelarna.

¹¹ Lif, M. (1998). *Adding Usability*

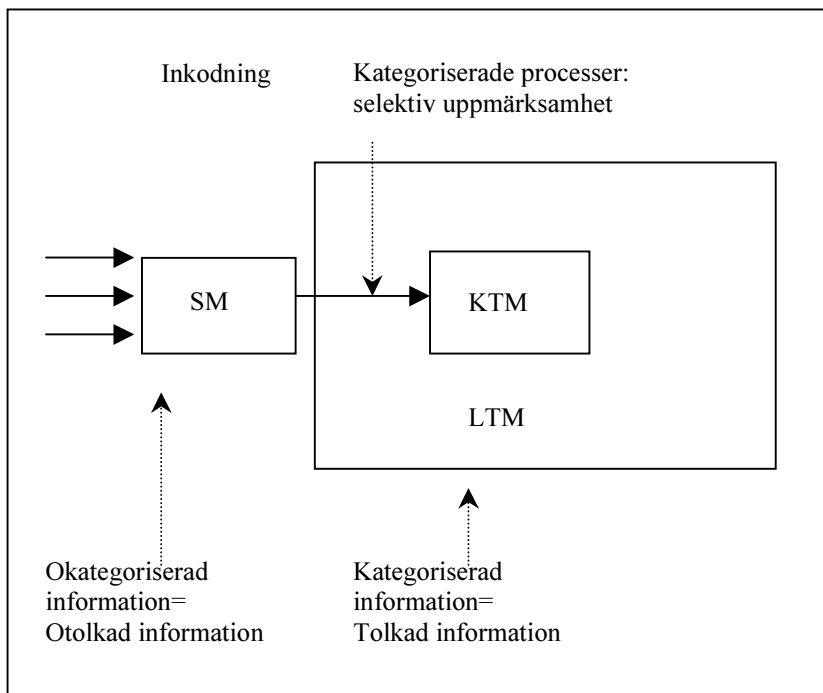
¹² Allwood, C. (1998). *Människa-datorinteraktion*

¹³ Ashcraft, M H. (1994). *Human Memory and Cognition*

¹⁴ Lif, M. (1998). *Adding Usability*

¹⁵ Andersson, J. (1990). *Cognitive Psychology and its implications*

¹⁶ Allwood, C. (1998). *Människa-datorinteraktion*



Figur 2.1: En allmän modell över människans minne
 Källa: Allwood C, (1998) Människa - datorinteraktion

Att lära sig något och att senare komma ihåg det, kan delas in i de tre olika faserna: inkodning, lagring och framtagning. Inkodning sker när information från omvärlden når minnet. Lagring innebär att information lagras i minnet i form av en föreställning eller färdighet. Framtagning sker när information i minnet plockas fram och aktiveras. Det sensoriska minnet är främst aktuellt under inkodningsfasen, medan kort- och långtidsminne är aktuella under alla tre faserna.¹⁷

2.2.2.1 Sensoriskt minne

Människan registrerar information från omgivningen via det sensoriska minnet (SM). Där stannar informationen helt otolkad innan den, efter att ha relaterats till individens förkunskaper i långtidsminnet, förs över till korttidsminnet. All information som hamnar i det sensoriska minnet placeras inte automatiskt i korttidsminnet, utan korttidsminnets begränsade storlek och kapacitet gör att hjärnan omedvetet selekterar ut den information som för tillfället verkar mest relevant och placerar det sedan i korttidsminnet.

Ögats sensoriska minne kallas för det ikoniska minnet. Den tid som ögats ikoniska minne håller kvar information har beräknats till ca en fjärdedels sekund, men detta kan variera något beroende på bakgrundens ljusförhållanden, kontrasteffekter, m.m. Ny information som tas in i det ikoniska minnet kan störa övrig information som redan finns där, och människan hinner inte uppfatta och kategorisera information innan den försvinner från det ikoniska minnet. Därför är det viktigt att inte samtidigt visa för mycket information på bildskärmen, eftersom användaren inte medvetet hinner med att uppmärksamma allt innehåll.¹⁸

¹⁷ Lif, M. (1998). *Adding Usability*

¹⁸ Lansdale, M W., Ormerod, T C. (1995). *Understanding Interfaces: A Handbook of Human-Computer Dialogue*

2.2.2.2 Korttidsminnet

Inom kognitionspsykologin definieras korttidsminnet (KTM) som ett minne där information av olika slag som t.ex föreställningar, känslor, målstrukturer, m.m. har en hög grad av aktivering. KTM använder vi för att lagra små mängder information under en kortare period, under tiden vi använder informationen. KTM är inte bara en passage mellan det sensoriska minnet och långtidsminnet utan även den instans där informationen behandlas för att användas. KTM kallas därför också för arbetsminne.¹⁹

Det är bland annat KTM som hjälper oss att rikta vår uppmärksamhet till den del av informationen som vid ett visst tillfälle är viktig för oss, till exempel att fokusera på en textrad åt gången istället för att se en hel textmassa. I KTM måste information lagras mellan 5 till 20 sekunder för att den ska stanna kvar i långtidsminnet. Människans KTM är begränsat och kan bara hantera en begränsad mängd information dvs hålla kvar 5 till 9 enheter samtidigt. En enhet kan vara en siffra, en bokstav eller ett ord. Det har visat sig att en person har lättare för att komma ihåg fler enheter om de är organiserade på något sätt, till exempel att bokstäver bildar ett ord eller att ord ingår i en mening. Gestaltpsychologer har visat att människor uppfattar element som är placerade nära varandra som om de hör ihop, och att element som utseendemässigt liknar varandra uppfattas som en enhet, en gestalt. Människan har förmågan att gruppera och tillföra information till objekt som ser ut eller är placerade på ett visst sätt.²⁰ Det är till exempel lättare att komma ihåg ett telefonnummer som är grupperat 031-48 75 87 än en rad med siffror 031487526. Genom att gruppera data kan människans KTM innehålla upp till 100 enheter. Människan har också lättare för att minnas något om hon kommer ihåg innebörden av de skapade orden eller meningarna och om de sätts in i ett sammanhang.²¹

Kunskap om perception är mycket användbart när man ska besluta om hur information på skärmbilden ska grupperas, vilka färger, typsnitt och storlekar som ska användas för att underlätta läsning och sökning för användaren. Om en samling objekt alltid har samma spatiala placering på skärmen eller om objekten alltid har samma unika färg och form, framträder till slut mönster som användaren kan använda som ledning när han eller hon använder programmet.²²

Information i korttidsminnet kan försvinna när nya enheter bildas, detta beror på att ny information stör den information som redan finns i korttidsminnet. Därför bör man som regel inte visa för mycket information samtidigt på bildskärmen. Om enheternas aktivitet inte förnyas kommer informationen att försvinna ur minnet. Men att informationen inte blir kvar i KTM innebär inte nödvändigtvis att den går förlorad, utan en del av den förs över till långtidsminnet.

I många fall kan det trots allt vara nödvändigt att visa mycket information på skärmbilden. Till exempel ska användaren helst inte behöva växla skärmbild mitt i ett beslut eftersom undersökningar har visat att de då riskerar att fatta sämre beslut, arbeta

¹⁹ Allwood, C M. (1998). *Människa-datorinteraktion – ett psykologiskt perspektiv*

²⁰ Lif, M. (1998). *Adding Usability*

²¹ Allwood, C M. (1998). *Människa-datorinteraktion – ett psykologiskt perspektiv*

²² Lif, M. (1998). *Adding Usability*

långsammare och uppleva arbetet som mer stressat.²³ Den mängd information som ska visas måste därför övervägas och anpassas till varje specifik situation.

2.2.2.3 Långtidsminnet

Långtidsminnet (LTM) innehåller information om det man tidigare har varit med om och det som man lärt sig. Vårt LTM har i kontrast till KTM möjlighet att lagra en oändligt stor mängd information.²⁴ När vi upplever att vi har glömt någonting har informationen inte försvunnit utan den är bara organiserad på ett mindre bra sätt. Glömska kan förklaras genom att likna minnet vid ett bibliotek. Liksom man i ett enormt stort bibliotek inte hittar en bok på grund av att man letar på fel hylla kan man i minnet förlora information genom att man inte använder lämpliga sätt att påminna sig om materialet.

LTM fungerar som en förvaringsplats för mentala representationer av olika typer av innehåll t.ex. föreställningar, kunskaper och färdigheter. Det innehåll i LTM som man vet mest om är våra föreställningar och färdigheter. I detta sammanhang skiljer man på två slags kunskaper: deklarativa och procedurella kunskaper. Deklarativa kunskaper består av olika typer av sakkunskaper bland annat fakta och regler dvs ”veta att”, och de procedurella kunskaperna av färdigheter, dvs ”veta hur”. Procedurell kunskap hanterar våra motoriska kunskaper och lärs in genom demonstration och övning, de innehåller information t.ex om hur man spelar gitarr, åker skidor, etc. De deklarativa kunskaperna är av två slag: episodiska och semantiska kunskaper där episodisk kunskap innehåller personliga minnen och upplevelser, medan semantiska kunskaper består av allmänna sakförhållanden som t.ex kunskaper om världen, regler, m.m.

Med hjälp av deklarativ kunskap kan användaren bygga upp en förståelse för hur systemet och programmet fungerar, genom att skapa en mental modell av systemet. Det är i och för sig möjligt att användaren skulle kunna interagera med systemet nästan enbart utifrån sina färdighetskunskaper men denna interaktion skulle många gånger vara osäker. När användarens färdighetskunskaper är ofullständiga kan aktivering av deklarativ förståelsekunskap hjälpa honom eller henne att dra korrekta slutsatser om hur den saknade kunskapen kan kompletteras. Det har t.ex visat sig att användare med god förståelsekunskap av systemet har bättre möjligheter än användare med sämre förståelsekunskap att lösa uppgifter som är något annorlunda än dem de tidigare löst. När användaren av någon orsak ger datorn felaktiga instruktioner kan han eller hon med hjälp av sina förståelsekunskaper komma fram till vilket felet är och hjälpa användaren att hitta tillbaka till den plats i programmet där felet gjordes.²⁵

2.2.2.4 En sammanfattning av människans minne

Det är lätt att konstatera att människans minnesfunktion är mycket komplicerad. Minnet har både enorm kapacitet och snäva begränsningar vid informationshantering. Genom att ta hänsyn till hjärnans kapacitet och begränsningar har vi större möjlighet att designa ett gränssnitt som belastar minnet mindre. Trots att långtidsminnet kan lagra nästan oändligt mycket information hindrar det sensoriska minnet och korttidsminnet hjärnan från att ta emot för mycket information på en gång. En skärmbild bör därför inte innehålla för mycket information. En användare bör inte heller tvingas att komma ihåg

²³ Allwood, C M. (1998). *Människa-datorinteraktion – ett psykologiskt perspektiv*

²⁴ Lif, M. (1998). *Adding Usability*

²⁵ Allwood, C M. (1998). *Människa-datorinteraktion – ett psykologiskt perspektiv*

information från en skärmbild som sedan ska användas på en annan. Korttidsminnets kapacitet att ta emot information ökar om informationen är organiserad på något sätt t.ex enligt gestaltarlagarna. Likaså blir information lättare för långtidsminnet att minnas om den är strukturerad och exempelvis stödjer användarens mentala modeller.

3. Användbarhet

Jacob Nielsen har sagt att termen användarvänlig introducerades när informationssystemutvecklare först insåg att deras system skulle skötas av användare med krav på produkten såsom tillgänglighet m.m.²⁶ Nielsen menar att termen inte är passande av flera anledningar. En anledning är att användare inte behöver maskiner för att de ska vara vänliga mot dem, de behöver maskiner som underlättar istället för att hindra dem i arbetet. En annan anledning är att olika användare har olika behov. En användare kan tycka att en maskin är mycket vänlig medan en annan användare kan tycka att samma maskin är mycket otrevlig. Nielsen föreslår istället termen användbarhet²⁷ som relaterar till om systemet kan användas för att uppnå vissa önskvärda mål. Nielsen delar in användbarhet i följande två kategorier:²⁸

- *Funktionell nytta*, behandlar den funktionalitet som måste finnas i systemet. Att testa funktionalitet innebär att man försäkras sig om att produkten fungerar enligt specifikationen. Om systemet inte löser huvudproblemet så spelar det ingen roll om det är lätt att använda, det kommer att bli ett dåligt system i vilket fall som helst.
- *Användbar*, är närmare relaterat till hur väl en användare kan använda funktionaliteten. Användbarhet hänvisar till hur människor arbetar med produkten. Kan användaren använda systemet och kan användaren använda det på ett effektivt sätt? Även om systemet utför exakt rätt saker i teorin, kommer det fortfarande att vara ett dåligt system om användaren inte kan komma på hur han/hon ska få det att fungera. Att testa användbarhet innebär att man försäkras sig om att människor kan hitta och använda de funktioner som de behöver. Användbarhet är måttet på kvaliteten hos användarens upplevelse när användaren interagerar med något, det kan vara en webbsajt, en traditionell mjukvaruapplikation eller vilken annan terminal som helst som användaren kan använda på något sätt.

Att funktionaliteten är riktig är kritiskt, men inte tillräckligt för att en produkt ska vara framgångsrik. En produkt i sig har inget värde utan får ett värde endast när den används. Användning kräver användare och därför är användarnas sätt att arbeta med produkten den grundläggande frågan för produktdesigners och utvecklare.

Användbarhet innebär att människor som använder produkten kan göra det för att snabbt och enkelt utföra sina egna uppgifter.²⁹ Denna definition vilar på följande punkter:

- *Användbarhet innebär fokusering på användaren*. För att utveckla en användbar produkt måste man känna, förstå och arbeta med människor som representerar verkliga eller potentiella användare av produkten.
- *Människor använder produkter för att bli produktiva*. Människor värderar hur lätt det är att lära sig och använda en produkt efter hur lång tid det tar att göra det som

²⁶ Lif, M. (1998). *Adding Usability*

²⁷ engelska: usefulness

²⁸ Nielsen, J. (1998). *What is "Usability"?*

²⁹ Dumas, J S., Redish, J C. (1994). *A practical guide to usability testing*

användarna vill göra, antalet steg de måste gå igenom och framgången de har med att förutse vilken handling som är den rätta att utföra. För att kunna utveckla användbara produkter måste man därför förstå användarnas mål och som utvecklare måste man ha kunskap om användarens arbete och de uppgifter som produkten ska automatisera eller förändra osv.

- *Användare är upptagna människor som försöker utföra olika uppgifter.* Människor förknippar användbarhet med produktivitet och hårdvara och mjukvara är verktyg för att hjälpa upptagna människor att utföra sitt arbete.
- *Användare bestämmer när en produkt är lätt att använda.* Det är användarna och inte designers eller utvecklare som avgör när en produkt är lätt att använda. Vi är alla så upptagna att vi hela tiden väger tiden och kraften vi anser att någonting är värt mot fördelarna som vi tror vi kan uppnå.

3.1 Användbarhet på webben³⁰

Användbarheten hos en webbplats beror på vad användarna försöker att uppnå. Surfar de? Forskar de? Köper de produkter? Laddar de ner mjukvara? Användbarheten beror också på organisationens mål med att skapa webbplatsen. Är den menad att marknadsföra en tjänst? Sälja varor? Göra information tillgänglig för anställda, kunder osv.

Vad målet än är, så är information ett centralt tema. Hur framgångsrik en webbplats är beror på hur väl den kan ge människor den information de behöver. Ju bättre en webbplats hjälper folk att hitta den information de söker, desto mer användbar är den.

³⁰ Spool, et al. (1999). *Web site usability - a designers guide*

4. Utformning av användargränssnitt

4.1 Användargränssnitt

Ett gränssnitt kan i allmänhet beskrivas som den mekanism som möjliggör förbindelse eller kommunikation mellan två enheter. När den ena kommunicerande enheten är en människa, en användare, och den andra en apparat talar vi om användargränssnitt. Telefonens knappar och bilens ratt och pedaler är exempel på användargränssnitt.³¹

Ett datasystems användargränssnitt kan beskrivas som systemets utseende och beteende och är den delen av datasystemet som användaren kommer i kontakt med.³² I början av datorns historia fanns det inget behov av och inga möjligheter att lägga ned tid och kraft på att designa användargränssnitt. Antalet designalternativ var mycket begränsat och det fanns t.ex inga möjligheter att använda grafik och ljud i användargränssnitten. Användare och programmerare var vanligtvis samma person och det var ingen idé att göra system som var användbara för andra människor.

Idag används de flesta datasystem av användare med varierande erfarenhet av att använda datorer. Grafiska gränssnitt och olika interaktionstekniker har ökat möjligheterna för att skapa informationssystem som är lätta att lära sig och effektiva att använda. För att användargränssnitten inte ska skapa problem är det viktigt att de är väl utformade och möter de krav som en användare kan ha när han eller hon arbetar i en viss situation. Användargränssnittet har mycket stor betydelse för huruvida ett informationssystem kommer att accepteras och uppskattas av sina användare.³³ Ett dåligt system kan inte gömmas bakom ett bra användargränssnitt men ett bra system kan förstöras av ett dåligt användargränssnitt.

