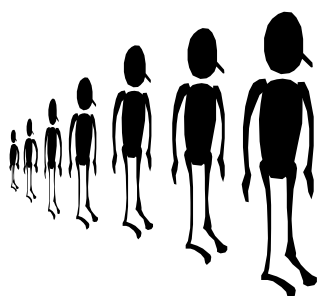


Gränssnittsdesign - med användaren i fokus



Abstrakt

Att designa ett användargränssnitt är inte särskilt svårt. Utmaningen ligger i att skapa ett gränssnitt som tar hänsyn till människans förmåga och beteende – egenskaper vi varken kan påverka eller förändra. För att utreda hur ett gränssnitt skall utformas för att bli användarvänligt har vi utgått ifrån *användarens* betydelse vid gränssnittsdesign. Vi har studerat människa-datorinteraktion (MDI), användbarhet och riktlinjer för design, samt genomfört två undersökningar. Utifrån detta har vi fört en omfattande diskussion där teorierna jämförts med undersökningsresultaten. Det viktigaste som framkom var att det är mycket viktigt att involvera slutanvändaren under designprocessen då det är användarens uppfattning om gränssnittet som slutligen avgör hur bra det är. Vidare kom vi fram till att kunskaper om MDI främst bör finnas hos de som utformar riktlinjer och inte nödvändigtvis hos varje systemutvecklare. Vi kunde även konstatera att principer och riktlinjer oftast är generella och måste anpassas efter varje unik situation.

Examensarbete 10 p, ADB-programmet 80 p
Vårterminen 1998

Författare: Josefine Ersson och Linda Kärrander
Handledare: Birgitta Ahlbom (Informatik) och Staffan Olsson (IFS AB)

1	INLEDNING.....	5
1.1	BAKGRUND OCH PROBLEMFÖRMULERING.....	5
1.2	AVGRÄNSSNINGAR.....	5
1.3	SYFTE	6
1.4	ANGREPPSSÄTT.....	6
1.5	RAPPORTSTRUKTUR	6
2	MÄNNISKA-DATORINTERAKTION	7
2.1	ALLMÄNT OM MÄNNISKA-DATORINTERAKTION	7
2.2	ALLMÄNT OM KOGNITIONSPSYKOLOGI	8
2.2.1	Minne	9
2.2.2	Gestaltpsykologi	12
2.2.3	Hågkomst	12
2.2.4	Metaforiskt tänkande.....	13
2.2.5	Mentala modeller.....	14
2.3	SAMMANFATTNING	15
3	ANVÄNDARNA.....	17
3.1	ANVÄNDARKATEGORIER.....	17
3.2	UTFÖRANDET AV EN UPPGIFT	18
3.3	ANVÄNDBARHET.....	21
3.3.1	Allmänt om användbarhet	21
3.3.2	Krav för god användbarhet.....	21
3.4	ANVÄNDBARHETSANALYS.....	23
3.4.1	Allmänt om användbarhetsanalys.....	23
3.4.2	Analysmetoder	24
3.5	SAMMANFATTNING	25
4	PRINCIPER OCH RIKTLINJER.....	27
4.1	DEFINITION AV DESIGN	27
4.2	DESIGNPROCESSEN	27
4.3	PRINCIPER OCH RIKTLINJER FÖR GRAFISKA GRÄNSSNITT	29
4.3.1	Allmänt om principer och riktlinjer.....	29
4.3.2	Åtta gyllene regler för dialogdesign	30
4.3.3	Hur bör ett gränssnitt se ut?.....	30
4.4	WEBB-GRÄNSSNITT.....	34
4.4.1	Framträdande skillnader mellan webb- och windows-gränssnitt	34
4.4.2	Viktigt vid webb-design	34
4.5	SAMMANFATTNING	35
5	UNDERSÖKNINGAR.....	37
5.1	FÄLTSTUDIE - GULLSPÅNG	37
5.1.1	Metod	37
5.1.2	Sammanfattning av intervjusvar.....	38
5.1.3	Övriga reflektioner	39
5.1.4	Slutsatser	40
5.2	LABORATORIEUTVÄRDERING FÖRSTAGÅNGSANVÄNDARE.....	40
5.2.1	Testscenario.....	40
5.2.2	Undersökningsresultat.....	41
5.2.3	Övriga reflektioner	47
5.2.4	Slutsatser	47
5.3	ÖVRIGA UNDERSÖKNINGAR	47
5.3.1	Saab.....	48
5.3.2	Ericsson.....	48
5.3.3	Slutsats	49
6	DISKUSSION.....	51
6.1	MÄNNISKA-DATORINTERAKTION	51

6.2	ANVÄNDARNA	53
6.3	PRINCIPER OCH RIKTLINJER.....	55
6.4	SLUTSATSER	56
7	VIDARE FRÅGOR.....	59
8	KÄLLFÖRTECKNING	61
8.1	BÖCKER	61
8.2	BIBLIOGRAFIER	61
8.2.1	Rapporter.....	61
8.2.2	Opublicerat material	62
8.2.3	Artiklar	62
8.2.4	Intervjuer	63
8.2.5	Testpersoner	63
BILAGOR		
	BILAGA 1A - GULLSPÅNG	
	BILAGA 1B - FRÅGEFORMULÄR	
	BILAGA 2A - FÖRSTAGÅNGSANVÄNDARE	
1.	BILAGA 2B - INSTRUKTIONSFOMULÄR	

1 Inledning

1.1 Bakgrund och problemformulering

IFS AB (Industrial and Financial Systems) är ett företag som utvecklar, implementerar och underhåller affärssystem för medelstora och stora företag, främst inom den tillverkande industrin. Deras produkt, IFS Applications, består av ett 40-tal fristående moduler med grafiskt användargränssnitt som kan sättas samman helt enligt kundens behov och önskemål.

Det har skett en kraftig utveckling på mjukvarumarknaden där webb-teknologiska lösningar får en alltmer framskridande roll. IFS har börjat tillmötesgå detta och utvecklat webb-gränssnitt för fem av modulerna som finns i IFS Applications. Dessa är:

- IFS/Time & Attendance
- IFS/Project Reporting
- IFS/Time Management
- IFS/Shop Floor Reporting
- IFS/Travel Expenses

Vi har under vår praktiktid på IFS arbetat med att utforma webb-gränssnittet för IFS/Travel Expenses. De största svårigheterna vi stötte på gällde frågor kring design och vi saknade teorier och riktlinjer om hur ett gränssnitt bör utformas. IFS gav oss fria händer att utföra designen, utan några som helst restriktioner. Därför vill vi i detta arbete beskriva teorier som underbygger ett bra gränssnitt, med utgångspunkt från användaren. Vi ställer oss därmed följande frågor:

Vilken betydelse har användaren vid utveckling av gränssnitt?
Hur uppfattar och bearbetar användaren information?
Hur kan användaren involveras i designprocessen?
Vilka riktlinjer bör man som designer följa?
Hur bör en skärmbild disponeras på för användaren bästa sätt?
Hur testar man användbarhet?

1.2 Avgränsningar

Följande områden kommer inte att behandlas i uppsatsen:

- Tekniker och verktyg som används för utveckling av gränssnitt.
- Hur riktlinjer för design tekniskt skall genomföras.
- In- och utmatningsenheter.
- Hjälpresurser och utbildning av användare.
- Teckenbaserade gränssnitt.

- Teorier angående funktionalitet.

Vi förlitar oss på tidigare gjorda psykologiska undersökningar inom människa-datorinteraktion (MDI) och har därför inte genomfört några egna sådana undersökningar.

1.3 Syfte

Syftet med denna uppsats är att utreda vilken betydelse användaren har vid gränssnittsdesign, samt att undersöka om utvalda teorier inom detta område stämmer i praktiken.

1.4 Angreppssätt

För att uppnå vårt syfte har vi studerat material inom området. Materialet har bestått utav ett antal böcker, internetdokument, tidningsartiklar och opublicerat material inom områden som människa-datorinteraktion, användbarhet och riktlinjer kring design. Vi har också genomfört två egna undersökningar, där den ena gjordes i form av intervjuer med användare av IFS/Time & Attendance webb-klient och den andra var ett användbarhetstest med förstagångs användare av IFS/Time & Attendance webb- och windows-klient. För korrekta uppgifter om IFS har vi samtalat med anställda på företaget.