Samspelet mellan människa och dator påverkas av de egenskaper som respektive användare har. Eftersom det är användaren som med hjälp av dataprogrammet ska utföra en viss arbetsuppgift, bör användaren utgöra det grundläggande målet och dataprogrammet utformas med utgångspunkt i hur människan tänker. För de flesta interaktiva system är displayen huvudkomponenten för en lyckad design och en källa till många argument. Täta eller röriga displayer kan framkalla ilska, och inkonsekventa format kan hämma utförandet.

Användargränssnitt ska vara utformade för att passa människan, så att arbetsuppgifter underlättas och blir korrekt utförda samt över huvudtaget blir möjliga att utföra. Det är viktigt att designen är pedagogiskt genomtänkt och väl formulerad så att hanteringen blir enkel för användaren. Det spelar ingen roll hur bra funktionaliteten i ett system är om användaren av någon anledning upplever att gränssnittet är komplicerat.³⁴

Användarna skiljer sig åt vad gäller kunskap och preferenser; nybörjare behöver enkla, logiskt organiserade och välformulerade displayer som guidar deras handlingar.

³¹ Peeter, K., Wingstedt, U. (1992). *Standarder för grafiska användargränssnitt*

³² Lif, M. (1998). *Adding Usability*

³³ Sundgren, B. (1992). *Databasorienterad systemutveckling*

³⁴ Shneiderman, B. (1997). *Designing the User Interface*

Expertanvändare behöver inte vägledas av lika utförliga texter, t.ex. rubriker på fält, utan det räcker att markera och placera på ett tydligt sätt.³⁵ Användare med normala kunskaper, dvs vardagliga användare, befinner sig på en nivå mitt emellan nybörjare och experter. Dessa personer har till skillnad från nybörjare använt systemet förut och kan på så sätt utnyttja sina kunskaper. Att utforma ett gränssnitt som användarna lätt kommer ihåg kan bespara användarna mycket tid och energi. Ett problem med dagens gränssnitt är att de ofta är uppbyggda enligt principen att så många objekt som möjligt skall vara synliga. Användarna behöver inte anstränga sig för att skapa en bild över funktionerna då allt redan finns presenterat. Studier som har gjorts visar att användare av dessa grafiska gränssnitt har svårt att komma ihåg innehållet under olika menyer när de inte var i kontakt med systemet, men att de utan problem kunde hitta rätt när de arbetade med systemet.³⁶

4.2 Allmänt om principer och riktlinjer

Det finns en rad principer och riktlinjer som man kan följa vid utformningen av gränssnitt för att genom dessa öka gränssnittets användbarhet. Skillnaden mellan principer och riktlinjer kan förklaras genom följande: En princip är ett mål, men man talar inte om hur detta mål ska nås. En princip kan leda fram till flera riktlinjer som i sin tur är mer specifika mål.³⁷

4.3 Åtta principer för gränssnittsdesign³⁸

Ben Shneiderman har presenterat åtta tumregler för gränssnittsdesign och de innehåller principer som kan användas som utgångspunkt vid utformning av användbara gränssnitt för de flesta typer av interaktiva system. Jacob Nielsen³⁹ har också kommit fram till en rad principer som i stora drag överensstämmer med Shneidermans resonemang. Här följer en genomgång av hur författarna anser att man kan uppnå användbarhet. Dessa grundläggande principer för gränssnittsdesign bör anpassas, förfinas och utökas för att passa varje enskilt fall.

1. Sträva efter konsekvens

Denna regel är till viss del svår att följa eftersom det finns många former av konsekvens. Konsekvens är en av de mest grundläggande principerna och syftar till att användaren inte ska behöva fundera på om olika ord, situationer, eller kommandon betyder samma sak. Samma information ska konsekvent presenteras på samma ställe på alla skärmbilder. Det är viktigt att informationen kommer i rätt ordningsföljd; att identisk terminologi används i prompten, menyer, och hjälpfönster; och att man är konsekvent vad gäller färger, layout, storlek, format, förkortningar, osv.

Utformningen bör följa allmänt vedertagna standarder för att formatet ska vara bekant för användaren. Dessa regler gäller bl.a vänsterjustering för alfanumeriska data för att underlätta snabb överblick, högerjustering för numeriska data, lagom luft

³⁵ Shneiderman, B. (1997). *Designing the User Interface*

³⁶ Alm, M. (1992). *Systemutveckling*

³⁷ Dumas, J S., Redish, J C. (1994). *A practical guide to usability testing*

³⁸ Shneiderman, B. (1997). *Designing the User Interface*

³⁹ Nielsen, J. (1993). *Usability Engineering*

mellan saker och ting, användning av logiska rubriker och antal decimalsiffror. Anta en logisk princip att ordna punkter efter. Där inga andra principer passar bör listan ordnas i alfabetisk ordning. Allt detta är viktigt för att gränssnittet ska uppfattas som enhetligt och därför bör man utarbeta en viss typ av standard som ska användas genomgående.

2. Gör det möjligt för vana användare att använda kortkommandon.

Användaren blir efter ett tag säkrare och mer erfaren och vill då kunna använda sig av kortkommandon, för att därigenom minska antalet steg i varje interaktion, öka takten på interaktionerna och få snabbare svarstider. Förkortningar, funktionstangenter, dolda kommandon och möjligheter till makron uppskattas speciellt av kunniga användare som ofta använder sig av applikationen.

3. Ge informativ feedback.

Systemet bör ge feedback på alla handlingar som användaren utför. Denna feedback bör inte bara ske efter att en felsituation har uppstått, utan också snabbt och i anknytning till utförd handling. För handlingar som utförs ofta och för handlingar som är av mindre betydelse bör graden av feedback vara låg, men för handlingar som utförs mer oregelbundet bör feedbacken vara mer påtaglig. Visuellt representation av objekt ger en lämplig miljö för att explicit visa ändringar.

Feedback bör även ges om en process kommer att exekveras under en längre tid. Det är en bra idé att använda två markörsymboler om svarstiderna varierar mycket. Om processen beräknas ta mindre än fem sekunder räcker det till exempel med att ändra muspekaren till ett timglas. Annars är det bra att visa hur lång tid det är kvar av processen, men allra helst anges återstående tid med hjälp av en förloppsindikator.

4. Utforma dialogen på ett sätt som tydliggör avgränsning

För att öka användarens förståelse i ett gränssnitt bör skärmlayouten bygga på gestaltteorins syn på mänsklig perception. Om man grupperar enheter och likartade element som tenderar att höra ihop, hjälper man användaren att förstå strukturen i gränssnittet. Dialoger bör delas in i sekvenser och varje moment i dialogen bör organiseras i grupper med en början, en mitt och ett slut. Genom att framhäva relevanta element kan man göra det enklare för användaren att prioritera de olika valmöjligheterna. Den informativa feedbacken som ges när man slutfört en sekvens ger användaren tillfredställelsen att ha gjort rätt och ger en indikation att det är dags att göra sig redo för nästa sekvens av handlingar.

Det är bra att inleda varje skärmbild med en titel eller rubrik och att kort beskriva skärmbildens sammanhang eller syfte. Lämna åtminstone en tom rad mellan titeln och resten av texten på displayen. Bra designade displayer bör visa all nödvändig data för att slutföra en viss uppgift vid ett visst tillfälle. Konsekvent och meningsfull gruppering av punkter och med beteckningar som är lämpliga med avseende på användarens kunskap är också viktigt. Grupper kan omges av blankrader eller av boxar, alternativt kan relaterade punkter indikeras med hjälp av bakgrundsskuggning, färg, eller speciell storlek. Inom grupper kan ordning åstadkommas genom vänster- eller högerjustering och placering i rak linje.

5. Förhindra fel och erbjud enkel felhantering.

Så långt det är möjligt bör systemet designas så att det är omöjligt för användaren att göra allvarliga fel. Det finns olika metoder för att förhindra att användaren hamnar i situationer där fel kan uppstå. Exempelvis finns det risk för felstavning så fort användaren ska mata in text. Denna typ av fel kan undvikas genom att istället för att använda blankettifyllning, placera alla alternativ i rullgardinsmenyer eller radioknappar, där användaren istället tvingas välja ett alternativ från en lista. Fel kan även undvikas genom att exempelvis inte tillåta bokstäver i numeriska fält och genom att be användaren bekräfta sitt val och på så sätt ge användaren en andra chans.

Om användaren gör ett fel bör systemet upptäcka felet och erbjuda enkel och konstruktiv hjälp. Användaren ska inte behöva göra om hela förloppet utan bara den del som orsakade felet. Felaktiga handlingar bör inte ändra något i systemet, eller så bör systemet ge instruktioner om hur användaren kan återskapa tillståndet.

Felmeddelanden bör uttryckas i vardagligt språk, konsekvent uppträda på samma plats, använda konsekvent terminologi och syntax, indikera vilket problemet är samt ge förslag på lösning. Bra felmeddelanden hjälper användaren att förstå systemet bättre eftersom han eller hon då motiveras att begrunda innehållet i meddelandet. Använd bekräftande, dvs positiva uttryck, hellre än negativa uttryck.

Felmeddelanden bör följa fyra grundregler:⁴⁰

- Ska vara skrivna på användarens språk och undvika koder.
- Ska precisera vad som är fel, inte bara tala om att något är fel.
- Ska konstruktivt komma med förslag på hur problemet ska lösas.
- Ska vara trevligt utformade och inte förnedra användaren vid fel.

6. Tillåt användaren att ångra handlingar.

I så stor utsträckning som möjligt bör utförda handlingar gå att ångra. Denna företeelse minskar oro eftersom användaren vet vilka fel som går att ångra. Det gör att användaren känner sig säkrare och uppmuntrar användaren till att utforska icke bekanta val eftersom denne vet att fel kan åtgärdas.

7. Användarkontroll.

Användare vill inte bli låsta av ett datasystem utan vill ha känslan av att de bestämmer över systemet och att systemet svarar på deras handlingar. Överraskande systemhandlingar, svårighet att få tag i nödvändig information och oförmåga att utföra den efterfrågande handlingen bidrar till oro och otillfredsställelse för användaren. Det händer ofta att fel funktioner väljs, vilket kräver tydligt markerade utvägar och lättillgängliga ångerfunktioner. En grundläggande princip för design av användargränssnitt är att anta att användaren, oavsett hur mycket gränssnittet förbättras, fortsätter att göra fel. Därför ska det vara enkelt för användaren att korrigera felen. För att inte belasta användarna bör utgångar och ångerfunktioner vara klart synliga i gränssnittet.

⁴⁰ Nielsen, J. (1993). *Usability Engineering*

8. Minimalt krav på användarens minne

Människans begränsade förmåga att behandla information i korttidsminnet kräver att displayen hålls enkel och inte belastar användarens minne i onödan. Det är viktigt att göra objekt synliga så att användaren inte behöver anstränga sig för att komma ihåg information från ett steg till ett annat. För att öka användbarheten bör systemet baseras på ett antal lättförståeliga standardregler som kan appliceras på hela gränssnittet. På så sätt känner användaren i de flesta fall igen sig i olika situationer och vet därmed hur han eller hon ska agera i en viss situation. Igenkännande är bättre än ihågkomst. Ju fler regler som behövs för att förstå hur systemet uppträder och dess struktur, desto svårare blir det att komma ihåg. Men om det inte finns några regler alls måste användaren lära sig alla dialoger och element, vilket är nästan helt omöjligt.

4.4 Riktlinjer för inmatning av data⁴¹

Inmatning av data kan uppta en väsentlig del av en användares tid och är ofta en källa till frustration och till eventuella farliga fel. Smith och Mosier ger fem viktiga mål för datainmatning:

1. *Konsekvens vid datainmatning.* Liknande sekvenser av handlingar bör användas under alla tillstånd; lika avgränsningar, förkortningar, osv bör användas.
2. *Minimal inmatning.* Så lite inmatning av text som möjligt innebär större produktivitet för användaren och att färre antal fel uppstår. Det är bättre att göra ett val genom en enkel knapptryckning, än att skriva en lång sträng av tecken. Att välja ett alternativ från en lista eliminerar behovet av memorering, strukturerar frågan som ligger till grund för beslutet och minskar möjligheten för typografiska fel. En annan aspekt angående denna riktlinje är att redundant datainmatning bör undvikas. Det är irriterande för användare att mata in samma information på två ställen, eftersom detta dubbelarbete uppfattas som slöseri med resurser och ökar risken att fel uppstår. När samma information efterfrågas på två ställen, bör systemet automatiskt kopiera informationen från det första stället till det andra. Användaren bör fortfarande ha möjligheten att ändra den automatiskt inkopierade informationen genom att skriva in nya data.
3. *Minimala krav på användarens minne.* När data matas in bör användaren inte behöva komma ihåg långa koder eller komplexa syntaktiska kommandosträngar.
4. *Likhet mellan inmatning och presentation av data.* Vid inmatning av information bör den visuellt visas på ett sätt som överensstämmer med hur informationen sedan presenteras på skärmen.
5. *Flexibel utformning.* Erfarna användare kan ibland föredra att inmatningsfält kommer i en viss ordning som de själva kan kontrollera. Till exempel kan det i en viss situation vara en fördel för användaren om ett visst fält visas överst på skärmbilden, för samma användare i en annan situation kan det vara bättre om ett

⁴¹ Shneiderman, B. (1997). *Designing the User Interface*

annat fält visas överst. Den här typen av flexibilitet bör användas med försiktighet, eftersom det strider mot principen för konsekvent utformning av användargränssnitt.

4.5 Utformning av menyer⁴²

Experimentella undersökningar har gjorts inom området meny-layout. Denna sektion innehåller många subjektiva omdömen, vilka är i behov av empirisk validering.

4.5.1 Rubrik

Menyer som har beskrivande rubriker är lätta för användaren att komma ihåg. Att välja en bra rubrik är komplext och för det krävs eftertanke. För enkla menyer är en enkel beskrivande titel som identifierar situationen allt som behövs. I en linjär sekvens av menyer bör rubrikerna tydligt representera stegen i den linjära sekvensen. Konsekvent grammatisk stil kan minska förvirring och kortfattade, men entydiga fraser är ofta tillräckligt. Det är även viktigt att tänka på att vara konsekvent vad gäller placeringen av rubriker.

För menyer som är utformade enligt en trädstruktur är det svårare att välja rubriker. Att använda rubriker som "Huvudmeny" eller ämnesbeskrivning som "Banktransaktioner" i roten av menyträdet indikerar tydligt att användaren befinner sig högst upp i menyn. En bra regel är att använda exakt samma ord i rubriker på högre nivåer som i rubrikerna för de sidor som menyerna leder till.

4.5.2 Ordval

Följande riktlinjer kanske verkar självklara men är inte alltid lätta att följa.

- *Använd bekant och konsekvent terminologi.* Den terminologi som ska användas bör väljas med omsorg och bör vara bekant för den tilltänkta användargruppen. Under utformning av gränssnittet kan det vara en hjälp att ha en lista över de termer som ska användas för att på så sätt försäkra sig om att termerna används på ett konsekvent sätt.
- *Se till att namn särskiljer sig från varandra.* Varje namn bör kunna skiljas klart och tydligt från andra namn.
- *Använd konsekvent och kortfattad namngivning.* Man bör gå igenom alla ord för att försäkra sig om att namngivningen är konsekvent. Användare lyckas bättre med namn som Djur, Grönsaker och Mineraler än med Information om djur, Välj mellan olika grönsaker, osv.
- *Placera nyckelordet till vänster.* Det är bra att försöka skriva menynamnet så att det första ordet hjälper användaren att känna igen och skilja de olika punkterna åt. Det är t.ex bättre att använda "Sidnummer" istället för "Sätt in sidnummer". Användare skummar menyinnehållet från vänster till höger. Om det första ordet indikerar att menyvalet inte är relevant finns det risk att de börja granska nästa val.

⁴² Shneiderman, B. (1997). *Designing the User Interface*

4.5.3 Menyers utseende

En konsekvent utformad meny blir förutsägbar och hjälper användaren att hitta nödvändig information, fokuserar användarens uppmärksamhet på relevant material och minskar användarens osäkerhet. Dessutom, eftersom desorientering i program är ett möjligt problem, kan tekniker som indikerar position i menystrukturen vara användbara. I böcker används olika storlekar och teckensnitt för att indikera kapitel, sektion och underavdelning. När linjära menyer används kan användaren få en enkel visuell presentation av positionen i sekvensen med hjälp av en positionsmarkör. I en sekvens av exempelvis 12 skärmbilder, kan man med en positionsmarkör (+) visa hur långt i sekvensen användaren har kommit. På den första skärmbilden befinner sig positionsmarkören

+-----

på den andra sidan

-+-----

och på den sista sidan

-----+

Med hjälp av positionsmarkören kan användaren se var han eller hon befinner sig och hur mycket som återstår.

4.6 Att fylla i formulär

Menyval är effektivt för att välja något från en lista, men vissa situationer blir ohanterliga med menyer. Om det till exempel krävs inmatning av personnamn eller numeriska värden är det bättre att använda tangentbordet. När mycket data ska matas in är den mest lämpliga interaktionsformen formulär. Formulär är bra därför att den kan visa mycket information på en gång och ger användaren en känsla av att de har kontroll över dialogen. Det krävs få instruktioner för att användaren ska förstå vad han eller hon ska göra eftersom dialogen på skärmen påminner om pappersblanketter.

4.6.1 Riktlinjer vid design av formulär⁴³

- *Meningsfull titel.* Identifiera ämnet och undvik dataterminologi.
- *Innehållsrika instruktioner.* Beskriv vad användaren ska göra i bekant terminologi. Var kortfattad och om mer information behövs, gör ett hjälpfönster tillgängligt för den ovane användaren. För att kunna vara kortfattad, beskriv bara den nödvändiga handlingen (t.ex ”Skriv adressen” eller bara ”Adress”) och undvik pronomen (”Du ska skriva adressen”) eller referenser till användaren (”Användaren ska skriva adressen”).

⁴³ Shneiderman, B. (1997). *Designing the User Interface*

- *Logisk gruppering.* Fält som hör ihop bör ligga nära varandra. Fälten bör separeras med blankrad för att separera grupper av fält. Fältens sekvens bör följa vanligt mönster, t.ex att gatuadress följs av postnummer och ort.
- *Visuellt tilltalande layout.* Man bör inte trycka ihop mycket information på en viss del av skärmen, utan försöka att fördela grupperna av fält så att det inte blir några stora tomma ytor på skärmen.
- *Bekanta namn på fält.* Allmänt vedertagna ord bör användas.
- *Använd konsekvent terminologi och förkortningar.* Detta kan t.ex uppnås genom att man gör en lista med ord och acceptabla förkortningar som genomgående ska användas.
- *Synligt utrymme och avgränsning för datainmatningsfält.* Användare bör kunna se längden på fältet för att bl.a kunna förutse om kortformer kommer att behövas.
- *Enhetlig förflyttning av markör.* Använd en enkel och synlig mekanism för att flytta markören, som t.ex TAB-tangenten eller piltangenterna.
- *Felkorrigering för enskilda tecken och för hela fält.* Gör det möjligt för användaren att göra enkla korrigeringar eller att ändra hela fält.
- *Förhindra fel.* Där det är möjligt, hindra användare från att mata in värden som inte är korrekta. T.ex i ett fält som efterfrågar ett positivt heltal, tillåt inte användaren att mata in bokstäver, minustecken eller decimaltecken.
- *Felmeddelanden för oacceptabla värden.* Om användare matar in oacceptabla värden bör felmeddelanden komma fram när inmatningen i fältet är klar. Meddelandet bör indikera tillåtna värden för fältet.
- *Markera fält som är valfria.* Fält som inte är obligatoriska för användaren att fylla i bör vara tydligt markerade.
- *Förklarande meddelanden för fält.* Om det är möjligt, bör förklarande information om ett fält eller tillåtna värden finnas på en standardposition som t.ex i ett fönster längst ner, när markören är i det fältet.
- *Signal som visar slutförande stödjer användarkontroll.* Det bör tydligt framgå vad användarna ska göra när de har fyllt i alla fälten. Allmänt bör designers undvika automatiskt slutförande när det slutliga fältet är ifyllt, eftersom användare kanske önskar titta på eller ändra tidigare fält.

5. Utformning av användargränssnitt på webben

5.1 Skillnad mellan traditionella gränssnitt och webbgränssnitt

Att designa för webben skiljer sig från design av traditionella användargränssnitt. Framförallt måste designern lämna ifrån sig kontrollen och dela ansvaret för gränssnittet med användarna och deras hård- och mjukvara.⁴⁴

Självklart finns det även likheter mellan att designa för webben och designa traditionella gränssnitt. Det mest grundläggande är att de båda är interaktiva system och båda är mjukvarudesign i motsats till design av fysiska objekt.

I traditionell gränssnittsdesign kontrollerar man varje pixel på skärmen. Man vet vilket system man designar för, vilka typsnitt som finns installerade samt hur stor skärmen i de flesta fall är. På webben fungerar detta inte på samma sätt. Användare kan använda webben via traditionella datorer, men även via en webb-tv, en handdator, en mobiltelefon osv. I traditionell design är skillnaden i skärmyta mellan en laptop och en arbetsstation faktor 6. På webben är det faktor 100 i skillnad på skärmstorlek mellan handdatorer och arbetsstationer och faktor 1000 i skillnad i bandbredd.

5.2 Riktlinjer för webbgränssnitt

De principer och riktlinjer som gäller för traditionella användargränssnitt gäller också för webbgränssnitt, men förutom dessa finns det även andra riktlinjer som är användbara när man designar och utvärderar ett webbgränssnitt.⁴⁵

- *Användaren har kontroll.* Eftersom det förekommer många olika webbläsare som tolkar HTML-koden (Hypertext Markup Language) på olika sätt, kan inte utvecklare vara säkra på vem eller vad som kommer att processa sidorna. Ju mer generell designen är, desto troligare är det att det går att uppnå en tillfredsställande användbarhetsnivå hos ett stort antal användare och över en lång tidsperiod.
- *Struktur.* I alla hypermediasystem är det viktigt att ha en grundläggande struktur för att hjälpa användarna att veta var de är och vart de kan gå härnäst. På webben där sökmotorer och länkar kan leda in användarna mitt i en sajt är det mycket viktigt med denna struktur.
- *Designa för förändring.* Webben förändras hela tiden och därför bör användargränssnittet kunna förändras både innehållsmässigt och designmässigt.

5.3 Kompromisser om användbarhet

Användbar gränssnittsdesign kan innebära kompromisser, att väga kostnader mot fördelar för att utveckla den bästa lösningen under rådande omständigheter. Webben

⁴⁴ Nielsen, J. (1997). *The difference between Web design and GUI design*

⁴⁵ Instone, K. (1997). *Usability engineering for the web*

kräver kompromisser som måste övervägas noggrant på grund av webbens låga bandbredd, webbläsarnas roll och eftersom det finns standarder för att skapa webbgränssnitt. Det är viktigt att förstå fördelarna och nackdelarna för att kunna fatta ett riktigt beslut.

- *Bandbredd.* Oavsett hur bra en design är så kommer den inte att synas om det tar för lång tid att ladda ner sidan. Användare tolererar inte långa fördröjningar. Om den stora allmänheten använder sig av modem för att se webbsajten bör man undvika att använda stora bilder på sidorna eftersom många inte är tålmodiga nog att vänta på att bilderna ska laddas ner.
- *Webbläsare.* En annan kompromiss inträffar på grund av webbläsarens avgörande roll. Webbläsaren erbjuder funktioner som ligger utanför webbdesignerns kontroll, t.ex senast besökta sida och bokmärken. Utvecklare måste lita på att webbläsaren ger viss funktionalitet, vilket kan begränsa skapandet av användbara sajter.
- *Standarder.* Den viktigaste standarden för utvecklare är HTML. Att känna till HTML-standarderna och följa den är mycket viktigt för att kunna fatta riktiga design- och implementeringsbeslut. Generellt gäller att ju mer noggrant standarden följs, desto mer användbar blir en sajt för den största publiken. Designers av intranät har det lite lättare än de som har en Internetpublik. Om man vet om att alla användare använder en speciell webbläsare och en viss version kan man säkrare utnyttja funktioner som inte är standard.

5.4 Navigering

Struktur och navigering är mycket viktigt på webben, mycket viktigare än på andra plattformar. Bland det viktigaste med navigering på webben är att demonstrera att applikationen och webbläsaren inte hör ihop. Tillbakaknappen som finns i alla webbläsare är t.ex inte en del av applikationen. Om användaren klickar på tillbakaknappen tre gånger vet inte applikationen om detta. Även om det kan se ut som om tillbaka är en ångra-funktion så innebär den endast att man får sidorna i omvänd ordning. På liknande sätt är inte framåtknappen ett "gör om"-kommando. Användare av webben, framförallt nybörjare kan på grund av detta ha svårt att se var webbläsaren slutar och applikationen börjar.

Det är lätt för användare att gå vilse i webbapplikationer eftersom de har otydligare struktur än andra applikationer. På webben kan designers skapa sidor av vilken längd som helst och med ett obegränsat antal länkar till andra sidor. Om användare gräver ner sig i information eller hamnar på ett sidospår kan de tycka att det är svårt att komma tillbaka till det som de höll på med. Navigeringsmodellen på webben skiljer sig från navigering i traditionella applikationer och det händer ofta att användare av webbapplikationer går vilse. Denna skillnad kan orsaka förvirring bland användare, men det finns ett par tekniker som applikationsutvecklare kan använda för att underlätta för användarna så att de kan använda applikationen på ett bra sätt.⁴⁶

⁴⁶ Shubin, H. (1997) *Interaction Design, Navigation in Web Applications*

Det viktigaste att tänka på är hur informationen ska arrangeras. På en informationssajt där surfande är huvudaktiviteten kan det vara en fördel att länka alla sidor på sajten till varandra eller till och med erbjuda länkar till relevant information på andra sajter. I en applikation bör man hjälpa användaren att antingen slutföra en uppgift genom att minimera distraherande länkar eller erbjuda ett tydligt val så att användaren kan avsluta uppgiften.

Webbsidedesignen hjälper också användare att hålla reda på var de är. Tydliga, konsekventa ikoner, bilder och textbaserade överblickar kan hjälpa användaren att hitta vad han eller hon söker utan att slösa bort för mycket tid.⁴⁷ Andra navigeringshjälpmedel som sidnamn, logotyper, banners, bakgrundsfärger osv kan också användas som visuell ledtråd till användarna. Man bör arrangera val i en tydlig struktur, t.ex i en hierarki så att användare kan förstå var de är och vart de kan gå, genom att snabbt titta på strukturen. Användare bör alltid kunna återkomma till huvudsidan och andra stora navigeringssidor i en sajt. Dessa länkar bör alltså finnas på alla sidor.

Användare vill kunna få information genom att gå igenom så få steg som möjligt vilket innebär att man bör designa en effektiv informationshierarki för att minimera antalet steg genom menysidor. Studier har visat att användare föredrar menyer som visar minst fem till sju länkar och att de föredrar ett fåtal sidor med många alternativ framför flera lager med förenklade menyer.

Ett bra sätt att strukturera en viktig uppgift är att endast ha relevanta kommandon på sidan och kräva att användarna antingen slutför uppgiften eller lämnar sidan innan de gör något annat. Det finns inte något annat sätt att avsluta en webbapplikation förutom genom att stänga webbläsaren och när man stänger webbläsaren finns ingen möjlighet att få frågan om man vill spara det man arbetat med. Vanligtvis lämnar webbanvändare en applikation genom att klicka på en länk till en annan sida eller sajt. Att se en tydligt definierad avsluta-länk kan hjälpa användarna att komma ihåg att avsluta sitt arbete innan de lämnar applikationen.

5.5 Sammanfattning av riktlinjer för webbdesign

Några enkla riktlinjer för att förhindra att användare lämnar webbsidan och att användarna förlorar sitt arbete genom att de lämnar webbsidan innan de har slutfört en operation:

- Ha en särskild avsluta-sida som visar externa länkar och annan användbar information.
- Placera länkar på säkra ställen så att användare inte förlorar arbete om de klickar vidare och lämnar applikationen. Påminn dem om att sätta ett bokmärke på sidan och återkomma.
- Använd ramar när användarna klickar på externa länkar. En ram visar målsidan och en annan ram visar en påminnelse om att återkomma till ursprungssidan.

⁴⁷ Lynch, P J., Horton, S. (1999). *Web Style Guide, Basic Design Principles for Creating Web Sites*

- Arrangera information noggrant så att användarna alltid vet var de är i applikationen. Det är lätt att tappa bort sig och ofta lämnar användarna en webbplats där de inte har kontroll över var de är.
- Fresta inte användarna att avbryta sitt arbete genom att följa länkar till andra sidor.
- Göm "gränsen" mellan webbläsaren och applikationen genom att lägga in navigeringskontroller i applikationen.
- Minimera nätverksfördröjningar genom tydlig navigering och reducerad nedladdningstid. I ca åtta sekunder har användarna tålamod att vänta på att sidorna ska laddas ner, sedan är det risk att de lämnar webbplatsen.⁴⁸
- Gör webbplatsen lätt för användarna att lära sig. En bra webbplats ska man inte behöva ha manual för att lära sig använda. Designa webbplatsen så att besökarna kan "gissa rätt" varje gång.

48 Story, D. (1999). *Usability Checklist for Site Developers*

6. Utformning av användargränssnitt för mobiltelefoner

Det blir mer och mer vanligt att använda mobiltelefon i vårt samhälle. Från att ha varit ett verktyg för en liten målgrupp, används de idag av nästan alla människor.⁴⁹ Denna breda grupp av användare gör det viktigt att utforma användbara gränssnitt för mobiltelefoner. Vi kommer i detta avsnitt att presentera WAP-tekniken, som gör det möjligt att använda mobila applikationer. Vi kommer även att redogöra för delar av den forskning som har gjorts inom området för utformning av gränssnitt för mobiltelefoner.

6.1 WAP

Mobiltelefonens främsta användningsområde är samtal mellan människor, men det finns även möjlighet att använda sig av mobiltelefonen för att exempelvis kunna utföra enklare bokningsärenden, få trafikinformation, ta del av nyheter och väder m.m. WAP-tekniken gör detta möjligt och innebär i korta drag att användaren får tillgång till Internet i mobiltelefonen.

WAP (Wireless Application Protocol) är en standard utvecklad av WAP Forum. WAP Forum grundades 1997 av Ericsson, Nokia, Phone.com och Motorola med syfte att definiera en standard för hur innehåll på Internet kan filtreras för mobil kommunikation. Standarden för mobilt Internet, WAP, sattes som en öppen standard, vilket i korthet innebär att företag som ansluter sig till WAP Forum förbinder sig att följa de standarder som utarbetas inom organisationen.⁵⁰

WAP gör det möjligt för användare att få tillgång till olika Internetbaserade tjänster via sin mobiltelefon, PDA (Personal Digital Assistant) eller annan terminal som är anpassad för WAP. Ergonomiskt sett kan en mobiltelefon knappast konkurrera med en PC.⁵¹ Telefonens skärm kan endast visa ett fåtal tecken, bandbredden är extremt begränsad och att skriva in text är mycket krångligt och tar lång tid. Men en av de stora fördelarna med WAP är att användarna har tillgång till omedelbar information via WAP-applikationer oberoende av var de befinner sig, bara de har sin mobiltelefon tillgänglig.

WAP är konstruerat för att kräva minimal datatrafik och för att information ska kunna visas på en liten skärm. Därmed passar tekniken mycket bra för att användas i mobiltelefoner.⁵² WAP är designat för att kunna erbjuda värdefulla tjänster genom att lägga intelligensen på WAP-servern och endast ha en microbrowser i mobiltelefonen. Microbrowser-baserade tjänster och applikationer ligger tillfälligt på servrar, inte permanent i telefonerna.

Målet med WAP-tjänster är inte att användaren ska surfa på nätet utan det är att hela tiden kunna ha aktuell information tillgänglig. Detta innebär att tjänsterna måste vara relevanta för användarna och anpassade efter användarna på ett annat sätt än de

⁴⁹ Nokia. (1999). *Wireless Application Protocol – The Corporate Perspective*

⁵⁰ Stedt, P. (1999). *WAP pratar alla om?*

⁵¹ Passani, L. (1999). *WAP and ASP - Part 1*

⁵² Nordlin, B. (1999) *Så här fungerar tekniken*

traditionella tjänster som finns för Internet.⁵³

Tekniken har kritiserats en hel del. Ett skäl kan vara mobiltelefonverkarnas svårigheter att få ut WAP-telefoner på marknaden. Ett annat skäl kan vara att flera av de WAP-tjänster som hittills lanserats helt enkelt visat sig för triviala och ogenomtänkta för att användaren ska se någon nytta med dem. En invändning mot WAP är också att avgifterna för att vara uppkopplad i gsm-näten med mobiltelefonen är så höga att många WAP-tjänster upplevs som alldeles för dyra att använda.⁵⁴

6.1.1 WML⁵⁵

WML (Wireless Markup Language) är det språk som används för att utveckla WAP-tjänster. WML är en bantad version av HTML, men en stor skillnad är att WML måste kompileras innan det kan användas. I WML har man försökt ta hänsyn till den begränsade bandbredden som finns idag, mobiltelefoners begränsade display och de svårigheter som förekommer vid inmatning av text med hjälp av en telefons siffertangenter.

Istället för att använda den dokumentstruktur som HTML tillhandahåller, så har man i WML-strukturen delat upp dokumenten i "decks and cards" dvs i kortlekar och kort. En kortlek består av en mängd kort. Ett kort är en interaktionsenhet som vanligtvis är så liten att den kan visas på en mycket liten skärm. En WAP-tjänst är ofta uppbyggd av en serie kort som användaren navigerar genom. Användningen av kortlekar och kort ger också en navigeringsmodell som inte kräver allt för mycket navigering inom en sida eftersom användaren guidas genom en serie av kort, istället för att behöva scrolla upp och ned på en stor sida.