1.5 Rapportstruktur

Uppsatsen består av sex avsnitt: Inledning, Människa-datorinteraktion, Användarna, Principer och riktlinjer, Undersökningar och Diskussion.

Inledningen beskriver bakgrund och angreppssätt, samt de avgränsningar som gjorts. MDI-kapitlet (Människa-datorinteraktion) behandlar kognitionspsykologi, där bl a människans förmåga att behandla och organisera information tas upp. Följande kapitel (Användarna) beskriver människan som slutanvändare, samt ett systems användbarhet och olika metoder för att testa denna. Därefter (i avsnittet Principer och riktlinjer) redogörs för några av de principer och riktlinjer som finns för gränssnittsdesign. Kapitlet om undersökningarna (Undersökningar) innehåller redogörelser för dels en fältstudie med användare av IFS/Time & Attendance webb-klient på Gullspång AB, och dels en laboratorieutvärdering där både webb- och windows-klienten för IFS/Time & Attendance testas av förstagångs användare. Här redogörs också kortfattat för två utvärderingar av webb-klienterna för IFS/Time & Attendance respektive IFS/Project Reporting gjorda av Saab och Ericsson. I diskussionsavsnittet (Diskussion) jämför vi om våra undersökningsresultat stämmer överens med teorierna och drar slutsatser utifrån detta. Till uppsatsen biläggs skärmutskrift av gränssnitten för IFS/Time & Attendance, samt intervjufrågor och uppgiftsinstruktioner.

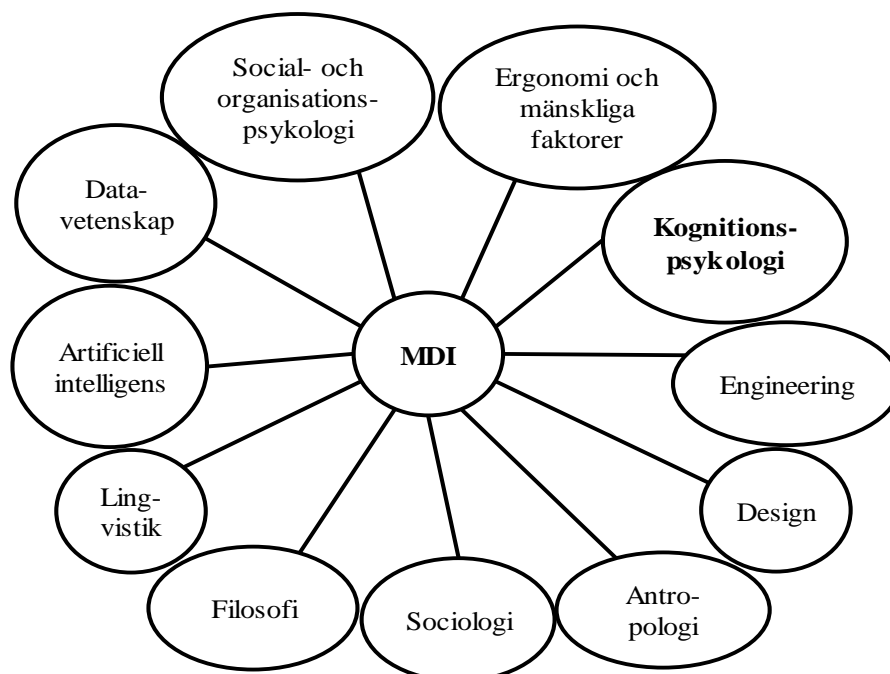
2 Människa-datorinteraktion

Följande kapitel handlar om människa-datorinteraktion, och undersöker vilka kognitionspsykologiska faktorer som påverkar människans sätt att hantera information. Tyngdpunkten ligger på att beskriva människans minne och hur hjärnan tar emot och organiserar information.

2.1 Allmänt om människa-datorinteraktion¹

Människa-datorinteraktion (MDI) är ett relativt nytt (ungefär 40 år gammalt), tvärvetenskapligt² ämne som förklarar samspelet mellan människa och dator. MDI kan ge vägledning om hur datorsystem bör konstrueras för att uppfattas så användarvänliga som möjligt av slutanvändaren. Alla människor är olika; personlighet, tidigare erfarenhet, kultur, kognitiv- och fysisk förmåga är några områden som skiljer individer åt. Det betyder att det inte finns en typisk användare, utan att gränssnitt måste designas så att det passar flera olika individer. Designarbetet kan underlättas om man under systemutvecklingen tar del utav vetenskapliga undersökningar om hur människan tänker, löser problem och organiserar information.

Flera vetenskaper, bl a datavetenskap, kognitionspsykologi, ergonomi och organisations- och beteendevetenskap, beaktas inom människa-datorinteraktion, se *Figur 1*. Vi har valt att koncentrera oss på kognitionspsykologi.



Figur 1 Vetenskaper inom MDI (Källa: J. Preece m fl, 1994)

¹ J. Preece m fl (1994)

² Innefattar flera andra vetenskaper

2.2 Allmänt om kognitionspsykologi³

Kognitionspsykologerna har försökt karakterisera människors informationsprocesser efter dess kapacitet och begränsningar. Vi kan urskilja två olika slags kognitiva processer, den *automatiska* och den *sekventiella*. Den automatiska, undermedvetna nivån tar emot och behandlar flera miljoner sinnesimpulser varje sekund. Det innebär att vi kan behandla ett stort antal informationsprocesser samtidigt utan att behöva tänka medvetet på det. Exempelvis kan en bilförare registrera både vägkrökning, motorljud, ratt rörelser m m utan att tänka aktivt på det. Den sekventiella, medvetna nivån gör att vi bl a kan göra bedömningar och hantera osäker information. Denna analytiska och logiska förmåga är stor, men kapaciteten är starkt begränsad. Det innebär att vi på den sekventiella nivån endast kan hantera en informationsprocess i taget. Vi är således bara medvetna om en liten del av alla de informationsprocesser som ständigt behandlas i våra hjärnor.

En informationsprocess kan bestå utav *perception*⁴, minne, inlärning, tänkande och problemlösning. Följande delar ingår i en informationsprocess:

- Inkodning av information till någon typ av intern representation.
- Jämföra representationen med tidigare lagrade representationer i hjärnan.
- Besluta om en passande respons.
- Organisera responsen och nödvändiga åtgärder.

En närmare beskrivning av, och exempel på, människans informationsprocesser, ges i avsnitt 3.2, *Utförandet av en uppgift*.

Kognitiv psykologi kan hjälpa till att förbättra ett systems design genom att:

- Förutsäga vad användare kan, och inte kan, förväntas göra.
- Identifiera och förklara det som orsakar de problem som användaren stöter på.
- Erbjuder modellverktyg och metoder för att bygga gränssnitt som är lätta att använda.

Följande områden tas upp inom området kognitiv psykologi:

- **Minne**
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.
- **Gestaltpsykologi**
Gestaltpsykologins lagar förklarar hur människan tenderar att gruppera information som ser ut på ett visst sätt. Dessa lagar kan vara till stor nytta då ett bra organiserat gränssnitt är enklare för hjärnan att hantera.

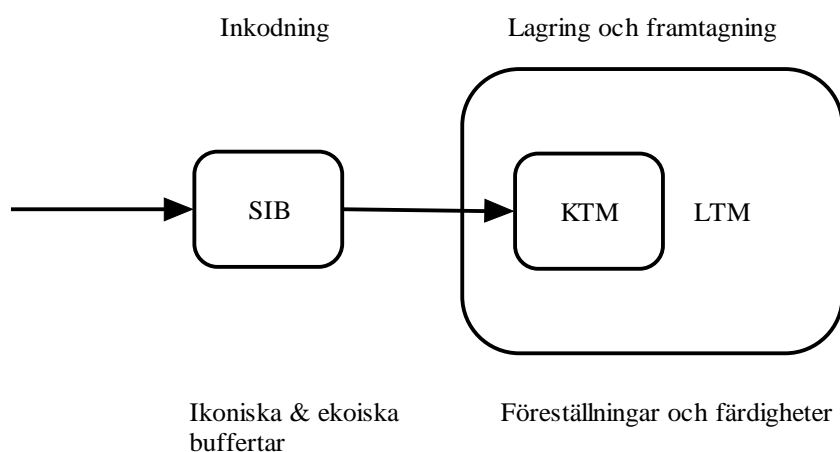
³ J. Preece m fl (1994)

⁴ Kan beskrivas som inkodning, varseblivning eller sinnenas intryck.