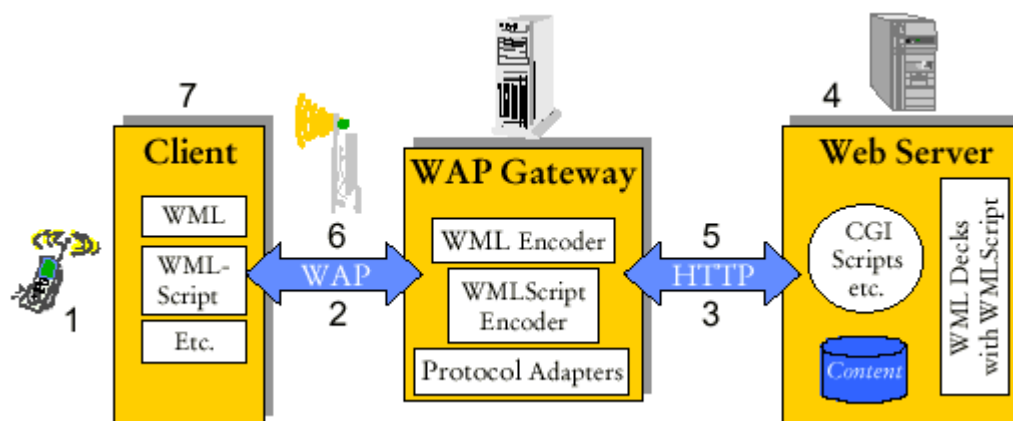
⁵³ Jakobsson, H. (2000). *Wap ger dig ständig uppgraderingsångest*

⁵⁴ Zirn, T. (2000). *Vänta inte på gprs*

⁵⁵ AU-system Radio. (1999). *WAP White Paper*

6.1.2 WAP-strukturen⁵⁶

WAP-applikationer adresseras med URL:er (Uniform Resource Locator) på samma sätt som på Internet. Figur 6.1 ger en beskrivning av strukturen för WAP-kommunikationen.



Figur 6.1:

1. Användaren använder telefonen för att välja en länk som är kopplad till en viss URL.
2. Telefonen skickar en URL-förfrågan till en WAP-gateway genom att använda WAP-protokollet.
3. WAP-gatewayen omvandlar URL:en till en vanlig HTTP-fråga och skickar den till webbservern.
4. Webbservern bearbetar HTTP-frågan. URL:en kan antingen leda till en statisk fil eller till en script-applikation av något slag. Är det en statisk fil så hämtar servern filen och lägger till en HTTP-header. Är det en script-applikation så kör servern applikationen.
5. Webbservern skickar tillbaka WML-decket med HTTP-headern eller WML-outputen från scriptet.
6. WAP-gatewayen verifierar HTTP-headern och WML-innehållet och kodar om det till binärt format. Gatewayen skapar sedan ett WAP-svar som innehåller WML och skickar det till telefonen.
7. Telefonen tar emot WAP-svaret, läser WML-svaret och visar användaren det första kortet som "decket" innehåller.

6.2 Applikationer för mobil användning⁵⁷

För att användarna ska vilja använda en elektronisk tjänst är det viktigt att utforma gränssnittet på ett enkelt sätt. I en mobil miljö är det ännu viktigare att bevara en applikation enkel eftersom användarna ofta måste koncentrera sig på andra saker runt omkring i sin omgivning samtidigt som de använder terminalen. Användaren kan vara

⁵⁶ Nokia. (1999). *Getting Started, Nokia WAP Toolkit*

⁵⁷ Phone.com. (1999). *Application Usability Guidelines from the Application Development Team*; Nokia. (2000). *Service Developer's Guide*; Ericsson. (2000). *Design guidelines for WAP services*

upptagen med utföra något annat eller vara inblandad i en situation som kräver hög visuell uppmärksamhet.⁵⁸

Samtidigt är det viktigt att komma ihåg att mobiltelefonen först och främst är en telefon. De flesta telefonerna idag med inbyggda browsers utformades i huvudsak för att fungera som en telefon och browsern lades till senare. De flesta användare köper sina telefoner för att de är små och har bra ljudkvalitet, och browsern kommer många gånger i andra hand.

När man bestämmer vilken information som bör finnas med i en mobil applikation bör man tänka på den situation som mobiltelefonen kommer att användas i och vilken typ av information som då är relevant. På grund av att mobiltelefoner har små skärmar kommer användaren främst att använda telefonen när han eller hon inte har tillgång till Internet på en PC. Användaren är i en sådan situation främst intresserad av att hämta kortfattad och snabb information. T.ex är snabb access till flygtiderna från mobiltelefonen relevant för en användare som är på väg till flygplatsen och behöver kolla avresetiden för ett visst flyg. Likaså är kortfattad information om dagens nyheter mer relevant än långa artiklar för en användare som är i rörelse. Mobila användare använder sina telefoner för att snabbt få tag på information, därför är det mindre sannolikt att användare kommer att vilja surfa på det traditionella sättet med hjälp av mobiltelefonen. Användare kommer förmodligen att föredra att surfa på Internet via PC:n när de befinner sig på jobbet eller i hemmet.

Alla användargränssnitt bör vara utformade på ett enkelt sätt för att vara lättanvända. Användbarhet är viktigt vad gäller alla terminaler, men är extra viktigt för små terminaler⁵⁹ på grund av att den begränsade skärmstorleken gör det svårare för användaren att överblicka informationen och skapa ett sammanhang. Följande stycken tar upp vad man bör tänka på när man utformar användargränssnitt för mobiltelefoner.

6.2.1 Stora skärmar kontra små skärmar

Mobiltelefonernas små displayer begränsar den mängd information som kan visas på varje skärmbild, jämfört med PC:n som ger mer plats och yta för t.ex bilder och extra information. På en mobiltelefon bör istället speciell uppmärksamhet riktas mot att prioritera informationen som visas på skärmbilden. Därför är det viktigt att man tänker igenom vilken typ av information som kan vara relevant vid mobil användning.

De flesta mobiltelefoner med browsers har små skärmar, ett utrymme som i genomsnitt motsvarar fyra rader med 12-16 tecken på varje rad. Därför bör man inte utforma applikationer som enbart riktar sig till speciella telefoner med större skärmar. Denna typ av telefoner finns ju visserligen, men de allra flesta telefoner har små displayer. Applikationer utvecklade för små displayer tenderar att se bra ut och fungera bra också på något större displayer, men applikationer som är utvecklade för stora skärmar ser inte bra ut och fungerar mycket dåligt på små displayer.

⁵⁸ Kristoffersen, S., Ljungberg, F. (1999). "Making Place" to Make IT Work

⁵⁹ Phone.com. (1999). *Developing User Friendly Applications*

6.2.2 Navigeringsprinciper

Applikationens navigeringsstruktur utgörs av hierarkin bland skärmbilderna. Ju fler undernivåer som läggs till i applikation ju mer komplex blir navigeringen och det finns en risk att användaren tappar bort utgångsläget och upplever att de inte har kontroll över situationen. Huruvida användaren är nöjd med applikationen beror till stor del på om användaren upplever att den är lätt att navigera i. De flesta användare vill känna sig trygga med applikationen och vill känna igen sig när de navigerar mellan de olika skärmbilderna.

De aktiviteter som man antar att användaren kommer att vilja använda mest bör designas så att användaren kommer åt dem med så få tangentryckningar som möjligt. Varje tangentryckning som måste göras minskar applikationens användbarhet eftersom användarna vill utföra så få tangentryckningar som möjligt. Många tangentryckningar innebär ofta att användaren kommer hierarkiskt långt ner i applikationen och ju längre ner i skärmbildshierarkin användaren kommer, ju mer oroar sig användaren för hur han eller hon ska ta sig tillbaka till utgångsläget. Eftersom användaren lätt går vilse i komplexa och djupa menysystem är det bra att sträva efter att hålla en platt hierarki bland skärmbilderna, det vill säga inte ha för många nivåer. En annan allmän navigeringsregel är att användaren lätt ska kunna gå tillbaka i applikationen, man bör till exempel inte kräva att användaren ska trycka på tillbaka-knappen många gånger. I de fall användaren använder sig av tillbaka-knappen bör den leda dit användarna tror att den ska gå och ta användaren till en logisk och konsekvent plats i applikationen, och inte låta användaren gå runt i cirklar på grund av att de inte hittar tillbaka till utgångsläget.

En bra applikation har alltså ett naturligt flöde med en intuitiv navigering. Användarna tänker i termer av det problem som de försöker att lösa, och inte i termer av hur verktyget löser problemet. Verktyget måste vara transparent för användaren. När användaren tvingas stanna upp för att fundera ut hur verktyget fungerar blir de distraherade och frustrerade och detta innebär en risk att de inte kommer på hur de ska komma vidare, eller glömmet bort problemet som de försöker lösa. För att användaren ska känna sig trygg i sin användning av applikationen bör han eller hon även ha möjlighet att ångra kommandon. Detta är speciellt viktigt i situationer då användaren är nyfiken och vill utforska applikationen och han eller hon behöver en lätt väg tillbaka, utan att för den skull förstöra någonting. Handlingar som inte går att ångra måste vara tydliga eller omöjliga att utföra. När användaren ska utföra en ny uppgift som han eller hon inte har utfört tidigare, som t.ex att skriva ett e-mail istället för att läsa ett, bör applikationen vara så intuitiv att användaren själv kommer på hur han eller hon ska utföra uppgiften utan extern hjälp från andra eller från manualer.

En applikation har ofta ett brett användningsområde, alla funktioner är viktiga, men bara 15-20% av funktionerna kommer att användas mest. Ett exempel är en användare som använder sig av mobiltelefonen för att läsa e-mail. Antagligen läser användaren många e-mail innan han eller hon vill svara på något av dem via telefonen. Att förstå den företeelsen eller aktiviteten är nyckeln till användaren och att reducera antalet steg för att utföra en sådan aktivitet kommer att göra att användaren upplever användningen av applikationen som mycket mer angenäm.

6.2.3 Text på små skärmar

Information och text som visas på mobiltelefonens skärm bör hållas kort och meningsfull, eftersom mobiltelefonens skärm ofta bara kan visa fyra rader text åt gången. Om texten på sidan är för lång finns det alltid en risk att användaren tappar bort sig i textmassan när han eller hon scroller i texten. Eftersom överblick är viktigt bör textlängden begränsas till ett ungefärligt maximum av längden av 4-5 skärmbilder, vilket ungefär motsvarar 16-20 rader.⁶⁰ Att använda länkar är ett bra sätt att komma åt mer information. Länkar är kraftfulla verktyg för att länka samman information och förhindra långa sidor med text.

Ordens längd orsakar ibland problem på små skärmar. Därför bör korta och meningsfulla ord väljas framför långa, dvs ord som är längre än 10 tecken. Orden bör vara korta men samtidigt är det bra att försöka undvika förkortningar, eftersom obekanta förkortningar kan vara svåra för användaren att förstå och kan misstolkas.

6.2.4 Rubriker

Rubriker och titlar beskriver skärmbildens innehåll och hjälper användaren att navigera i applikationen eftersom de fungerar som en påminnelse om var användaren befinner sig i applikationen. Rubriktexten bör bestämmas av det val användaren gjort på föregående skärmbild. Till exempel talar rubriken "Bokmärken" om för användaren att skärmbilden innehåller en lista av bokmärken i applikationen och att det förra valet som användaren gjorde var just "Bokmärken".

Det är bra om rubriker inte är för långa eftersom de på grund av platsbrist på mobiltelefonens skärm kommer att brytas av med en radbrytning och kan bli svåra att läsa. Om rubriken är en lång fras, t.ex. "Vädret i Göteborg", är det bättre att göra om rubriken så att den blir mindre specifik t.ex. "Väder". Men i vissa fall kan längre rubriker vara att föredra framför vissa förkortningar, eftersom användaren antagligen blir mer förvirrad av obekanta förkortningar och de kan vara svåra att förstå. Man bör även undvika att stryka under ord, eftersom understruken text lätt kan förväxlas med hyperlänkar.

6.2.5 Användning av länkar

Länkar är en bra sätt att länka samman information och är ett alternativ för att undvika långa sidor med text. Internet har gjort användarna vana vid länkar och användarna vet intuitivt att ett klick på en länk leder vidare till mer information. För att öka läsbarheten är det bra att försöka hålla länkarnas längd så kort som möjligt och helst inte låta dem vara så långa att de sträcker sig över två rader. Navigeringslänkar bör placeras i början eller i slutet och inte mitt inne i en textmassa.

6.2.6 Inmatning av text

Att mata in text på mobiltelefoner är alltid svårt. De flesta mobiltelefoner har bara vanliga telefontangenter och därför bör man i telefonapplikationer, i de fall det är möjligt, undvika att användaren själv matar in information. Det är bättre att använda listor som användaren kan välja ifrån.

⁶⁰ Nokia. (2000). *Service Developer's Guide*

När användaren ska mata in lösenord behöver inte lösenordet döljas. Terminalen är en telefon, skärmen är liten och användaren står bortvänd från andra. Att dölja lösenordet försvårar bara för användaren.

6.2.7 Konsekvent terminologi

Det är viktigt att hålla terminologin konsekvent. Inkonsekvent terminologi förvirrar ofta användaren och gör navigeringen komplex. Konsekvent terminologi gör det möjligt för användaren att tänka i termer av skärmbilder och att känna igen varje skärmbild när de navigerar mellan dem. Ett konsekvent utformat program gör det också lättare för användarna att upptäcka hur de ska utföra olika aktiviteter.

För att underlätta användning och snabb sökning i listor bör innehållet sorteras efter sammanhang, t.ex typ, datum, alfabetiskt, m.m.

7. Metod

I detta avsnitt kommer vi att beskriva de metoder vi har använt oss av för att besvara vår problemställning och komma fram till ett resultat. De metoder vi har använt oss av är litteraturstudier, prototyping och användbarhetstestning. Här nedan följer beskrivning av vad prototyping och användbarhetstestning innebär.

7.1 Prototyping

Uppgiften att utveckla datasystem är komplex och det finns inga enkla lösningar. Det finns en uppsjö av teorier, metoder och tekniker som föreskriver hur man ska utveckla datasystem. Oberoende av vilken metod man väljer vid systemutvecklingen är det dock viktigast att man får det system som man efterfrågar, att det är till belåtenhet för användarna. Det är ju för användarna som systemet finns till och det ska underlätta deras arbete. Det är därför viktigt att användarna får vara med att utforma systemet och får möjlighet att ange sina önskemål och krav. Prototyping tar hänsyn till denna aspekt och är en systemutvecklingsmetod som fokuserar på osäkerhet i samband med kravspecifikation och föreslår en experimentell strategi för problemlösning.

7.1.1 Vad är prototyping?

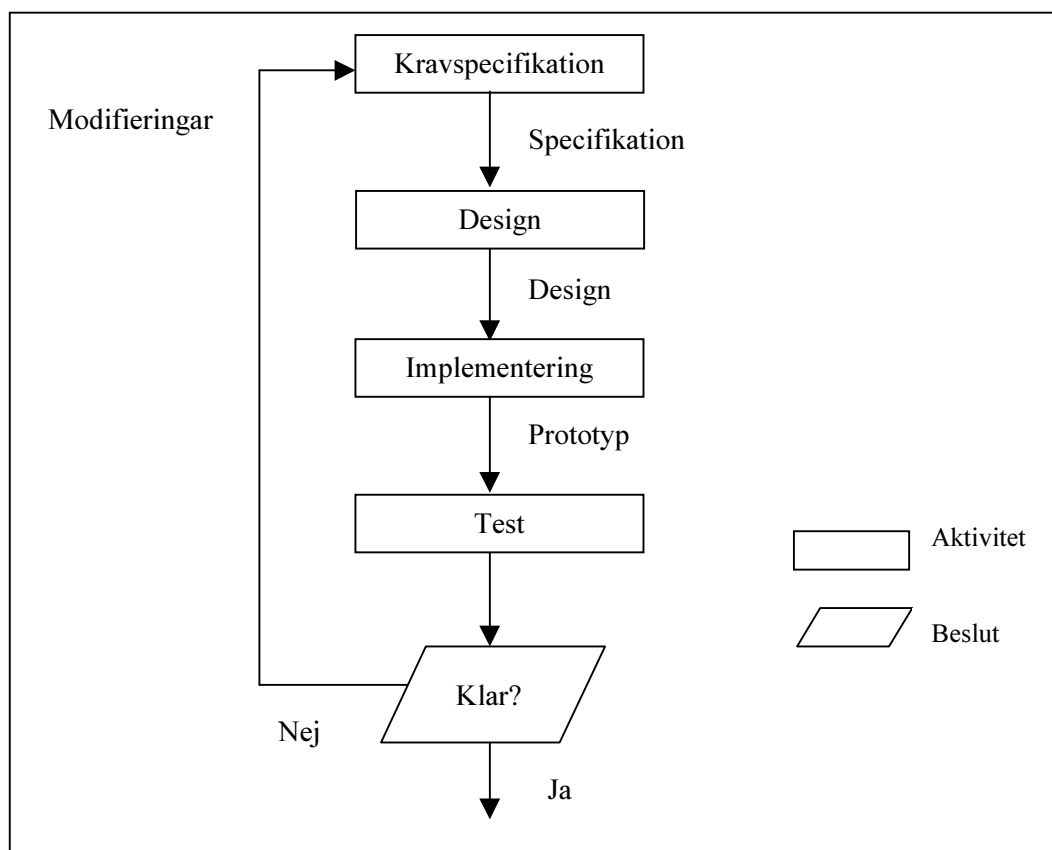
Prototyping har fått en speciell innebörd i systemutvecklingssammanhang och används i samband med kravspecifiering och utveckling av system.⁶¹ Någon bra svensk synonym finns inte, men man skulle kunna uttrycka sig med några ord och kalla prototyping för konstruktion av en systemmodell, som kan användas praktiskt i vissa sammanhang t ex för kravspecifiering och testverksamhet.

Prototyping innebär byggandet av en serie systemprototyper, där varje prototyp utformas efter de senaste erhållna kraven från användarna. När en prototyp är färdig testas den för att få användarna att kommentera och uttala sig om dess operationer och möjligheter. Användarnas kommentarer tillsammans med den nuvarande prototypen formar specifikationen till nästa prototyp som ska utvecklas. I vissa fall kan prototyping resultera i ett färdigt produktionssystem men prototyping tillämpas mest för kravspecifikation i situationer när användarkraven inte är definierade.⁶²

Konceptuellt kan prototyping ses som en serie av mycket snabba upprepningar av de aktiviteter som återfinns inom det traditionella sättet för systemutveckling. Dessa aktiviteter består av flera faser som är: analys (resulterar i en kravspecifikation), design och implementation. Varje upprepning av dessa aktiviteter resulterar i en prototyp som testas. Detta visas i figur 7.1. Fördelen med detta sätt att arbeta är att processen fortgår steg för steg, en liten bit i taget. Då tappar man så att säga inte ”bort sig”, utan man vet hela tiden vad man har ändrat på från en gång till en annan och kan då direkt bedöma om modifikationen var bra eller mindre bra.

⁶¹ Andersson, E S. (1994). *Systemutveckling – principer, metoder och tekniker*

⁶² Dahlbom, B., Mathiassen, L. (1993). *Computer in context*



Figur 7.1: Prototyping som en serie iterationer av den traditionella utvecklingsaktiviteten

Skillnaden mellan prototyping och traditionell systemutveckling ligger inte i de utförda aktiviteterna, utan i sättet, antal upprepningar av aktiviteterna och tiden som förflyter mellan det att de utförs. Om någonting i systemet skulle behöva modifieras är det enkelt i prototypingsammanhang att gå tillbaka till kravspecifikationen i utvecklingsprocessen.⁶³ Detta gör man vanligtvis inte när det gäller traditionell systemutveckling. Det tar alltför mycket tid i anspråk och upprättandet av systemet skulle därigenom bli försenat, och det skulle dessutom vara kostsamt.

I det traditionella tillvägagångssättet för systemutveckling krävs det att man fastställer kraven på systemet för att sedan kunna gå vidare till implementeringsfasen. Detta innebär ofta att man låser sig vid en viss lösning. Ändringar och anpassningar efter driftsättning blir svåra att göra, och i vissa fall helt omöjliga. Den vanligaste orsaken till dessa problem är att kravspecificeringen är svår att avgränsa, beroende på att nya idéer och önskemål uppträder i de följande faserna och att alla fakta och möjligheter inte är kända från början utan framträder under arbetets gång.⁶⁴ Detta kan medföra att det slutliga systemet fungerar otillfredsställande eller till och med blir oanvändbart ur användarnas synpunkt. Därför har prototyping blivit ett effektivt redskap för att reducera den tid det tar för att få fram en fullständig kravspecifikation då systemkraven

⁶³ Dahlbom, B., Mathiassen, L. (1993). *Computer in context*

⁶⁴ Guimaraes, T., Saraph, J V. (1991). *The role of prototyping in executive decision systems*

är vagt definierade. En exakt systemspecifikation är viktig för framgångsrik systemutveckling och implementering.

Användarna har ofta svårt för att definiera systemkrav, vilket t ex kan bero på att:⁶⁵

- problemet är komplext och svårt att förstå.
- användaren kanske förstår problemet, men har svårt att klart och tydligt kommunicera detta till systemutvecklarna.
- problemet ändrar karaktär under tiden som systemet utvecklas.
- problemet är ostrukturerat och detta hindrar en klar specifikation att ta form till dess att en prototyp har utarbetats och blivit prövad av användarna.
- systemet ska betjäna många användare med olika behov och intressen, och detta medför att systemkraven blir oklara och flyktiga.

En annan viktig fördel med prototyping är att projektgruppen kan ta tillvara program som utvecklats i tidigare skeden eller som finns till hands i verktygslådan. Utvecklingen kan då liknas vid att lägga ett komplicerat pussel, varvid man även kan ändra på pusselbitarna så att slutprodukten blir så som man har tänkt sig.

Prototyping är lämpligt för små problem och är särskilt användbart för att fastställa krav på systemet, men för komplexa problem har prototyping blivit kritiserat. Ett annat exempel på när prototyping inte är lämplig som systemutvecklingsmetod är t.ex vid utveckling av ett väldefinierat system, där osäkerheten är liten, programmets komplexitet är hög, det förekommer stora datavolymer och transaktionsvolymer. I sådana fall kommer antagligen inte prototyping att fungera på ett tillfredställande sätt. I situationer där systemets komplexitet är hög, men osäkerheten är låg, dvs det finns klart definierade systemkrav, bör strukturerade metoder användas. Medan i fall där komplexiteten är låg, men osäkerheten är hög kring systemkraven, är prototyping lämplig som systemutvecklingsmetod. I fall med både hög komplexitet och hög osäkerhet är det bra att kombinera prototyping med en eller flera mer strukturerade metoder.⁶⁶

7.1.2 Användarmedverkan

Datorstödet ska diskuteras utifrån användarens behov och användarna ska själva vara med och bestämma innehållet. Det är därför viktigt att användarna och andra personer från verksamheten får delta aktivt i analysarbetet. Prototyping kräver en mycket hög nivå av interaktion mellan användare och systemutvecklare, vilket medför att användarna får ett system som de vill ha och efterfrågar. Det har även visat sig att detta ökar användarnas acceptans och ökar användandet av systemet.⁶⁷

Prototyping kan användas för att få fram ett bra användargränssnitt. En prototyp över användargränssnittet består ofta av en hierarki av menyskärmbilder och ett schema över in- och utmatning. Sedan får användaren navigera genom dessa och kommentera utformningen och sammansättningen av skärmbilderna. Användaren bedömer graden av

⁶⁵ Johannson, L. (1994). *Analysen är grunden för ett lyckat utvecklingsprojekt*

⁶⁶ Shoval, P., Pilskin, N. (1988). *Structured Prototyping: Integrating Prototyping into Structured System Development*

⁶⁷ Guimaraes, T., Saraph, J V. (1991). *The role of prototyping in executive decision systems*

lätthet att röra sig mellan de olika skärmbilderna och huruvida de är ordnade efter logiskt tänkande, så att användaren kan använda sig av sitt spatiala tänkande och göra sig en slags karta i huvudet hur han eller hon ska ta sig till nästa skärmbild, osv. Användaren får även här reagera på skärmbildernas utseende för att systemutvecklaren ska kunna förbättra dem både visuellt och funktionellt. I samband med detta kontrollerar man också att felhantering av de olika inmatningsfälten fungerar tillfredställande och att förklaringsstexter är tillräckligt tydliga.⁶⁸

7.1.3 Sammanfattningsvis

Prototyping hjälper till att definiera bättre systemspecifikation och förbättrar kommunikationen mellan användare och utvecklare, samt ger ett flexibelt system som är anpassat för att man lätt ska kunna åstadkomma ändringar. Som ett resultat är prototyping tänkt att reducera tiden för systemutvecklingen, förbättra systemets kvalitet och öka användarnas tillfredsställelse med systemet.

7.2 Användbarhetstestning⁶⁹

Användbarhetstestning⁷⁰ är ett systematiskt sätt att observera verkliga användare som testar en produkt och samlar information om på vilka sätt produkten är lätt eller svår att använda. När användbarhetstestning introducerades på 1980-talet gjorde man testet på samma sätt som man hade gjort funktionstestning och därför gjordes testningen endast en gång i slutet av utvecklingsprocessen. Antagandet var i de flesta fall att testet skulle verifiera att allting var bra och om det var några mindre problem så skulle de lätt kunna åtgärdas. Det första användbarhetstesterna upprörde människor eftersom allting i programmen inte var bra och att fler problem dök upp i testerna än vad som kunde åtgärdas. Under 90-talet har modellerna för både funktionstestning och användbarhetstestning förändrats. Användbarhetstestning används för att hitta problem, inte för att verifiera att allt är bra. Testningen ger också bäst resultat om den görs tidigt och ofta och inte en gång i slutet när det är för sent att göra ändringar. Testningen bör genomföras som en del i en process som fokuserar på användbarhet, inte som ett enstaka tillfälle då man tänker på användarna.

7.2.1 Vad är användbarhetstestning?

Alla användbarhetstest har följande fem karakteristika oberoende av var och hur de genomförs:

1. *Det primära målet är att förbättra produktens användbarhet.* För varje test sätts mer specifika mål som används som utgångspunkt vid planering av testet. Ett annat mål med testningen är att förbättra processen under vilken produkter designas och utvecklas så att man kan undvika att samma problem uppstår igen på andra produkter. Detta skiljer ett användbarhetstest från en forskningsstudie där målet är att undersöka förekomsten av ett visst fenomen. Användbarhetstest skiljer sig också på grund av detta från funktionstest som har som mål att bedöma om en produkt

⁶⁸ Shoval, P., Pilskin, N. (1988). *Structured Prototyping: Integrating Prototyping into Structured System Development*

⁶⁹ Dumas, J S., Redish, J C. (1994). *A practical guide to usability testing*

⁷⁰ engelska: Usability testing

följer specifikationen.

2. *Deltagarna representerar verkliga användare.* De personer som ska testa produkten måste tillhöra den grupp som använder eller kommer att använda produkten. Om deltagarna är mer erfarna än de verkliga användarna kan man missa problem som kan dyka upp när produkten ska användas på riktigt. Om deltagarna är mindre erfarna än de verkliga användarna kan förändringar göras som inte leder till förbättringar för de verkliga användarna.
3. *Verklighetstrogna uppgifter.* Uppgifterna som ska utföras i testet måste vara riktiga uppgifter som användarna kommer att använda produkten till i sitt hem eller på jobbet. För att kunna ta fram realistiska uppgifter måste man förstå användarnas arbete och de arbetsuppgifter som produkten ska användas till. Vid många användbarhetstest, framförallt sådana med funktionsrika och komplexa mjukvaruprodukter, kan endast några av alla de relevanta uppgifterna att kunna testas. Därför måste uppgiften, förutom att vara realistisk och relevant för användarna, även relatera till testets mål. Man bör också välja uppgifter som med hög sannolikhet leder till att användbarhetsproblem upptäcks.
4. *Observation och insamling.* I användbarhetstest används vanligtvis flera personer som en i taget kommer för att arbeta med en produkt. Deltagaren observeras och man registrerar både hur de använder produkten och de kommentarer användarna har. Det är också lämpligt att fråga deltagarna om åsikter som de har om produkten. Ett användbarhetstest inkluderar både det tillfälle då deltagarna utför uppgifter med produkten och den tid när de fyller i frågeformulär om produkten.
5. *Analysera insamlade data, diagnostisera problemen och rekommendera ändringar.* Att samla in data är viktigt men inte tillräckligt i ett användbarhetstest. Efter själva testet måste data analyseras. Kvantitativa och kvalitativa data från deltagarna läggs ihop med de egna observationerna och användarnas kommentarer. Allt detta används för att diagnostisera och dokumentera produktens användbarhetsproblem och för att rekommendera lösningar på problemen.

Resultatet används för att förändra produkten och processen. Ett användbarhetstest är framgångsrikt endast om det hjälper till att förbättra produkten som testades och processen under vilken produkten utvecklas. Att förbättra processen och undvika att misstag upprepas har mer långsiktig inverkan på användbarhet än att endast förbättra produkten.

7.2.2 När är ett användbarhetstest lämpligt?

Det finns inget som begränsar ett användbarhetstest till att endast genomföras en gång i slutet av ett projekt utan testet kan genomföras iterativt under hela design- och utvecklingsprocessen.

Användbarhetstest kan förändra människors attityd till användare. Designers, utvecklare, chefer med flera som närvarar vid användbarhetstest för första gången får ofta en annan syn på och förståelse för användare och användbarhetsfrågor. Att se ett fåtal människor kämpa med en produkt har större påverkan än många timmars diskussioner om hur viktigt det är med användbarhet och förståelse för användare.

7.2.3 Användbarhetstestning jämfört med forskningsstudier

Användbarhetstestning och vetenskaplig forskning är båda empiriska metoder vilket innebär att de båda fokuserar på observation av verkliga beteenden. På ytan ser ett användbarhetstest ut som en forskningsstudie. Tester genomförs ofta i laboratorier och deltagarna representerar den användargrupp som är av intresse för studien. När man väljer ut och tränar deltagarna vidtas vissa åtgärder för att vissa variabler ska kunna kontrolleras för att inte försvåra tolkningen av resultatet. Både objektiva och subjektiva mätningar görs. Sedan analyseras data och en rapport skrivs utifrån empiriska data som stödjer det man kommit fram till.

Men de finns också stora skillnader mellan användbarhetstestning och vetenskaplig forskning:

- *Mål.* Den största skillnaden ligger i målen. Målet med en forskningsstudie är att testa om ett visst fenomen existerar eller inte. För att kunna avgöra det måste testet utföras med ett tillräckligt stort antal deltagare. Målet med användbarhetstestning är att upptäcka problem, inte att demonstrera existensen av ett visst fenomen. Erfarenhet har visat att man kan hitta de flesta stora problem med hjälp av användbarhetstest med relativt få deltagare. Det går däremot inte att applicera samma statistiska tester på data som det går att göra i en vetenskaplig studie.
- *Deltagare.* I forskningsstudier är det mycket viktigt hur urvalet av de personer som ska delta i testerna görs. I användbarhetstest väljer man personer från en passande användargrupp som för tillfället finns tillgängliga.
- *Observation.* I användbarhetstest läggs större vikt vid det som observerats än vad man gör i forskningsstudier.
- *Statistik.* Beskrivande statistik såsom medelvärden, frekvenser m.m. används i användbarhetstester, men i forskningsstudier drar man ofta slutsatser ur statistikresultaten som sedan ofta ligger till grund för diskussioner om datan.

7.2.4 Planering

Planering är mycket viktigt för att kunna göra ett lyckat användbarhetstest. Att ta in ett par personer för att se hur de arbetar med en produkt är bortkastad tid om man inte tänkt på följande:

- Vilka aspekter på produkten är kanske inte så användbara som de borde vara?
- Hur väl representerar deltagarna de verkliga användarna av produkten?
- Vilka uppgifter ska deltagarna utföra under den korta tid de har till förfogande?
- Vilken information ska samlas in när deltagarna observeras?
- Hur ska den insamlade informationen analyseras?
- Vad ska man göra med informationen när den har analyserats?

Planering innebär följande steg:

- Definiera målen med testet.
- Bestäm vilka som ska delta.
- Rekrytera deltagare.

- Välj och organisera uppgifter att testa.
- Skapa uppgiftsscenarion.
- Bestäm hur användbarheten ska mätas.
- Förbered annat material för testet.
- Förbered testmiljön.
- Förbered testteamet genom att dela ut roller, utbilda teammedlemmar och öva innan testet startar.
- Genomför ett pilottest och gör ändringar om det behövs.

7.2.4.1 Välja mellan mål

Vid planering av ett användbarhetstest måste man nästan alltid göra vissa val, är målet t.ex att se om människor överhuvudtaget använder vissa funktioner eller om funktionerna är lätta att använda? Det kan vara svårt att samtidigt få svar på båda frågorna och frågorna bör istället testas var för sig i olika tester.

7.2.4.2 Gå från generella frågor till specifika

För att planera testet bör man gå från generella frågor till mer specifika. Att säga att man vill veta om användarna tycker att en produkt är lätt att använda är en bra början, men är en alldeles för vagt ställd fråga. Den som planerar testet bör istället fundera över vilka aspekter av produkten som är mest bekymmersamma och vilken användargrupp som är intressantast. Man bör välja ut de uppgifter som man anar har användbarhetsproblem och tänka igenom hur man på bästa sätt tar reda på om användarna har problem med att utföra en uppgift.