- Hågkomst
En människa är bättre på att känna igen än att komma ihåg.
- Metaforiskt tänkande
Ett metaforiskt tänkande bygger på tidigare erfarenheter.
- Mentala modeller
Människan använder sig av mentala modeller vid interaktion med exempelvis ett system. Hjärnan har redan innan interaktionen format en bild över hur systemet skall se ut, där mallen kan ha varit ett annat system eller någon annans intryck.

2.2.1 Minne⁵

Människans minne kan delas in i tre olika delar; sensoriska informationsbuffertar (SIB), korttidsminne (KTM), även kallat arbetsminne, och långtidsminne (LTM). *Figur 2* beskriver enkelt sambandet mellan de olika minnesdelarna.



Figur 2 Människans minne (Källa: Jämför C. M. Allwood, 1991)

Att lära sig något, och senare komma ihåg det, kan delas in i tre olika faser; *inkodning*, *lagring* och *framtagning*. Inkodning, även kallat perception, sker när information från omvärlden genom t ex syn och hörsel når minnet. Lagring innebär att information lagras i minnet, antingen i form av en föreställning eller en färdighet. Framtagning sker när information i minnet plockas fram och aktiveras. SIB är främst aktuella under inkodningsfasen medan KTM och LTM involveras under alla tre faserna.

2.2.1.1 Sensoriska informationsbuffertar

Inkodning uppstår när information från omvärlden når våra sinnen, och således också våra sensoriska informationsbuffertar. Här stannar informationen helt otolkad innan den, efter att informationen relaterats till individens förkunskaper i långtidsminnet,

⁵ C. M. Allwood (1991), E. Nygren (1991), J. Preece m fl (1994)

förs över till korttidsminnet. Allt som hamnar i SIB placeras inte automatiskt i korttidsminnet. Korttidsminnets begränsade storlek och kapacitet gör att hjärnan omedvetet selekterar ut information från SIB, som anses mest relevant för tillfället, och placerar den i korttidsminnet.

Man skiljer på det *ikoniska* minnet, ögats sensoriska informationsbuffert, och det *ekoiska* minnet, örats sensoriska informationsbuffert. Det ikoniska minnet kan hålla information upp till en fjärdedels sekund. Det ekoiska minnet kan hålla information lite längre, mellan 2 till 4 sekunder. Information som redan är inhämtad av det ikoniska minnet kan störas av att ny information når SIB. Människan hinner helt enkelt inte uppfatta och kategorisera informationen innan den försvinner från det ikoniska minnet. Därför bör man inte visa mycket information under alltför kort tid på en bildskärm.

2.2.1.2 Korttidsminnet

Flera forskare menar att korttidsminnet är en del av långtidsminnet. Inom kognitionspsykologin definieras korttidsminnet som ett minne där information av olika slag, föreställningar, känslor, målstrukturer m m har en hög grad av aktivering. I korttidsminnet måste informationen lagras mellan 5 till 20 sekunder för att den skall "fastna" i långtidsminnet. Korttidsminnet är begränsat och kan bara innehålla 7 (plus/minus 2) enheter samtidigt. En enhet kan beskrivas som en siffra, en bokstav eller ett ord. Det är lättare att komma ihåg fler enheter om de är organiserade på något sätt, som att bokstäver ingår i ett ord eller om ord ingår i en mening. Genom att bunta ihop grupper av data kan korttidsminnet innehålla upp till 100 enheter. Exempelvis är det enklare att komma ihåg ett telefonnummer om siffrorna buntas ihop. Telefonnumret 427587 blir lättare att minnas som 42 75 87. När vi bildar enheter gäller *gestaltlagarna*, se vidare avsnitt 2.2.2, *Gestaltpsykologi*. Vi för samman objekt som liknar eller är grupperade med varandra och vi tillför information om någonting saknas.

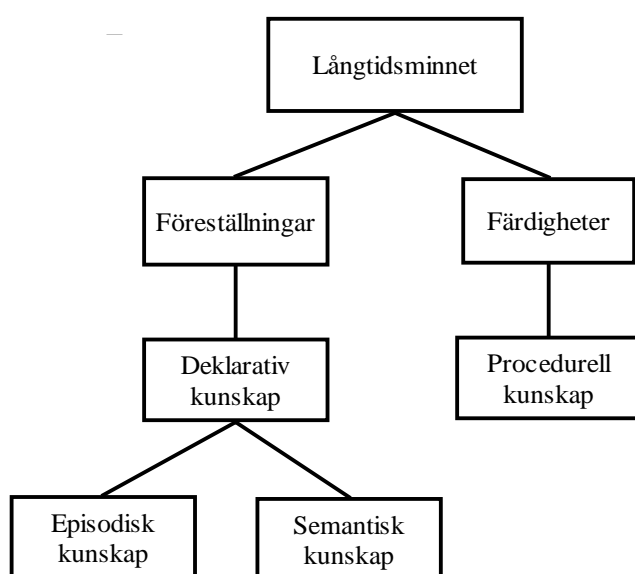
Informationen som finns i korttidsminnet kan försvinna på två olika sätt, antingen genom att nya enheter bildas eller att enheternas aktivitet inte förnyas. När nya enheter bildas störs den gamla informationen som finns lagrad i korttidsminnet. Därför bör man inte visa för mycket information på bildskärmen samtidigt. Om enheterna inte förnyas kommer informationen att försvinna ur minnet. Jämför det med leken där man skall titta på och memorera ett antal föremål, för att sedan vända sig om medan någon plockar bort en/ett av dessa. När du sedan vänder dig om skall du säga vilket föremål som saknas.

I många fall kan det dock vara nödvändigt att på skärmbilden visa mycket information på en gång. Användarna skall t ex inte behöva växla bild mitt i ett beslut då undersökningar har visat att de då riskerar att fatta sämre beslut, arbeta långsammare och uppleva arbetet som mer stressat.

2.2.1.3 Långtidsminnet

Vårt långtidsminne har möjlighet att lagra hur mycket information som helst. När vi upplever att vi glömt någonting har informationen alltså inte försvunnit, den är bara organiserad på ett mindre bra sätt. Jämför det med att tappa bort en bok inne i ett stort bibliotek; boken finns där men den går inte att hitta.

Långtidsminnet kan beskrivas som en förvaringsplats som innehåller mentala representationer av två olika typer, *föreställningar* och *färdigheter*. Dessa består i sin tur av *deklarativa* respektive *procedurella* kunskaper, se *Figur 3*. Deklarativa kunskaper kan beskrivas som att ”veta att” medan procedurella kunskaper kan jämföras med att ”veta hur”.



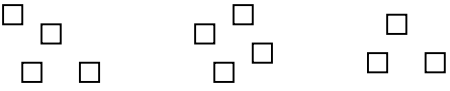
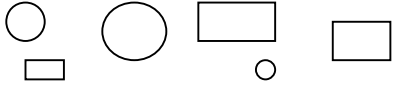
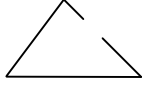
Figur 3 Långtidsminnet

Deklarativa kunskaper kan liknas med mentala representationer av verkligheten. Exempelvis associerar vi till Rom om vi hör orden Italien och huvudstad. Deklarativa kunskaper kan delas in i *episodiska* och *semantiska* kunskaper där episodisk kunskap kan vara ett minne av en specifik upplevelse medan semantiska kunskaper består av ett antal upplevelser som länkats samman till en förståelse. Dessa kunskaper kan lagras i olika sensoriska format, t ex visuella, auditiva och verbala. Olika människor är olika duktiga på att hantera och lagra information i skilda format. Vissa är bättre på att komma ihåg någonting de sett medan andra är bättre på att komma ihåg någonting de hört. Deklarativa kunskaper kan ha olika hög grad av aktivering. När aktiveringsnivån inte är tillräckligt hög för att innehållet skall nå medvetandet kallas det för att innehållet är ”primat”. Pratar vi t ex om begreppen gran och björk, associeras dessa ord till kategorin träd. Om vi i nästa mening nämner ordet bok, ligger det närmast att anta att kategorin fortfarande är träd och inte bibliotek.

Procedurell kunskap skapas genom färdigheter. Det är lättare att minnas någonting om man genererat informationen själv. Det är t ex lättare att komma ihåg namnet på en fil om man själv skapat namnet. En färdighet kan t ex vara motorisk eller språklig. Att knyta ett skosnöre kan betraktas som en motorisk färdighet. Du tänker inte i stegen att först göra en ögla, sedan dra igenom skosnöret och slutligen knyta en rosett, utan förfarandet sker på automatisk nivå, se avsnitt 2.2, *Allmänt om kognitionspsykologi*.

2.2.2 Gestaltpsykologi⁶

Av människans fem sinnen är synen den viktigaste att ta hänsyn till vid människa-datorinteraktion. Teorier kring hur synen uppfattar strukturer och kategoriserar dessa kallas *gestaltpsykologi*. Gestaltpsykologin har resulterat i en antal lagar som är intressanta att ta del av vid design av gränssnitt. Lagarna rör begrepp som närhet, storlek, slutenhet, symmetri, enkelhet, likhet och upprepning. Följande illustrationer (*Figur 4*) beskriver några av gestaltpsykologins lagar:

	Objekt som ligger nära varandra uppfattas som att de hör ihop.
	Objekt som liknar varandra utseendemässigt uppfattas som att de hör ihop, form- eller färgmässigt.
	Människan har en förmåga att fylla i information som saknas.

Figur 4 Gestaltlagar(Källa: U. Wiss, 1996)

Människas tidigare erfarenheter, *erfarenhetslagen*, spelar också roll i tolkningen av vad som ses. De påverkar vårt sätt att organisera information och en bra organisation bidrar, som tidigare nämnts, till att informationen blir lättare att minnas.

Av ovanstående gestaltlagar dras slutsatsen att människan har en förmåga att gruppera och tillföra information till objekt som ser ut eller är placerade på ett visst sätt.

2.2.3 Hågkomst⁷

En bild säger mer än tusen ord. Vi är mycket bättre på att känna igen, *igenkänning*, än att komma ihåg, *återgivning*. Skillnaden mellan igenkänning och återgivning kan exemplifieras av följande:

⁶ C. M. Allwood (1991), J. Löwgren (1993), U. Wiss (1996)

⁷ C. M. Allwood (1991)

En kvinna skall identifiera en misstänkt brottsling. Vittnet får frågan om hon kan identifiera brottslingen bland tio andra personer. Det kallas igenkänning. Om vittnet däremot får frågan om hon kan beskriva brottslingen med egna ord, kallas det återgivning.

Ovanstående exempel kan jämföras med menyer (igenkänning) och kortkommandon (återgivning), se *Figur 5*. Det är svårare att komma ihåg ett kortkommando än att välja en händelse i exempelvis en meny eller kombobox. Nybörjare och sällananvändare föredrar att använda menyer, medan experter och superanvändare hellre använder sig av kortkommandon.

Populate	F2
Query...	F3
Clear	F4
Save	F12
<hr/>	
New	F5
Duplicate	F6
Remove	F7
<hr/>	
List of values...	F8
Zoom...	Shift+F9
Connected Objects...	F9
History...	Ctrl+F9

Figur 5 Meny med kortkommandon (Källa: IFS/Time & Attendance, 1998)

2.2.4 Metaforiskt tänkande⁸

Att tänka metaforiskt innebär att använda sig av tidigare erfarenheter för att lösa problem. Tänkandet är alltså i hög grad beroende av individens färdigheter, se avsnitt 2.2.1.3, *Långtidsminnet*. Det finns flera olika typer av metaforer, bl a *verbala* metaforer och *gränssnittsmetaforer*. En verbal metafor kan vara en muntligt eller skriftligt instruktion som kan hjälpa en användare att förstå ett system. En gränssnittsmetafor baserar sig på fysiska metaforer. Ett exempel på detta är att man kan låta en ikon ha samma innebörd i flera program. Därför designas många system så att de skall likna Windows standard, MS/Windows Style Guide, så mycket som möjligt.

Metaforer hjälper användaren att:

- Aktualisera redan kända problemlösningsmetoder, strategier och handlingsmönster.
- Föra över denna kunskap och erfarenhet till den nya situation som utgörs av det datorbaserade systemet och som därigenom känns välbekant och naturligt.

⁸ C. M. Allwood (1991), J. Preece m fl (1994),

- Utveckla och stärka en fungerande mental modell av objekt och funktioner i systemet.

Det kunskapsområde som den använda kunskapen kommer ifrån kallas *ursprungsdomän* och det kunskapsområde som kunskapen används i kallas *tillämpningsdomän*. Det finns flera olika slags ursprungsdomäner och vilka som används beror på vilken typ av program det gäller och vilket uppgiftsområde programmet gäller. Följande domäner är att räkna med i de flesta program:

- Andra delar av det program användaren använder.
- Andra program som användaren har bekantskap med.
- Konventioner i naturligt språk.
- Konventioner i samband med ansikte-mot-ansikte-kommunikation.

De ursprungsdomäner användaren utnyttjar är av stor praktisk betydelse. Dessa påverkar en användare med bristfälliga datorkunskaper när denne skall lära sig ett nytt program.

Att använda sig av ursprungsdomäner kan resultera i både positiva- och negativa effekter, där de positiva effekterna i regel är fler. Positiva effekter underlättar inläringen av programmet för användaren och minskar antalet begångna fel. En positiv effekt kan t ex uppstå när en användare skall övergå från skrivmaskinsteknik till ett ordbehandlingsprogram. Då tangenterna är placerade på exakt samma sätt kan samma fingerteknik användas när man skriver. Negativa effekter försvårar inläringen och får användaren att göra fler fel. Det kan leda till att användaren tappar förtroendet för systemet. Negativa effekter kan exempelvis uppstå när en användare skapar förväntningar av ett nytt system utifrån kunskaper från det tidigare körda systemet. Användaren förväntar sig då att det nya systemet skall likna det gamla och blir missnöjd när design och funktionalitet inte stämmer överens med det gamla systemet.

Ett metaforiskt tänkande bygger alltså på individens färdigheter och därmed skiljer sig effekternas betydelse från användare till användare. Därför berörs en expert mindre av negativa effekter än vad en nybörjare gör.

2.2.5 Mentala modeller⁹

När användarens färdighetskunskap är ofullständig kan denne ta hjälp av sin förståelsekunskap för att lösa en uppgift. Denna förståelse kallas ofta för *mental modell* och är uppbyggd via erfarenheter. Mentala modeller är ofullständiga och analoga. De korrigeras sällan eftersom de ofta möjliggör rimliga men felaktiga förklaringar. Modellen är en abstrakt representation som människor använder för att strukturera sina erfarenheter om sig själva, andra, omgivningen och saker de interagerar med. Ett exempel på en mental modell kan vara en världskarta. En europé har inte samma mentala modell över världskartan som en amerikan har. En amerikansk världskarta centrerar Amerika medan en europeisk världskarta har Europa i mitten.

⁹ D. Norman (1988), J. Preece m fl (1994)

En typ av mental modell är den *konceptuella* modellen som beskriver olika sätt ett datorsystem förstås av olika människor, exempelvis slutanvändare, systemutvecklare och forskare. En konceptuell modell är konkret och försöker ge en komplett och konsistent bild av systemet.

När människan interagerar med ett system formar hon interna mentala modeller. För att användaren skall bilda sig en korrekt mental modell av ett system bör man som systemutvecklare tänka på att:

- Ge feedback till användaren.
- Vara konsistent.
- Utgå ifrån kända metaforer.
- Göra osynliga delar och processer synliga.