Att sätta upp generella frågor hjälper ofta till att bestämma vilka personer som ska delta i testet. Genom att göra de generella frågorna mer specifika kommer man fram till vilka uppgifter deltagarna ska utföra, hur testet ska utföras och vad som ska mätas.

7.2.4.3 Hur många ska delta i testet?

Hur många deltagare som behöver medverka i ett användbarhetstest är en stor och omdiskuterad fråga. Man måste komma ihåg att användbarhetstest inte är en forskningsstudie. Syftet med testet är att upptäcka de allvarligaste problemen som användare kan tänkas uppleva med en produkt. Nielsen och Molich kom fram till att knappt hälften av alla stora problem upptäcktes med tre deltagare. Virzi såg att 80% av problemen upptäcktes med mellan 4 och 5 deltagare och 90% av problemen upptäcktes med 10 deltagare.

Antalet personer som ska delta i testet är beroende av hur många olika användargrupper med olika erfarenheter och färdigheter man behöver testa på. Hur mycket tid och pengar man har möjlighet att lägga på testet och hur viktigt det är att kunna beräkna statistisk signifikans på resultaten har också en avgörande betydelse.

En allmän regel är att det är bättre att göra en utvärdering med ett litet antal deltagare än att inte göra någon empirisk utvärdering alls. Utvärdering med hjälp av ett litet antal användare visar sig ofta ge väldigt mycket nyttig information. Det är inte alltid att ett tillskott av ytterligare personer ger en utdelning som är värd besväret.⁷¹

⁷¹ Allwood, C M. (1998). *Människa datorinteraktion – ett psykologiskt perspektiv*

7.2.4.4 Val av medverkande användare

När det gäller att bestämma vilka användare som ska delta i de empiriska utvärderingar som genomförs bör man först skaffa sig en överblick över vilka kategorier av personer som kommer att använda systemet. I många sammanhang är det inte realistiskt att sträva efter ett slumpmässigt urval. I stället är det oftast frågan om att handplocka de användarrepresentanter som deltar i utvärderingen. Detta minskar inte värdet av testet på ett avgörande sätt, eftersom man inte i första hand är ute efter strikt vetenskapliga slutsatser utan efter att öka sin förståelse för hur man kan producera bättre produkter.

7.2.4.5 Scenario

När en lista över uppgifter för testet är fastställd måste man bestämma hur uppgifterna ska presenteras för deltagarna. Ett bra sätt är att ge deltagarna scenarion, situationer där uppgiften finns inbäddad i en mycket kort historia. Ett scenario används för att instruera deltagarna om vad de ska utföra under testet.

Ett bra scenario är kort, använder användarens ord och inte produktens. Vidare ska ett scenario vara otvetydigt så att alla deltagare kan förstå det och det bör ge deltagarna tillräcklig information för att de ska kunna utföra uppgifterna. Scenariot bör ha en tydlig koppling till testets uppgifter och mål. Alla scenarion måste skrivas ut som en del av testplanen. Även om deltagarna inte ser det skrivna scenariot måste man försäkra sig om att varje deltagare får samma scenario framställt på likadant sätt.

7.2.5 Att mäta användbarhet

I ett användbarhetstest mäts följande :

- Mått på användningen, dvs vad och hur deltagarna gör när de använder produkten observeras. Dessa mått är kvantitativa, man kan t.ex mäta hur lång tid något tar, hur många fel som görs, hur många gånger samma fel upprepas. Dessa mått kräver noggrann observation.
- Subjektiva mått, dvs deltagarnas perception, åsikter och omdömen. Dessa mått kan vara antingen kvantitativa eller kvalitativa. Exempelvis kan man ge deltagarna en 5- eller 7-gradig skala och be dem gradera hur lätt eller svår en produkt är att använda. Bedömningen är subjektiv men man får kvantitativ respons. Deltagarnas spontana kommentarer om produkten kan också registreras genom att de ombeds tänka högt när de arbetar med produkten. Kommentarererna är både subjektiva och kvalitativa och det går även se hur många som kommenterade ett visst problem.

7.2.5.1 Frågeformulär

Under ett test finns det flera tillfällen då man vill ställa frågor till deltagarna t ex om deras bakgrund, åsikter om en viss uppgift, uppfattning om en produkt osv. Man bör tänka igenom alla frågor som man vill ställa och skriva ned dem. Det finns två anledningar till att ha skrivna frågeformulär. De är 1) att försäkra sig om att samma frågor ställs till alla deltagare och 2) att man inte glömmer bort att ställa en fråga.

Enkäter är också enkla att administrera och analysera, men man kan inte vara säker på att användarna uppfattar frågorna på det sätt man tänkt sig.⁷² Dessutom finns ingen

⁷² Allwood, C M. (1998). *Människa datorinteraktion – ett psykologiskt perspektiv*

möjlighet att ställa uppföljningsfrågor om det är något som är oklart eller om något kommer upp som ligger utanför frågan, men verkar intressant.

7.2.5.2 Intervju

En stor fördel med intervjuer är just att det finns möjlighet att ställa uppföljningsfrågor av olika slag och därmed finns också en chans att försäkra sig om att den intervjuade uppfattar frågan på det sätt man tänkt sig. En nackdel med både enkäter och intervjuer är att det i efterhand kan vara svårt för användare att komma ihåg vad som var problematiskt under datorinteraktionen. Detta bekräftas av flera studier som visar att användare efteråt mest kommer ihåg de riktigt stora problemen. Tidsavståndet mellan användning och datainsamling kan också göra att de svårigheter man till slut lyckades ta sig ur inte längre framstår som problematiska och därmed inte värda att rapportera.

7.2.5.3 Uppmana deltagarna att tänka högt

Människor tänker vanligtvis inte högt när de arbetar och testdeltagare varierar mycket i sin förmåga att berätta vad de tänker medan de arbetar. Som en del av instruktionerna till deltagarna kan man berätta för dem att man är intresserad av hur de tänker, eftersom deras reaktioner vid användning av produkten är värdefulla. Användarna bör därför uppmanas att tänka högt och om de är tysta för länge bör man åter påminna dem om att tänka högt.

7.2.6 Användbarhetstest av webbsidor

Enligt Jacob Nielsen är det inte mödan värt att göra mer än fem användartester av en webbsida. Hans forskning har visat att det första testet avslöjar flest brister och att nästa test fortfarande ger rätt mycket information men inte lika mycket. Kör man femton tester så bör nästan alla brister som finns i designen upptäckas, men det femtonde testet tillför inte särskilt mycket. Störst utbyte ger de första tre till fem testerna⁷³.

⁷³ Geijer, E. (2000). *Fem personer räcker för att testa din webbplats*

8. Genomförande av studie

För att undersöka i hur stor utsträckning man kan tillämpa de riktlinjer som används vid traditionell gränssnittsdesign vid utformning av användargränssnitt för små skärmar, tyckte vi det var lämpligt att själva utforma en applikation för mobiltelefoner. Ett annat skäl var att bilda oss en uppfattning om de befintliga riktlinjerna för utformning av gränssnitt för mobiltelefoner är tillräckliga, eller om det finns något ytterligare som man bör tänka på. Vi ville inte endast förlita oss på vad andra sagt utan själva se vad som var möjligt att göra med den teknik som fanns tillgänglig. Vi valde att ta fram en prototyp för att sedan utvärdera den och utifrån användarnas åsikter utveckla ytterligare en prototyp.

På Semcon fick vi möjlighet att utveckla en WAP-applikation. Vår uppgift var att utöka deras webbaserade tidsrapporteringssystem, TimeIT, med en mobil klient som skulle implementeras med hjälp av WAP-tekniken. Kravet på WAP-klienten var att den skulle ha ungefär samma funktioner och innehåll som det befintliga systemet. Eftersom kravspecifikationen över innehåll och funktioner var tydlig, men det saknades en specifikation över hur gränssnittet för mobiltelefon skulle se ut, koncentrerade vi oss på att utforma ett användargränssnitt. Vid utformningen utgick vi ifrån de riktlinjer som finns beskrivna i litteraturen.

Vid utveckling av WAP-applikationer finns det ett antal olika utvecklingsmiljöer som simulerar en WAP-telefon på datorskärmen. Simulatorens har samma funktionalitet som en riktig mobiltelefon. Fördelen med simulatorens är bland annat att det går snabbare att köra programmen och det finns stöd för felsökning vilket underlättar utveckling och testning. Vi började med att använda Nokias simulator, men övergick efter en tid till Ericssons utvecklingsmiljö. Det fanns flera skäl till detta. Det främsta skälet var att vi hade tillgång till Ericssons R320s WAP-telefon och vi ville så småningom kunna köra skarpt. Ett annat skäl var att Nokias simulator inte var en riktig WAP-telefon och hade dessutom en ganska liten skärm. De riktiga mobiltelefonerna och simulatormiljöerna tolkar WML-koden lite olika och därför ser applikationen inte likadan ut i telefonen som i simulatorens. Det förekommer också skillnader mellan hur de olika telefonmärkena tolkar och grafiskt visar WML-koden. Ericssons telefoner visar t ex inte inmatningsfält på samma sätt som Nokia gör.

Efter att ha testat hur programmeringsspråket WML fungerade började vi utveckla användargränssnittet till tidsrapporteringssystemet. Vi försökte så långt det var möjligt att följa de riktlinjer för traditionell gränssnittsdesign som vi läst om och som finns beskrivna i teoriavsnittet. Vi försökte också att följa de befintliga riktlinjerna för utformning av gränssnitt för små skärmar. Målet med utformningen av det första gränssnittet var att utforma det så att det var lätt att använda och enkelt att navigera i.

8.1 Användbarhetstest av prototypens gränssnitt

Vår avsikt med testet av applikationen var att undersöka hur väl de riktlinjer som vi kommit fram till genom litteraturstudier och egna erfarenheter överensstämmer med användarnas uppfattning om vad som anses användbart. Personer som inte har varit involverade i utformning och utveckling av en applikation tenderar att upptäcka möjliga

användbarhetsproblem som inte har varit uppenbara för dem som har designat applikationen.

För att undersöka hur användarna uppfattade gränssnittet, se hur intuitivt det var, vad de tyckte om det, och för att få reda på vad som var bra och vad som kunde förbättras, gjorde vi ett test för att utvärdera användbarheten. Med ledning av vad vi läst i litteraturen beslutade vi att en testgrupp på sex personer var lämpligt. Personerna i testgruppen bestod av de framtida användarna av applikationen, dvs personer anställda på Semcon. Vi valde personer som fanns tillgängliga så att vi skulle kunna återkomma till dem med eventuella ytterligare frågor. Vi ville att alla testpersoner skulle vara välbekanta med Semcons tidsrapporteringssystem för att det inte skulle uppstå problem med de termer som används i systemet.

När användarna skulle testa prototypen gjorde vi ett användarscenario som innehöll uppgifter som användaren skulle utföra. Innan testet började hade vi uppmanat användarna att under testets gång tänka högt och kommentera vad de gjorde och om något kändes oklart. Under tiden som användaren utförde uppgifterna observerade vi vad de gjorde och hur de utförde uppgifterna. Vi ställde även frågor under tiden för att få klarhet i vad användarna ansåg om utformningen och de fick ge allmänna omdömen om hur de uppfattat användningen av applikationen och vad som kunde förbättras. Under användbarhetstestetets gång förde vi anteckningar för att senare kunna utvärdera gränssnittets användbarhet.

Efter användbarhetstestet sammanställde vi materialet, både våra egna och användarnas iakttagelser och drog slutsatser utifrån detta. Utvecklingen av den andra prototypen gjordes med utgångspunkt från de åsikter och den erfarenheter vi fått genom testet av den första prototypen. Även den andra prototypen genomgick ett användbarhetstest som utfördes på samma sätt som det första. Avsikten med det andra användbarhetstestet var att förvissa oss om att de förändringar vi hade gjort medförde att applikationen blev mer användbar. Resultatet av testerna presenteras i kapitel 9, *Resultat och diskussion*.

9. Resultat och diskussion

I detta avsnitt har vi valt att beskriva WAP-applikationen (TimeIT) och de aspekter vi beaktade när vi utformade användargränssnittet. Vi kommer även att presentera och diskutera de resultat som framkommit genom användbarhetstesterna, litteraturstudierna och våra egna erfarenheter.

9.1 Utformning av applikationen

Ett av våra mål med utformningen av gränssnittet var att göra en applikation så enkel att användarna skulle kunna använda den direkt utan några instruktioner eller manualer. Det enda som skulle krävas av användarna var att de skulle känna till och ha använt det befintliga tidsrapporteringsystemet och veta hur en WAP-telefon fungerar.

En av uppgifterna som skulle kunna utföras med hjälp av applikationen var rapportering av antal timmar en person har arbetat på ett visst projekt en viss vecka, detta kallas för "Tidrad". Uppgiften att rapportera in en tidrad består i sin tur av sju värden som ska matas in eller väljas från en lista. Vi valde att lägga de olika värdena på var sin skärmbild, vilket innebar att man utförde uppgiften genom att gå igenom en sekvens som bestod av sju steg. Vi strävade efter att varje skärmbild skulle vara enkelt utformad och endast bestå av ett inmatningsfält eller en länk till en lista med olika val, samt en länk vidare till nästa sida ("Next"). Se figur 9.1. Ett skäl till att vi valde att placera inmatning av värden i en sekvens var att det är lättare för användare att veta vad de ska göra om uppgifterna utförs i en följd i en sammanhängande sekvens⁷⁴. En sekvens underlättar för användare genom att den inte ger så många chanser för användaren att trycka fel och komma vilse i menyer och nivåer.



Figur 9.1: En skärmbild i sekvensen. Tidrad 2:8 betyder att användaren befinner sig på skärmbild två av åtta i sekvensen för att utföra en tidrad.

En annan anledning till att vi delade upp uppgiften så att den skulle utföras i en sekvens var att vi inte ville gömma någon information. Vi ville undvika att användaren skulle behöva scrolla för att se innehållet på skärmbilden⁷⁵, samtidigt som vi ville visa så mycket information så att användaren skulle kunna göra rätt val.

⁷⁴ Se kap 4.3

⁷⁵ Se kap 6

Ett annat försök att underlätta för användaren var att med hjälp av siffror indikera var i sekvensen de befann sig, t.ex Tidrad 2:8. Se figur 9.1. Det kan ofta vara en fördel att användaren vet var i sekvensen han eller hon befinner sig och hur många moment som återstår.⁷⁶ Eftersom användning av sekvenser till viss del minskar överblickbarheten valde vi att på sekvensens sista skärmbild visa en sammanställning över de valda värdena, för att användaren skulle kunna kontrollera att rätt värden valts innan han eller hon valde att spara tidraden. I testet av gränssnittet sa de flesta användare att de tyckte att det var bra med en sammanställning i slutet som visar det man matat in och vad som kommer att sparas.

I alla typer av mjukvaruapplikationer är det bra att ha så få knapptryckningar som möjligt och det är extra viktigt i mobila applikationer där användaren inte kan ägna hela sin uppmärksamhet på applikationen utan ofta måste koncentrera sig på andra saker runt omkring samtidigt som de använder terminalen. För att reducera antalet knapptryckningar⁷⁷ ville vi lägga in förvalda värden, dvs värden anpassade efter vad användaren oftast väljer. Denna typ av individuell anpassning skulle innebära att användaren direkt får upp det förvalda värdet och slipper gå in i en lista för att göra ett val och kan istället gå direkt till "Next".

9.2 Resultat och diskussion av användbarhetstesterna

Vi har delat in resultatet av användbarhetstesterna i fem avsnitt som följer nedan.

9.2.1 Mobiltelefonvana

Vid användbarhetstestets början ställde vi bakgrundsfrågor angående deltagarnas mobiltelefonvana. Anledningen till att vi ville skapa oss en bild av deltagarnas erfarenhet av att hantera en mobiltelefon var att vi ansåg att mobiltelefonvana var en viktig faktor att ha vetskap om, eftersom användare som är obekanta med mobiltelefonens funktionalitet eventuellt skulle ha svårt att använda applikationen. Vi ansåg att vi skulle få problem att avgöra om det var applikationen som var svår att använda eller om problemet låg i telefonens utformning, om vi inte klargjorde frågan om mobiltelefonvana.

De personer som deltog i testerna var vana vid att använda mobiltelefon och de flesta deltagarna hade störst erfarenhet av Nokias mobiltelefoner. Vårt val att använda Ericssons simulatormiljö (R320s) för testning av vår prototyp medförde därför att de som var ovana vid Ericssons telefoner till en början använde fel tangenter för att bekräfta val och följa länkar samt gå tillbaka. Men efter en stunds användning hade de lärt sig vad telefonens tangenter hade för funktioner.