2.3 Sammanfattning

Sammanfattningsvis kan konstateras att människans minnesfunktion är mycket komplicerad. Trots att långtidsminnet kan lagra enorma mängder information begränsar korttidsminnet och de sensoriska informationsbuffertarna hjärnan från att ta emot för mycket information på en gång. En skärmbild skall därför inte innehålla för mycket information på en gång. En användare bör heller inte tvingas att komma ihåg information vid växling mellan två skärmbilder.

Korttidsminnets kapacitet att ta emot information ökar om informationen är organiserad på något sätt, enligt t ex gestaltlagarna. Likaså blir information lättare för långtidsminnet att minnas om den är strukturerad och exempelvis stödjer användarens metaforer och mentala modeller.

3 Användarna

I nedanstående kapitel beskrivs användarna och vilka kategorier de kan delas in i utifrån olika faktorer. Vidare berörs användbarhetens betydelse för användarnas produktivitet och olika sätt att testa användbarheten på.

3.1 Användarkategorier¹⁰

Vid utveckling av gränssnitt är det viktigt att känna till systemets framtida användare och den miljö som systemet till slut skall hamna i. Användare är den eller de personer som kommer att använda systemet och interagera med gränssnittet. All design bör börja med att man skapar en förståelse för de tilltänkta användarnas olika profiler utifrån ålder, kön, fysisk förmåga, utbildning, kultur, motivation, mål och personlighet.

Användarna kan grovt delas in i tre olika grupper:

- **Noviser**
Förstagångs användare eller nybörjare har ofta liten eller ingen kunskap om hur systemet skall användas och behöver en logisk förståelse för vad som skall göras. Vid design av gränssnitt som är avsedda för denna typ av användare är det viktigt att designern använder en sådan vokabulär att användarna känner igen termer och uttryck. Designen bör inte erbjuda för många valmöjligheter, utan användaren skall kunna utföra ett fåtal enkla uppgifter för att bli säkrare och stärka sitt självförtroende. En manual med steg-för-steg-instruktioner kan vara ett lämpligt hjälpmedel för denna användarkategori.
- **Medelgoda användare**
Dessa användare vet hur programmet fungerar, men lägger inte kortkommandon etc på minnet. Medelgoda användare vill gärna använda sig av menyer, då det är lättare att känna igen än att komma ihåg, se avsnitt 2.2.3, *Hågkomst*.
- **Experter**
Expertanvändaren kan programmet väl och använder sig i stor utsträckning av kortkommandon för att utföra sina arbetsuppgifter snabbare. Dessa användare vill ha snabba svarstider, kort och snabb feedback och i stor utsträckning kunna använda sig av förkortningar.

Att göra en design som passar var och en av dessa användarkategorier är i regel inga problem. Vanligen består dock slutanvändarna av en blandning av dessa kategorier, vilket gör designarbetet betydligt svårare. Dessutom kan användarna delas in i ytterligare grupper, beroende på hur ofta de använder programmet. Man talar bl a om

¹⁰ B. Shneiderman (1992)

sällan- och *superanvändare*, där sällananvändaren kanske använder programmet en gång i veckan medan superanvändaren lägger ner flera timmar per dag. Detta faktum försvårar specificeringen av användarna ännu mer.

Det är alltså viktigt att ta hänsyn till de skillnader som finns mellan användarna. Vid internationalisering av användargränssnitt bör man också känna till vilka företeelser som kan tänkas skilja sig från varandra beroende på t ex kultur. Bokstäver, siffror, specialtecken, tid- och datumformat, valuta, namn, titlar, personnummer, stavning, grammatik, knappar, färger, ikoner och metaforer är ett axplock av saker att tänka på.

För att, som utvecklare, lyckas riktigt bra måste man vara näst intill expert inom det område som användarna arbetar och kommer att använda systemet i. En arbetsplats innehåller många faktorer som har inflytande på de anställda och hur de utför sina arbetsuppgifter. Därför är det stor skillnad på att utveckla ett gränssnitt som t ex skall användas i kontorsmiljö än att utveckla ett för en verkstadsindustri.

3.2 Utförandet av en uppgift¹¹

Forskning inom psykologin visar att det finns en del saker som människor har gemensamt, när det t ex gäller hur vi utför uppgifter. D. Norman har gjort en MDI-modell som visar vad som händer när man utför en uppgift. Norman delar in själva utförandet i två delar; att utföra aktionen i sig och sedan evaluera resultatet.

Utförandet har sju steg, se *Figur 6*.

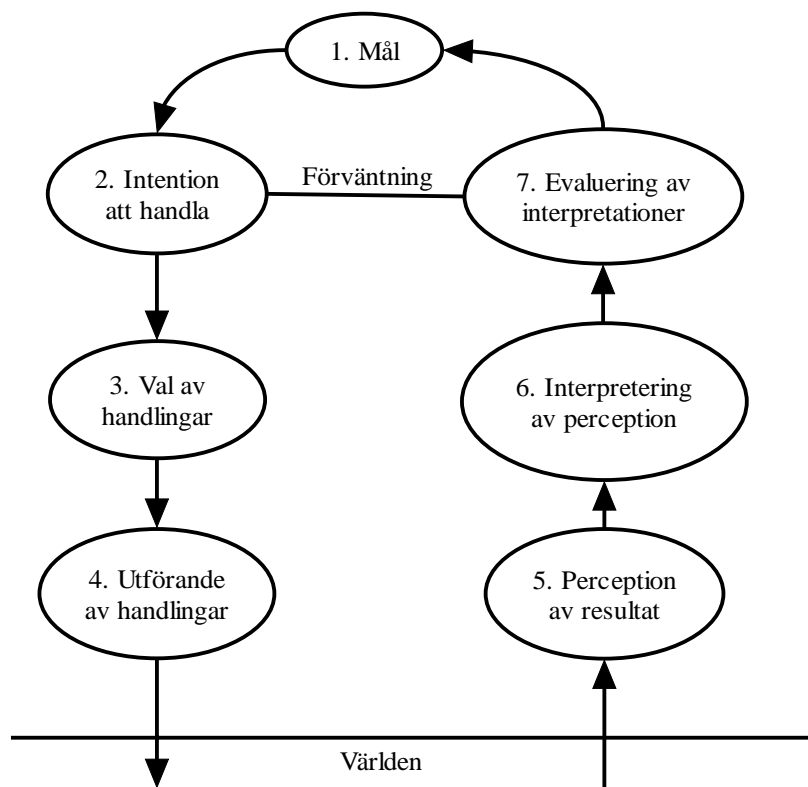
1. Utforma målet. Målet är basbegreppet för vad användaren vill ha gjort.
2. Översätta målet till påståenden om vad som behöver göras. Dessa påståenden är intentionerna att handla.
3. Välja handlingsalternativ.
4. Utföra handlingarna.
5. Observera resultatet (jämför inkodning, avsnitt 2.2.1.1, *Sensoriska informationsbuffertar*)
6. Förstå, *interpretera*, observationen.
7. Evaluera resultatet genom att jämföra förståelsen med den förväntade händelsen.

Om resultatet inte överensstämmer med förväntningarna repeteras denna ”cykel” tills användaren når det önskade målet eller helt enkelt ger upp i frustration.

Detta kan beskrivas med ett enkelt exempel. Ett stycke text skall redigeras så att de översta raderna hamnar underst i stycket. Detta är målet (steg 1). Man kommer fram till att raderna behöver klippas ut och sedan klistras in på önskad plats. Detta kan göras genom att markera texten och antingen välja *cut* i menyn eller under höger musknapp eller med hjälp av kortkommandot Ctr-X. Sedan kan texten klistras in på motsvarande vis (med hjälp av *paste* i någon av menyerna eller Ctr-V). Detta är handlingsalternativen (steg 2). Ett eller flera alternativ väljs (steg 3) varefter handlingarna utförs (steg 4). Man blir medveten om resultatet, d v s systemets status

¹¹ C. von Dorrien och L. Magnusson (1997), J. Löwgren (1993), D. Norman (1988), B. Shneiderman (1992)

(steg 5) och tolkar och förstår detta (steg 6). Slutligen jämförs resultatet med förväntningarna (steg 7).



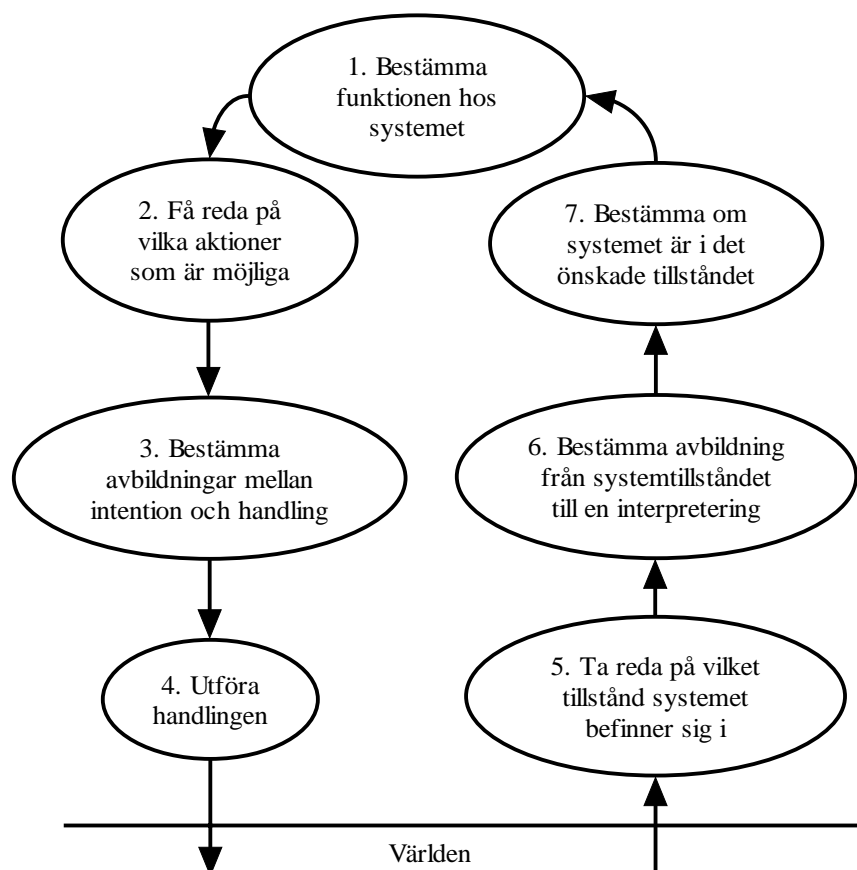
Figur 6 Sju steg i att utföra en uppgift (Källa: D. Norman, 1988)

Under utförandet måste användaren tänka på hur uppgiften skall fullgöras. Detta utförande kan vara *rutin-* eller *beslutsbaserat*. Ett rutinutförande baseras främst på minnen och används i situationer där man känner igen sig väl. Detta utförande sker alltså på den automatiska nivån, vilket gör att användaren samtidigt kan tänka på sekventiell nivå, se avsnitt 2.2, *Allmänt om kognitionspsykologi*. Man tänker t ex inte medvetet på att man måste ta tag i musen och trycka ner vänster musknapp för att markera texten.

Ett beslutsbaserat utförande används också i familjära situationer, men sker inte automatiskt. Ett schema över situationen hämtas från minnet och om det inte finns något sådant schema så skapar minnet ett. Detta skapande kräver dock övning och det räcker inte med att få instruktioner. Om man t ex i exemplet ovan inte kände till kortkommandot Ctr-X och någon talar om att det kan användas för att klippa ut textstycket, så lär man sig det inte förrän man själv har gjort det några gånger, jämför procedurrell kunskap, avsnitt 2.2.1.3, *Långtidsminnet*.

Denna sju-steps-struktur kan användas som riktlinjer för grundläggande design, se *Figur 7*, genom att fungera som en checklista för hur väl systemet motsvarar ställda krav på utförande- och evalueringsnivå.

Man kan fråga sig hur lätt det är att:



Figur 7 Sju steg som riktlinjer (Källa: C. von Dorrien och L. Magnusson, 1997)

Dessa frågor resulterar i några principer för god design:

- *Synbarhet*: Systemstatus och handlingsalternativ måste vara synliga för användaren.
- *God konceptuell modell*: Designern skall förse användaren med en bra konceptuell modell, konsistent¹² presentation av operationer och resultat, samt en sammanhängande, konsistent systembild. En god konceptuell modell låter användaren förutse effekterna av sina handlingar.
- *Lättförståeliga avbildningar*: Det skall vara möjligt att bestämma relationer mellan handlingar och resultat, mellan kontroller och deras effekter, samt mellan systemstatusen och vad som är synligt. Gränssnittet måste alltså innehålla bra kartläggning som visar relationerna mellan de olika stegen.
- *Feedback*: Användaren skall få fullständig och kontinuerlig feedback om resultat av handlingar.

¹² Konsistent innebär här att ett visst utseende eller en viss rörelse alltid betyder samma sak.

3.3 Användbarhet¹³

3.3.1 Allmänt om användbarhet

Med ett systems användbarhet menas att de personer som använder systemet skall kunna utföra sina uppgifter på ett snabbt och lätt sätt. Denna definition vilar på fyra punkter:

- Användbarhet innebär fokusering på användarna.
- Användarna använder systemet för att bli mer produktiva.
- Användarna är upptagna människor som försöker utföra olika slags uppgifter.
- Användarna bestämmer när ett system är lätt att använda.

Användarna bedömer hur lätt ett system är att lära sig och använda utifrån hur lång tid det tar att utföra sina uppgifter, hur många steg de måste gå igenom, samt hur väl de kan förutse resultatet av en handling.

Användbarhet är ett nyckelkoncept inom MDI-området och det är viktigt att tidigt i utvecklingen sätta upp *användbarhetsmål*. Poängen med användbarhetsmål är att utifrån användarens synpunkter på systemet formulera mätbara kriterier. Om dessa mål ingick i kravspecifikationen¹⁴ skulle de tas på lika stort allvar som övriga krav. Problemet är att kravspecifikationen ofta görs på ett tidigt stadium, innan projektet ens har startat. Då har man ännu inte hunnit göra användar- och arbetsstudier.

3.3.2 Krav för god användbarhet

Forskning visar att önskvärdheten av en programegenskap beror på hur denna egenskap fungerar tillsammans med programmets och användningssituationens övriga egenskaper. Den främsta anledningen till att man använder sig av datorprogram är att man på ett lättare sätt skall kunna lösa en viss typ av uppgifter. I de flesta fall handlar det även om att med hjälp av datorprogrammet öka *produktiviteten* genom att höja kvaliteten på resultatet. För att nå dessa mål räcker det inte med att tillhandahålla en tillräcklig *funktionalitet*, d v s att funktionerna i programmet klarar av relevanta uppgifter. På senare år har man blivit alltmer medveten om att programmen dessutom måste ha en god *användbarhet* för att kunna uppnå bästa möjliga resultat, se *Figur 8*.

Figur 8 visar att ett programs användbarhet bestäms utifrån olika egenskaper i användningssituationen. Tre viktiga komponenter i detta sammanhang är *användaracceptans*, *användarkompetens* och *användarvänlighet*.