9.2.2 Anpassa efter typ av mobiltelefon

Vi har märkt att en applikations användbarhet är beroende av vilken telefon som används. Placeringen av telefonernas funktionstangenter (tangenter för att bekräfta ett val eller gå tillbaka) kan göra att applikationen blir mer eller mindre intuitiv att använda. Exempelvis har funktionstangenterna på Ericssons mobiltelefoner texten "Yes" och "No". För att bekräfta ett val känns "Yes-tangenten" naturligt att använda,

⁷⁶ Shneiderman, B. (1997). *Designing the User Interface*

⁷⁷ Se kapitel 6

medan benämningen "No" inte blir lika självklar att använda för att gå tillbaka i applikationen. Därför informerade vi användarna om knappens funktion innan de började använda applikationen. Vi såg under testets gång att användarna visste hur de skulle gå tillväga för att gå tillbaka i applikationen och ångra om de hade gjort fel.

Några av deltagarna ansåg att utseendet på layouten stämde bra överens med bl a Ericssons upplägg av telefonens interna navigeringsförfaranden och de tyckte att applikationen hade en logisk följd och kändes lätt att navigera i.

Under testets gång observerade vi vissa återkommande feltryckningar. Användarna hade en tendens att vilja använda knappen "Yes" för att gå vidare (d v s bekräfta ett val) när de såg att det förvalda värdet stämde, istället för att ställa markören på "next" och därefter trycka "Yes". Eftersom markören stod på inmatningsfältet när de kom in på skärmbilden så ledde en tryckning på "Yes-knappen" till att listan med val kom upp ändå, trots att användarna inte ville välja ett annat värde. Se figur 9.2. Vi såg att detta var det problem som var mest återkommande och skapade förvirring för användarna. Vi beslutade därför att försöka åtgärda detta till prototyp nummer två.



Figur 9.2: Om användarna trycker på "Yes-knappen" direkt istället för att flytta markören till "Next" och därefter trycka "Yes" kommer listan med val upp.

Applikationer bör också anpassas efter den typ av mobiltelefon som de ska visas på eftersom olika terminaler tolkar WML-koden på olika sätt. Det innebär att applikationen får ett annorlunda utseende beroende på vilken terminal som används.

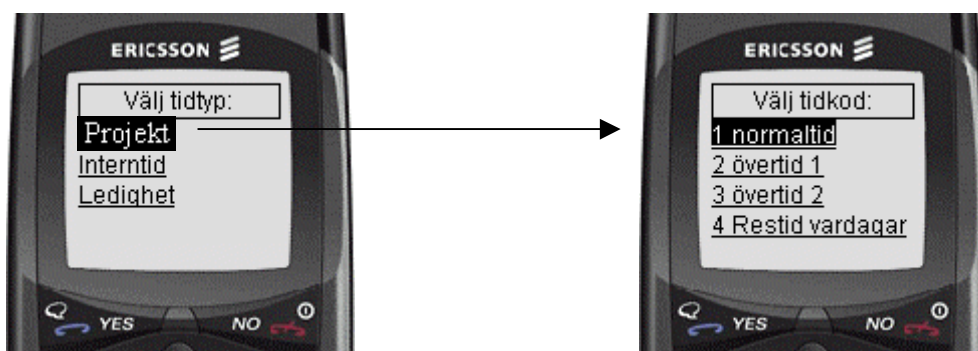
Vi har sett att man som utvecklare av applikationer för mobiltelefoner bör anpassa sig efter telefonens utformning och inbyggda funktionalitet. Det finns inte någon möjlighet att påverka knapparnas text och placering på samma sätt som i ett traditionellt gränssnitt där utvecklaren själv kan anpassa placering och innehåll efter varje specifik tillämpning.

9.2.3 Rubriker

Vid utformningen av gränssnittet till den första prototypen försökte vi göra tydliga rubriker till varje skärmbild och även använda en siffra (ex Tidrad 2:8) som på ett enkelt sätt skulle visa var i applikationen användaren befann sig. Till varje skärmbild lade vi till en underrubrik som beskrev vad användaren skulle göra.

Vid användbarhetstestet sa användarna att de tyckte att rubrikerna var bra och gjorde att de visste var i applikationen de befann sig. Angående siffrorna som indikerade var i

sekvensen användaren befann sig rådde det delade meningar om. En användare sa att siffrorna var mycket bra och var till stor hjälp. Vissa användare hade inte uppmärksammat siffrorna eller ansåg att de inte hade någon användning för dem, men att det inte heller störde att de fanns där. En användare kom med idén att det vore bättre om huvudrubrikerna angav vad som ska väljas eller matas in på respektive skärmbild. I prototyp nummer två, tog vi fasta på den idén, vilket innebar att vi omarbetade varje skärmbild. Varje rubrik beskrev nu vad användaren skulle kunna välja eller göra på just den skärmbilden och vi försökte hålla rubrikerna korta och informativa. Istället för att använda det extra steget med en länk som leder vidare till ett nytt fönster med val, valde vi att direkt visa listan i det fönster man kommer till. Detta innebar att man inte behövde flytta ner markören till "Next" för att komma vidare till nästa sida i sekvensen. Användaren kunde istället direkt klicka på det önskade valet och automatiskt komma vidare till nästa sida. Se figur 9.3.



Figur 9.3: Gränssnitt enligt prototyp 2.

Användaren kan här direkt använda "Yes-knappen" när markören står på önskat val, och kommer därmed vidare till nästa skärmbild.

Fördelen med det nya gränssnittet var att vi kunde reducera antalet knapptryckningar avsevärt och vi kom ifrån den höga frekvensen av feltryckningar som vi nämnt tidigare (användarna glömde ofta att flytta markören till "Next" innan de tryckte "Yes"). Men denna förändring innebar även att överskådligheten, var i sekvensen man befann sig, inte kunde visas på ett lika tydligt sätt. Vi tyckte ändå att denna layoutmässiga kompromiss var positiv och det bekräftades även av användarna i testet av prototyp nummer två. Användarna tyckte att det var mycket enkelt att förstå vad de skulle göra och de ansåg att det var ett naturligt flöde i användningen av applikationen, vilket gjorde det enkelt för användaren.

9.2.4 Mycket information på skärmen

De uppgifter som man skulle utföra med hjälp av applikationen krävde ibland att den mängd information som visades tog för mycket plats för att allt skulle kunna synas på en gång, på en skärmbild. Eftersom användarna inte hade möjlighet att se all information i listan, trodde en del användare därför att det inte fanns mer information än den som syntes. Användarna tänkte inte på att de kunde scrolla nedåt och visste därmed inte hur de skulle komma vidare. Vi tror att detta kan bero på att det saknas en indikering som visar att det finns mer information längre ned på sidan. I de flesta andra datormiljöer finns t.ex. rullningslistor och något liknande tror vi skulle vara mycket användbart även i telefonapplikationer. I den mobiltelefon, en Ericsson R320s, som vi

har testat finns i den inbyggda funktionaliteten indikatorer som visar när en textmassa inte ryms på en skärmbild, men vi har inte hittat något sätt att använda det i egna applikationer.

9.2.5 Inmatning av text

Inmatning av text på en mobiltelefon är ganska krångligt och bör därför göras i så liten utsträckning som möjligt. Ett alternativ till att användaren själv matar in tecken är att ha en lista med val som användaren kan välja ifrån. Listor bör användas i de fall där det är möjligt, men ibland går det inte att undvika att användaren själv måste mata in tecken. En annan fördel med listor är att man minimerar risken för att felaktiga värden matas in.

I vår applikation försökte vi i så stor utsträckning som möjligt att använda förvalslistor men i vissa fall var det inte möjligt, t.ex vid inloggningen måste användaren själv ange användarnamn och lösenord. I användbarhetstestet observerade vi att användarna hade vissa svårigheter att skriva in text, framförallt när de måste blanda siffror och bokstäver. Vad gäller inmatning av lösenord hade användarna svårighet att mata in rätt tecken eftersom lösenorden ofta var långa och bestod av både siffror och stora och små bokstäver. I de flesta applikationer brukar lösenord döljas, men i Ericssons R320s syns lösenordet en kort stund för att sedan visas som stjärnor. Deltagarna i testet tyckte att detta var en bra lösning och hade inget emot att lösenordet syntes eftersom displayen var så liten och risken att någon annan skulle se lösenordet inte var så stor.

9.3 Hur traditionella principer går att applicera på mobiltelefongränssnitt

Här beskriver vi de principer och riktlinjer som traditionell systemutveckling föreslår och på vilket sätt vi har kunnat applicera dem på den applikation som vi har utformat.

9.3.1 Konsekvent utformning

Vid utveckling av alla användargränssnitt är det viktigt att utforma dem på ett konsekvent sätt för att underlätta användningen. Det gäller även mobiltelefonens gränssnitt, men i vissa fall krävs det att man gör vissa avvikelser från denna princip. Vissa typer av avvikelser måste antagligen accepteras för att ge utrymme åt andra fördelar som t ex att minska antalet knapptryckningar.

Vid design av gränssnitt för mobiltelefoner kan det i vissa fall vara svårt att vara konsekvent i utformningen. Vi försökte hålla en konsekvent utformning av vårt program både vad gäller ordval och skärmbildernas utseende, men vi märkte efter ett tag att det inte var genomförbart vad gäller utformningen av skärmbilderna. Vi utformade skärmbilderna så att, i de fall det var möjligt, användaren kunde göra sina val via listor. Men vi var ibland tvungna att avvika från det när situationen krävde att användaren själv matade in text. Detta gjorde att alla skärmbilder inte kunde se likadana ut och vi upptäckte i användbarhetstestet att användarna reagerade på detta och blev avbrutna i sin arbetsgång och hade svårt att genast förstå vad de skulle göra. Ett annan orsak till att skärmbildernas utseende kan komma att skilja sig från varandra är om ett ord är för långt för att få plats på en rad. Det påverkar resten av gränssnittets utseende så att det inte ser likadant ut som de andra skärmbilderna där användaren kan utföra liknande uppgifter.

En av orsakerna till att vi inte till fullo kunde följa riktlinjen om konsekvent utformning är att programmeringsspråket som applikationer till mobiltelefoner skrivs i, WML är begränsat och man kan som utvecklare inte alla gånger skapa det gränssnitt man önskar. Graden av konsekvens måste givetvis vägas mot om man kan uppnå andra fördelar som ökar användbarheten. Detta måste övervägas i varje specifikt fall och beroende på att mobiltelefonernas gränssnitt är begränsat kan man inte uppnå alla fördelar, utan måste ofta välja att utforma gränssnittet på ett sätt som ger störst användbarhet.

9.3.2 Kortkommandon

Vi har inte sett att det finns några möjligheter att lägga in kortkommandon i WAP-applikationer som tekniken ser ut idag. I de flesta mobiltelefoner finns ofta möjlighet att använda kortkommandon för att snabbt utföra en funktion eller uppgift, t.ex ändra inställningar. Eftersom denna typ av kortkommandon finns tillgängliga skulle det vara önskvärt att man på liknande sätt skulle kunna använda sig av det i WAP-applikationer, och kanske kommer det att vara möjligt i framtiden. Vi tror att detta skulle vara mycket användbart eftersom användaren i de situationer han eller hon använder mobiltelefonen vill utföra uppgiften snabbt och har ofta inte möjlighet att ägna hela sin uppmärksamhet åt telefonen.

9.3.3 Feedback

Feedback till användaren gör att han eller hon får reda på programmets status och får vetskap om när exempelvis ett moment är slutfört eller om något har gått fel. I vår applikation använde vi feedback för att användaren skulle få en bekräftelse på om en tidrad hade sparats eller en vecka hade klarmarkerats. Se figur 9.4. I samband med användbarhetstesterna sa användarna att feedback var mycket viktig för att de skulle veta huruvida uppgiften de utförde var slutförd eller ej.



Figur 9.4: Bekräftelse på att veckan är klarmarkerad.

På Ericssons WAP-telefon R320s finns det en förloppsindikator som är i rörelse och på så sätt visar att applikationen är aktiv. Indikatorn talar om för användaren att en process håller på att utföras och att användaren måste vänta tills processen är slutförd. En visuell indikation till användaren om att någonting håller på att utföras gör ofta att användaren känner sig lugnare och får mer tålamod att vänta de sekunder det tar att slutföra processen. Men på grund av att storleken på mobiltelefonens display är begränsad, finns det inte heller möjlighet för en indikator att vara så stor och tydlig som man skulle önska. Det finns en risk att den blir så pass liten att det är svårt för användaren att avgöra om den är i rörelse eller inte.

På mobiltelefoner går det i ganska stor utsträckning att ge bra och tydlig feedback till användarna även om detta inte riktigt kan ske på samma sätt som på terminaler med stora displayer. Exempelvis tar ett meddelande till användaren upp hela mobiltelefonens skärm istället för att visas som en litet dialogfönster, vilket ofta är fallet på en traditionell PC.

9.3.4 Gruppering

För att gränssnitt ska vara lätt att använda och överblicka är det bra om innehållet grupperas efter en viss struktur, exempelvis enligt innehållsmässig likhet eller användningsområde. När vi utformade vår applikation såg vi snart att det inte fanns möjlighet att dela in skärmbildens innehåll i olika delar på grund av det begränsade utrymmet på mobiltelefonens display. Vi valde istället att utforma dialogen så att den utfördes i en sekvens där varje skärmbild var för sig visade det innehåll som hörde ihop. Vi ansåg att detta sätt att gruppera information var ett bra alternativ och främjade gränssnittets enkelhet.

Det negativa vi kunde se med att utforma en sekvens, på det sättet vi gjorde, var att det kunde uppstå problem med den totala överblickbarheten. Men om vi å andra sidan hade lagt all inmatning och information på en skärmbild hade den blivit alltför lång och det hade knappast heller ökat överblickbarheten.

Gruppering av information i gränssnitt för mobiltelefoner måste många gånger ske på ett lite annorlunda sätt jämfört med traditionella gränssnitt på grund av den lilla displayen.

9.3.5 Felhantering

För att förhindra att användaren matar in fel värden använde vi, i så stor utsträckning som möjligt, listor där användaren endast kunde välja giltiga alternativ och undvek att användaren själv skulle skriva in värden. Där inmatning ändå var nödvändig, utformade vi programmet så att det endast skulle tillåta att godkända värden t.ex numeriska värden, kunde matas in.

Ett annat sätt att förhindra fel var att utföra uppgifterna ”Tidrad” och ”Klarmarkera” i en sekvens, som bestod av enkla skärmbilder. Detta bidrog till att användaren inte i så stor utsträckning kunde trycka fel och komma vilse i applikationen.

Det finns goda möjligheter även för små gränssnitt att tillämpa bra felhantering även om det inte går att förhindra alla typer av fel. Om användaren t.ex lämnar applikationen innan han eller hon har slutfört ett moment kommer det att innebära att användarens inmatade uppgifter och valda värden inte sparas. Användaren får i vissa lägen ingen möjlighet att bekräfta att han eller hon verkligen vill avsluta programmet. Man får överlåta en del av felhanteringen till användarens sunda förnuft, eftersom telefonen har knappar som wap-designern inte har kontroll över.

Bra utformade felmeddelanden är också viktigt och hjälper användaren att förstå systemet bättre. I WAP-applikationer finns det inga hinder för att ge tydliga felmeddelanden men ett litet problem är att felmeddelanden täcker hela telefonens display och därmed även den skärmbild där felet uppstod, vilket kan leda till att användaren kan bli osäker på felets ursprung.

9.3.6 Gör det möjligt för användaren att ångra val

Den ångra-funktion som går att tillämpa i WAP-applikationer är ”tillbaka-knappen” som hela tiden kan användas för att gå tillbaka till föregående sida och ändra de val man gjort. Detta kan liknas vid webb-gränssnitt där det inte är möjligt att ångra den senaste handlingen, på annat sätt än att använda sig av ”tillbaka-knappen”.

9.3.7 Användarkontroll

Användare av dataprogram föredrar att inte helt och hållet bli styrda av programmet utan vill själva kunna avgöra vad de ska göra och om de vill avbryta något som de har påbörjat.

I WAP-applikationer finns det möjlighet att lägga in ”nödutgångar” för användarna om de vill ta sig från en skärmbild till en annan. Detta kan t.ex göras genom att placera länkar längst ned på sidorna, men man bör vara medveten om att för många länkar och för mycket text ibland kan vara ett störningsmoment genom att användaren får för mycket information på den lilla skärmen.

Vi valde i vår applikation att inte lägga in länkar till menysidan på varje skärmbild, eftersom vi inte ville ta upp mer utrymme på skärmen än nödvändigt. Men det finns möjlighet att utnyttja en funktion som innebär att en dold meny visas och man kan med hjälp av den förflytta sig till förvalda platser i applikationen.

9.3.8 Minimalt krav på användarens minne

Människan har svårt att komma ihåg mycket information på en gång och därför bör användarens minne inte belastas i onödan. Ett sätt att underlätta för användaren är att utforma skärmbilderna så att de hålls enkla samtidigt som det måste finnas med så mycket information så att användaren kan fatta ett beslut. Mobiltelefonernas lilla display kräver att skärmbilderna är enkla, men displayens lilla utrymme medför även en svårighet eftersom det vid vissa tillfällen kan vara svårt att visa all relevant information som användaren behöver ha tillgänglig.