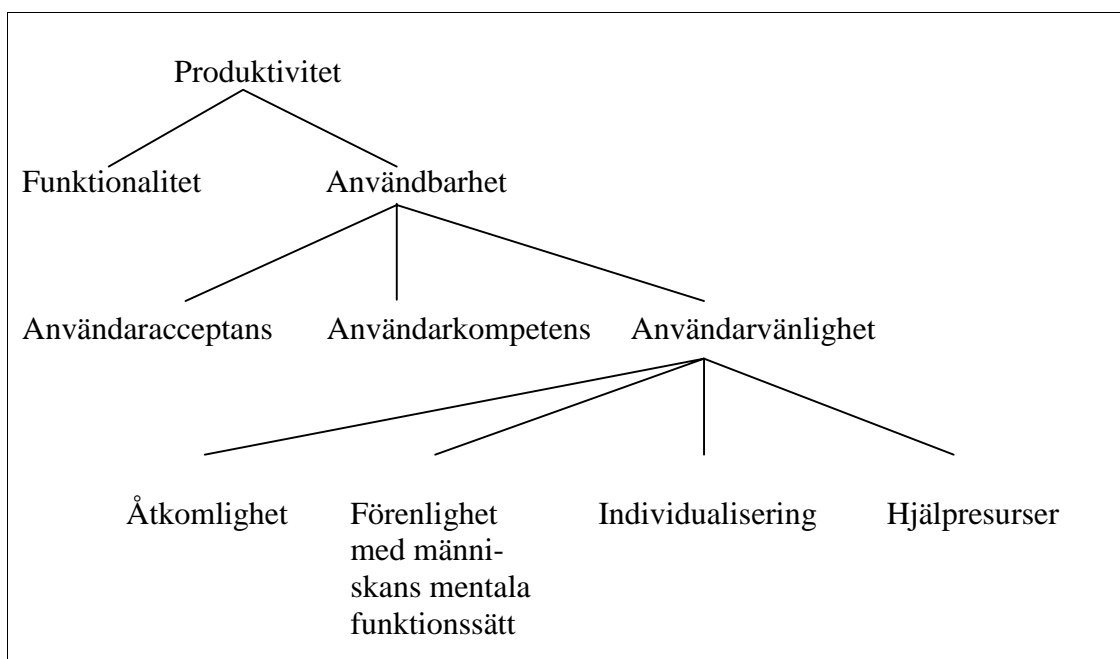
3.3.2.1 Användaracceptans

Med användaracceptans menas att det är viktigt att användarna har en positiv inställning till, och är motiverade att använda sig av, programmet. Om användar-

¹³ C.M. Allwood (1991), J.S. Dumas m fl (1993), B. Göransson (1997), J. Preece m fl (1994)

¹⁴ Ett dokument med kundens krav och önskemål.

acceptansen saknas finns det en risk att användaren inte lär sig att använda programmet eller att programmet används oengagerat och på fel sätt.



Figur 8 Produktivitet beror på funktionalitet och olika former av användbarhet (Källa: C.M. Allwood, 1991)

3.3.2.2 Användarkompetens

Användarkompetens innebär att användaren har tillräcklig förståelse för och tillräckliga färdigheter att använda programmet på ett effektivt sätt. För att uppnå detta krävs naturligtvis utbildning i systemet. Dagens utbildningar som ges i syfte att underlätta användningen av applikationsprogram är bra, men skulle kunna förbättras.

3.3.2.3 Användarvänlighet

Användarvänlighet kan delas in i fyra aspekter. En självklar sådan är *åtkomst*, som innebär att användaren måste ha tillgång till systemet för att kunna använda det. En annan aspekt är att de krav programmet ställer på användaren är *förenliga* och därmed stämmer överens med användarens sätt att fungera mentalt. Ett exempel är att programmet inte bör kräva att användaren skall hålla mer information aktuell i minnet än vad som är möjligt vid ett visst ögonblick. För att bedöma om ett program uppfyller detta krav behöver man studera människans mentala funktionssätt, vilket beskrivs närmare under avsnitt 2.2.1, *Minne*. En tredje aspekt i sammanhanget är *individualisering*. Eftersom alla användare är olika är ett program som ger stöd åt flera olika typer av användare mer användarvänligt. Dessutom är chansen då större att det passar den enskilde användaren. Den sista aspekten avser *hjälpresurser*. Det är viktigt att användaren har tillgång till hjälp i form av t ex andra människor, handböcker, manualer och programmets hjälpfunktion.

3.4 Användbarhetsanalys¹⁵

3.4.1 Allmänt om användbarhetsanalys

Ju större förståelse designern har för användarnas behov, desto bättre blir designen. Det bästa sättet att utreda vad användarna vill ha och vilka problem de stöter på är att göra en *användbarhetsanalys*. Detta begrepp beskrivs i litteraturen även under benämningar som *evaluering*, *test* och *utvärdering*. Vi har här valt att använda oss av begreppet användbarhetsanalys, nedan även kallat *analys*.

Användbarhetsanalys är en central del av användarcentrerad systemdesign. Utan någon form av analys är det omöjligt att veta om designen uppfyller användarnas behov, och hur väl det passar in i den fysiska, sociala och organisatoriska miljö där den skall användas. En generell regel är att vilken slags användbarhetsanalys som helst är bättre än ingen alls. Några saker som är viktiga att tänka på, oavsett vilken typ av analys det handlar om, är:

- Utmärkande drag hos användarna (eller de tänkta användarna) som deltar i analysen, t ex erfarenhet, ålder, kön, fysiska och psykiska kännetecken.
- Vilka slags uppgifter (eller tänkta uppgifter) användarna kommer utföra. Det kan vara specifika uppgifter som bestäms och kontrolleras av en utvärderare eller uppgifter som användaren själv väljer.
- Vilken miljö analysen utförs i. Det kan vara allt ifrån en kontrollerad laboratorie-situation till en naturlig arbetsmiljö, se vidare avsnitt 3.4.2, *Analysmetoder*.
- Vilken slags produkt, *artefakt*, som analyseras. Det kan vara några enkla utkast, en prototyp eller en färdigutvecklad produkt.

Användbarhetsanalysen kan utföras i princip när som helst under utvecklingsfasen. Tidigt i processen handlar det om att försöka förutsäga användbarheten och kolla upp designteamets förståelse för användarnas krav, genom att se hur de använder redan existerande system och pröva idéer snabbt och informellt. Det är alltså av stor vikt att redan från början ha koll på användarnas behov, så att de i slutändan är nöjda och inte saknar det eventuella föregående systemet. I designfasens senare del försöker man identifiera användarnas svårigheter för att kunna möta deras behov och förbättra systemet.

Det finns flera anledningar att göra en användbarhetsanalys. En anledning kan vara att man vill förstå verkligheten. Exempelvis kan man ställa sig frågorna hur användarna använder teknologin på arbetsplatsen, och om designen kan förbättras så att den passar miljön bättre. En annan anledning kan vara att man har två olika designförslag och vill jämföra dessa med varandra. Vid design mot ett visst mål vill man veta om designen är tillräckligt bra, om målet kan nås, liksom att man vill kolla om produkten är anpassad till en viss standard.

¹⁵ J. Preece m fl (1993), J. Preece m fl (1994)

3.4.2 Analysmetoder¹⁶

Att sätta upp mål för användbarhet för ett system som skall utvecklas innebär, som vi tidigare nämnt, att man skriver ner mätbara krav på användbarheten. Det är dock inte alltid möjligt att sätta upp dessa mål. Problemet med att mäta användbarhet är att det lättast görs först när systemet är installerat. Det finns flera modeller som avser att sätta upp mål för användbarhet. En i dessa sammanhang känd modell som fokuserar på att mäta användbarhet i ett tidigare skede är den s k *REAL-modellen*. Det menas att användbarhet är resultatet av *Relevans*, *Effektivitet*, *Attityd* och *InLärning* (*Relevance*, *Efficiency*, *Attitude*, *Learnability*).

- Relevans avser hur väl systemet täcker användarens behov.
- Effektivitet visar hur effektivt användaren kan lösa sina uppgifter med hjälp av systemet.
- Med attityd menas användarens subjektiva känsla för systemet.
- Inlärning visar hur lätt och hur snabbt användaren kan lära sig systemet, och dessutom behålla dessa kunskaper under en längre tid.

För att mäta dessa egenskaper kan systemets användbarhet studeras genom en rad olika tester. Det talas ibland om *formativ* och *summativ* evaluering, där formativ evaluering är ett test som görs under utvecklingsfasen, och summativ evaluering innebär att man testat ett färdigt system för att se hur bra det är. Vid gränssnittsdesign är formativ evaluering mest intressant, då designers under sitt arbete ofta behöver svar på om deras idéer verkligen är vad användarna behöver eller vill ha. Det är dessutom viktigt att utveckla eventuell användardokumentation eller hjälpsystem parallellt med systemutvecklingen, för att kunna testa dessa samtidigt.