Ett annat sätt att underlätta för användaren är att utforma gränssnittet efter ledorden ”igenkännande är bättre än ihågkomst”. För att en WAP-applikation ska vara lätt att använda bör den utformas på ett telefonnära sätt, det vill säga på ett sätt som påminner om mobiltelefonens egna funktioner. Detta gör att användaren vid användning av applikationen känner igen arbetsgången att utföra operationer på från den vanliga mobiltelefonanvändningen. Denna typ av igenkänning gör att användarna efter att ha använt en applikation en gång oftast känner igen sig nästa gång de använder applikationen och på så sätt kommer ihåg hur de ska utföra uppgifterna.

10. Slutsatser

Uppsatsen syfte var att svara på följande två frågor:

- Går det att på mobiltelefonernas små skärmar behålla de egenskaper som ett traditionellt användargränssnitt bör ha för att vara användbart?
- Vad bör man tänka på när man utformar ett användargränssnitt för mobiltelefoner så att det, ur användarnas synpunkt, blir överskådligt och lätt att navigera i?

10.1 Går det att på mobiltelefonernas små skärmar behålla de egenskaper som ett traditionellt användargränssnitt bör ha för att vara användbart?

Det går i ganska stor utsträckning att följa de grundläggande riktlinjer som finns för utformning av traditionella gränssnitt även vid utveckling av gränssnitt till mobiltelefoner. Gränssnittets utformning måste givetvis anpassas till mobiltelefonens begränsningar, som till exempel skärmens storlek och komplicerad inmatning av text. Att gränssnitten måste anpassas får ofta konsekvensen att man inte kan uppfylla alla krav som de traditionella riktlinjerna ställer på ett användbart gränssnitt. Kompromisser mellan vilka riktlinjer man väljer att uppfylla måste övervägas i varje specifikt fall, eftersom de är beroende av vilken typ av applikation som ska utvecklas.

Här nedan följer en sammanfattning av de traditionella riktlinjerna och på vilket sätt vi har sett att de kan appliceras på gränssnitt för mobiltelefoner:

- *Konsekvent utformning.* Det är viktigt med en konsekvent utformning av applikationer, t.ex. layout och ordval. Samma information bör konsekvent presenteras på samma ställe på alla skärmbilder vilket gör att användarna lättare hittar den information de söker. Vid design av gränssnitt för mobiltelefoner går det i hög grad att utforma gränssnittet på ett enhetligt sätt exempelvis vad gäller ordval, men i vissa fall kan det vara svårt att vara konsekvent i skärmbildernas utseende. Detta beror framförallt på tekniska begränsningar och skärmens brist på utrymme.
- *Kortkommandon.* Erfarna användare bör ha möjlighet att kunna utföra frekventa operationer med hjälp av kortkommandon och genvägar. Med den teknik som finns idag är det inte är möjligt att använda kortkommandon i WAP-applikationer.
- *Feedback.* Systemet bör alltid informera användaren om vad som händer genom att ge tydlig feedback. Feedback på mobiltelefoner kan ske i stor utsträckning och innebära bra och tydlig information till användarna även om detta inte riktigt kan visas på samma sätt som på terminaler med stora displayer.
- *Gruppering.* För att öka läsbarheten bör informationen på skärmen grupperas på ett väl genomtänkt sätt. Genom att framhäva relevanta element kan man göra det enklare för användaren att prioritera olika valmöjligheter. På en och samma mobiltelefondisplay är det svårt att gruppera olika typer av information eftersom den är så liten. Man kan istället lösa det genom att dela upp informationen och placera de delar som hör ihop på olika skärmbilder.

- *Felhantering.* Det är viktigt att skapa applikationer där användarens handlingar inte medför att allvarliga fel uppstår. Det går i hög grad att tillämpa bra felhantering i WAP-applikationer och det görs i princip på samma sätt som i webb-applikationer.
- *Ångra.* Utförda operationer och handlingar bör i så stor utsträckning som möjligt gå att ångra. I mobiltelefonen, på samma sätt som på webben, finns det inga andra möjligheter att ångra utförda handlingar än genom att använda ”tillbaka-knappen”.
- *Användarkontroll.* Användare vill inte styras av datasystem utan känna att de har kontroll över händelserna. Det händer ibland att användare väljer fel funktioner och det kräver tydligt markerade utvägar så att användaren kan ta sig tillbaka till sin ursprungsposition. Denna typ av användarkontroll på mobiltelefoner kan skapas genom länkar, men det är inte i alla situationer säkert att det bidrar till ökad användbarhet eftersom länkar tar upp plats på den lilla skärmen.
- *Minimalt krav på användarens minne.* God design belastar inte användarens minne i onödan. Informationen bör därför göras synlig så att användaren inte behöver komma ihåg den från en skärmbild till en annan. Att avgöra hur mycket information som ska visas åt gången i en mobiltelefonapplikation kan vara svårt eftersom gränssnittets innehåll bör vara enkelt utformat samtidigt som det ska vara innehållsrikt och tillgodose användaren med tillräcklig information. Ett annat sätt att minska belastningen på användarens minne är att utforma gränssnittet efter ledorden ”igenkänning är bättre än ihågkomst”. En WAP-applikation blir lättare att använda om den utformas på ett sätt som påminner om mobiltelefonens egna funktioner, vilket gör att användaren känner igen sättet att utföra operationer på från den vanliga mobiltelefonanvändningen.

10.2 Vad bör man tänka på när man utformar ett användargränssnitt för mobiltelefoner så att det, ur användarnas synpunkt, blir överskådligt och lätt att navigera i?

Vi har kommit fram till att man bör tänka på följande när man utformar applikationer för mobiltelefoner:

- Applikationen bör anpassas efter vilken typ av mobiltelefon som den ska användas på. Detta beror delvis på att olika telefoner är utformade på olika sätt vad gäller tangenter och vad man kan utföra med dem, vilket påverkar hur intuitivt programmet blir att använda. En annan anledning är att olika telefoner tolkar WML-koden på olika sätt och det resulterar i att gränssnittets utseende inte ser likadant ut i alla terminaler. Man får anpassa sig till hårdvarans tekniska begränsningar, vilket kan hämma användbarheten.
- På grund av skärmens begränsade utrymme bör man använda tydliga rubriker och titlar som beskriver skärmbildens innehåll. Rubrikerna hjälper användaren att navigera i applikationen genom att de fungerar som en påminnelse om var någonstans användaren befinner sig i applikationen.

- I de fall man har möjlighet att påverka den information som ska presenteras på skärmen, bör ordens längd anpassas efter displayen så att rader och ord inte bryts så ofta. Detta kan åstadkommas genom att använda korta och informativa ord.
- Om en text blir så lång att den inte ryms på en skärmbild måste man som utvecklare vara medveten om att användaren kanske aldrig kommer att upptäcka den del av texten som inte är synlig. Idag finns det inget bra sätt att, med hjälp av WML, indikera att texten är längre än vad som ryms på skärmen. I den mobiltelefon, en Ericsson R320s, som vi har testat finns det i den inbyggda funktionaliteten indikatorer som visar när en textmassa inte ryms på en skärmbild. Liknande indikatorer skulle vara till stor nytta även i WAP-applikationer.
- Att mata in text i mobiltelefonen är ganska komplicerat och man bör därför så långt det är möjligt undvika att användaren själv matar in text. I de situationer där det är genomförbart bör man istället använda listor där användaren kan välja bland ett antal olika alternativ. Detta förhindrar också att användaren matar in felaktiga tecken. Ett annat sätt att underlätta för användaren är att använda förvalda värden som ofta kan minska antalet knapptryckningar. Detta kan uppnås genom att ge användaren möjlighet använda PC:n för att sätta upp en användarprofil eller att applikationen kommer ihåg vad användaren gjorde för val förra gången.

Vid inmatningen av lösenord behöver inte tecknen döljas eftersom skärmen är liten och användaren i de flesta fall står bortvänd från andra personer. Om tecknen är dolda är det svårt för användaren att kontrollera att rätt tecken matas in och man bör inte göra saker och ting svårare för användaren än nödvändigt.

- Så få knapptryckningar som möjligt är viktigt för att underlätta för användaren och för att han eller hon ska uppfatta att applikationen är lätt att använda. För att uppnå användbarhet genom så få knapptryckningar som möjligt fick vi i vårt fall göra avkall på gränssnittets konsekventa utformning. Liknande kompromisser tror vi att man då och då tvingas göra och det måste övervägas i varje enskilt fall.

10.3 Reflektioner

I dagens läge är det i första hand teknikintresserade personer och andra entusiaster som är villiga att använda sig av mobiltelefonen för att surfa på de Internetsidor som är optimerade för WAP. Framförallt beror detta på att det finns få WAP-tjänster som har ett verkligt värde för användaren, att överföringshastigheten av data är låg och att kostnaden för att vara uppkopplad via mobiltelefonen fortfarande är hög.

Med den kommande GPRS-tekniken (General Packet Radio Service) kommer WAP att kunna utnyttjas mer effektivt. I dagens läge används gsm-näten för dataöverföring med en hastighet på 9,6 kbit/s.⁷⁸ GPRS gör det möjligt att överföra data snabbare med hastigheter runt 40 - 50 kbit/s vilket ungefär är lika snabbt som ett vanligt modem.⁷⁹ En annan fördel med GPRS är att man ständigt är uppkopplad till mobilnätet och inte behöver koppla upp sig som fallet är nu. Nästa steg i utvecklingen är UMTS (Universal

⁷⁸ Olofsson, K. (2000). *Framtiden är trådlös*

⁷⁹ Jakobsson, H. (2000). *Wap ger dig ständig uppgraderingsångest*

mobile telecommunication system), den tredje generationens mobiltelenät, som förväntas ersätta gsm-näten. UMTS-näten är så gott som enbart inriktade på mobil datakom. Överföringshastigheten blir 386 kbit/s i hela näten och 2 Mbit/s i nära anslutning till vissa basstationer, vilket kommer skapa helt nya möjligheter för utveckling av nya, snabba tjänster.

Eftersom den tekniska utvecklingen kommer att innebära ökade möjligheter för mobila applikationer, kommer det att fortsätta att vara viktigt att utforma användbara gränssnitt. I nuläget kan WAP-applikationer användas på diverse olika terminaler med varierande skärmstorlekar. Vi har sett att de nuvarande mobiltelefonernas små skärmar gör det svårt att utforma ett tillfredsställande användargränssnitt även om alla föreskrivna riktlinjer efterföljs. Eftersom det framförallt är mobiltelefonens skärm som sätter begränsningarna för att skapa ett användbart gränssnitt, är det troligen först när utrymmet på skärmen ökar till en storlek som kan jämföras med en handdators skärm, som det finns möjlighet att göra riktigt användbara gränssnitt enligt de traditionella riktlinjerna. Därför skulle det vara intressant att i fortsatta studier undersöka hur användargränssnitt för handdatorer kan göras så användbara som möjligt.

Mobiltelefonens storlek gör det möjligt för användarna att ständigt ha den med sig och därmed ha tillgång till information oberoende av tid och rum. Vi tror att detta gör att användarna kan ha överseende med att användargränssnittet har vissa brister vad gäller utseende och layout. Man kan däremot diskutera om användarna verkligen kan vara mobila om terminalernas användargränssnitt ser ut som på en stationär dator med grafiskt gränssnitt. Ett komplement till detta skulle vara andra alternativa gränssnitt såsom röst-input m.m., vilket också skulle vara ett intressant ämne för vidare studier.

11. Referenser

Böcker

Allwood, C M. (1998). *Människa-datorinteraktion - Ett psykologiskt perspektiv*. Andra upplagan. Studentlitteratur, Lund.

Andersen, E S. (1994). *Systemutveckling – principer, metoder och tekniker*. Andra upplagan. Studentlitteratur, Lund.

Andersson, J. (1990). *Cognitive Psychology and its implications*. Freeman and Company, New York.

Ashcraft, M H. (1994). *Human Memory and Cognition*. Harper Collins College, New York.

Dahlbom, B., Mathiassen, L. (1993). *Computer in context*. Blackwell Publishers Inc, Oxford.

Dumas, J S., Redish, J C. (1994). *A practical guide to usability testing*. Ablex publishing corporation, Norwood, New Jersey.

Johansson, L. (1994). *Analysen är grunden för ett lyckat utvecklingsprojekt*

Lansdale, M W., Ormerod, T C. (1995). *Understanding Interfaces: A Handbook of Human-Computer Dialogue*. Academic Press, London.

Lif, M. (1998). *Adding usability, Methods for modelling, User interface Design and Evaluation*. Eklunds Grafiska, Uppsala.

Lynch, P J., Horton, S. (1999). *Web Style Guide, Basic Design Principles for Creating Web Sites.*. Yale University Press, New Haven and London.

Nielsen, J. (1993). *Usability Engineering*. Academic Press Inc, London.

Shneiderman, B. (1997). *Designing the User Interface*. Addison-Wesley.

Spool, J., DeAngelo, T., Scanlon, T., Schroeder, W., Snyder, C. (1999). *Web site usability - a designers guide*. Harcourt Publishers Ltd.

Sundgren, B. (1992). *Databasorienterad systemutveckling*. Studentlitteratur, Lund.

Forskningsrapporter

Peeter, K., Wingstedt, U. (1992). *Standarder för grafiska användargränssnitt*. Rapport nr 19, Svenska institutet för Systemutveckling

Artiklar

- Alm, M. *Systemutveckling*, Computer Sweden nr 14 (1992).
- Eriksson, M. *Människa-dator interaktion*, Computer Sweden nr 46 (1997).
- Geijer, E. *Fem personer räcker för att testa din webbplats*, Computer Sweden nr 32 (2000).
- Guimaraes, T., Saraph, J V. *The role of prototyping in executive decision systems*, Information & Management nr 21 (1991).
- Jakobsson, H. *Wap ger dig ständig uppgraderingsångest*, Computer Sweden nr 3 (2000).
- Nordlin, B. *Så här fungerar tekniken*, Computer sweden, nr 92 (1999).
- Olofsson, K. *Framtiden är trådlös*, Computer sweden, nr 22 (2000).
- Shoval, P., Pilskin, N. *Structured Prototyping: Integrating Prototyping into Structured System Development*, Information & Management nr 14 (1988).
- Stedt, P. *WAP pratar alla om?* Nollett nr 2, (1999).
- Zirn, T. *Vänta inte på gprs – wap räcker för många tjänster*, Computer Sweden nr 23 (2000).

Webbdokument

- AU-system Radio. (1999). *WAP White Paper*. [WWW dokument].
URL <http://www.ausys.se/servlet/PSP/PSP?iNodeID=1653&iLastNodeID=303>
- Dahlbom, B., Ljungberg, F. (2000). *Mobile Informatics*, [WWW dokument]. URL http://www.informatik.gu.se/~dahlbom/papers/papers_1999/mobile_informatics.html
- Ericsson. (2000). *Design guidelines for WAP services*. [WWW dokument].
URL <http://www.ericsson.com>
- Instone, K. (1997). *Usability engineering for the web*. [WWW dokument].
URL <http://www.w3j.com/5/s3.instone.html>
- Kristoffersen, S., Ljungberg, F. (1999). *"Making Place" to Make IT Work: Empirical Explorations of HCI for Mobile CSCW*. [WWW dokument].
URL <http://www.viktoria.informatik.gu.se/groups/mi3/results/papers/group99.pdf>

- Nielsen, J. (1997). *The difference between Web design and GUI design*. [WWW dokument]. URL <http://www.useit.com/alertbox/9795a.html>
- Nielsen, J. (1998). *What is "Usability"?* [WWW dokument]. URL <http://www.zdnet.com/filters/printerfriendly/0,6061,2137671-84,00.html>
- Nokia. (1999). *Getting Started, Nokia WAP Toolkit, Version 1.2*. [WWW dokument]. URL <http://www.forum.nokia.com>
- Nokia. (2000). *Service Developer's Guide*. [WWW dokument]. URL <http://www.nokia.com>
- Nokia. (1999). *Wireless Application Protocol – The Corporate Perspective*. [WWW dokument]. URL <http://www.nokia.com/corporate/wap>
- Oestreicher, L. (1994). *Vad är MDI?*. [WWW dokument]. URL <http://www.stimdi.se/public/mdi.html>
- Passani, L. (1999). *WAP and ASP - Part 1*. [WWW dokument]. URL <http://www.asptoday.com/articles/19991115.htm>
- Phone.com. (1999). *Application Usability Guidelines from the Application Development Team*. [WWW dokument]. URL <http://www.phone.com/>
- Phone.com. (1999). *Developing User Friendly Applications*. [WWW dokument]. URL <http://www.phone.com/>
- Shubin, H. (1997). *Interaction Design, Navigation in Web Applications*. [WWW dokument]. URL <http://www.user.com/webapps/webapps.htm>
- Story, D. (1999). *Usability Checklist for Site Developers*. [WWW dokument]. URL <http://webreview.com/pub/1999/10/15/usability/index.html>
- wap.com. (2000) *What is WAP ?* [WWW dokument]. URL <http://www.wap.com/share/osas/cache/artid50005.html>

Konferenshandlingar

- Wiråker, P. Aspiro. (2000). *Life in motion*. Mobilt Internet. Göteborg