En användbarhetsutvärdering kan t ex utföras på följande tre sätt.

- *Expertutvärdering*
Testerna utförs av användargränssnittsexperten med erfarenhet av gränssnittsutveckling. Här deltar inte användarna, men det är viktigt att experterna känner till användarnas arbetssituation, bakgrund, kunskaper etc. Experterna granskar system, program och manualer, samt provkör om det är möjligt. Systemet utvärderas mot riktlinjer för användargränssnitt, s k *styleguides*, *checklistor* och regler för användbarhet. En expertutvärdering är ett snabbt och billigt sätt att hitta grundläggande användbarhetsproblem, även om det inte ger full relevans till användarens uppgiftslösande. Resultatet blir en rapport med defekter och förändringsförslag.
- *Laboratorieutvärdering*
En grupp som kan anses motsvara de tilltänkta användarna och/eller flera ”ovana” användare får testköra systemet. De får först fylla i ett formulär som skall ge en användarbakgrund. Utvärderaren, som i detta fall bara är en passiv åskådare som samlar in data, studerar hur användarna löser uppgifterna, vilka problem de ställs inför etc. De data som samlas in under testkörningen analyseras först efteråt. Med en laboratorieutvärdering hoppas man kunna upptäcka både funktionella problem och elementära inlärningsproblem. En

¹⁶ B. Göransson och T. Sandbäck (1997), J. Löwgren (1993)

brist ligger i att utvärderingen sker i en annan miljö än vad användarna normalt är vana vid. Hur resultatet ser ut beror på den användarkategori man valt. Analysen resulterar i rapporter och förändringsförslag.

- *Fältstudier*

Denna utvärdering, som utförs på redan implementerade system, är mest verklighetsanknuten och ger möjlighet att utvärdera systemet i en realistisk användarmiljö. Precis som i laboratorieutvärderingen får användaren svara på bakgrundsfrågor som skall ge en användarprofil, samt fylla i en enkät om hur denne upplever att systemet fungerar. Studien utförs hos användaren i dennes dagliga arbetsmiljö, där utvärderaren följer med i användarens arbete. Resultatet blir dels en rapport med defekter och förändringsförslag och dels statistik över hur systemet uppfattas och accepteras av användarna. Nackdelarna med denna utvärderingsmetod är att den kan bli väl kostsam, då man går in och stör användarna i deras arbete, och att den bara kan tillämpas på system som är i drift.

Vilken metod som är lämplig att använda beror dels på *målsättning* - vad man vill analysera (en arbetsstudie, ett designförslag, en detaljerad prototyp eller ett mer eller mindre färdigt system), dels på vilka *resurser* man har att tillgå i projektet, samt *ambitionsnivå*.

Vilka mått man använder vid mätningen beror på vilken slags analys det gäller. Skalor är lämpliga om man använder checklistor i undersökningen. Tidtagning och felräkning är mått som används vid t ex laboratorieutvärderingar. Relevans och attityd mäts bäst och enklast genom att man direkt frågar användarna vad de tycker. Även vid djupare studier av inläring kan direkta frågor vara användbara. För att mäta relevans och inläring är det viktigt att jämföra om den mentala modell användaren skapat över systemet stämmer med den modell användaren angivit i arbetsstudien. Om den inte stämmer får man antingen överväga att göra om arbetsstudien (då har man misslyckats ur relevanssynpunkt) eller så har designen inte lyckats förmedla den avsedda modellen (vilket beror på dålig inläring).

3.5 Sammanfattning

Användarna kan delas in i olika kategorier, t ex noviser, medelgoda användare och experter, beroende på *hur* de använder ett datorsystem. Det är viktigt att designern känner till användarnas arbetsmiljö, vilka kategorier de tillhör och är medveten om likheter och olikheter mellan individerna. Hur användbart ett system är bestäms bl a utifrån hur snabbt och lätt användaren kan utföra sina uppgifter med hjälp av systemet. För att uppnå god användbarhet krävs användaracceptans, användar-kompetens och användarvänlighet. För att få reda på hur väl designen uppfyller användarnas krav och behov görs någon form av användbarhetsanalys, exempelvis en fältstudie. Analysens syfte är att se om målen för användbarhet kan nås, samt att förstå vad som behöver ändras för att systemet skall bli bättre.

4 Principer och riktlinjer

Följande kapitel beskriver kortfattat hur en systemutveckling med hög användarmedverkan kan gå till. Kapitlet presenterar även riktlinjer och principer som kan tillämpas vid gränssnittsdesign, samt ger konkreta förslag på vad man bör tänka på vid utformandet av ett gränssnitt.

4.1 Definition av design

Design kan beskrivas som att skapa något nytt och användbart som tidigare inte existerat. Ofta används ordet design lite slarvigt för hela systemutvecklingen, vilket skulle göra alla systemutvecklare till användargränssnittsdesigners.

Användargränssnittsdesign kan enligt T. Sandbäck och B. Göransson (1997) definieras som;

Skapandet av en formell beskrivning av utseende och uppförande, baserad på delvis formella och delvis informella resultat av en analys.

4.2 Designprocessen¹⁷

En väl fungerade arbetsmiljö kan lätt bli förstörd av en illa genomtänkt datorisering, ett problem som kan undvikas genom användarcentrerad utvecklingsmetod. Därför bör användaren och dennes uppgifter stå i centrum vid utveckling av ett system. Den användarcentrerade utvecklingsmetoden bör kombineras med följande informationskällor:

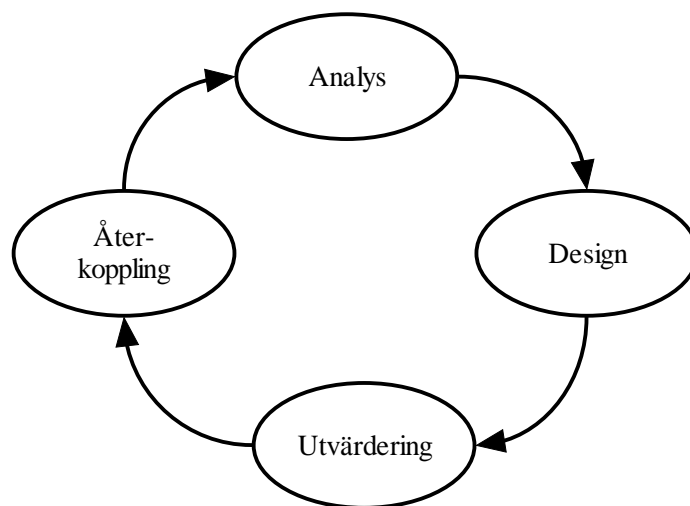
- Vetenskaplig kunskap, såsom kognitiv psykologi, datavetenskap, sociologi, filosofi, ergonomi etc, se **Figur 1**, s. 7.
- Etablerade tekniker för in- och utmatning, menyer, formulär, markörkontroller och andra hjälpmedel.
- Erfarenhet från andra utvecklingsprojekt och kunskap från andra system.

Ett användarvänligt system kan bara byggas om användaren involveras i alla steg av utvecklingen. Utvecklandet av användargränssnitt kan delas in i följande inledande faser, se även *Figur 9*:

- **Analys av verksamheten**
Den nuvarande verksamheten analyseras för att man skall komma fram till om ett nytt system innebär kostnads- och arbetsbesparingar. Vidare undersöks det tänkta systemets målgrupp, verksamhetens informationsbehov, samt användargruppens tillämpning av informationen.

¹⁷ B. Göransson (1997), J. Preece m fl (1994)

- **Analys av arbetsuppgifterna**
Användarens arbetsuppgifter analyseras och en bild av användarens föreställning om sina arbetsuppgifter, användarens mentala modell, tas fram och generaliseras i en metafor.
- **Prototypfas**
En prototyp av systemet designas utifrån analysen av arbetsuppgifterna.
- **Utvärderingsfas**
Prototypen utvärderas utifrån användarens förväntningar om hur systemet skall lösa dennes arbetsuppgifter.
- **Återkoppling**
Arbetet med att utforma gränssnittet fortsätter utifrån utvärderingsfasen.



Figur 9 Designprocessen (Källa: B. Göransson, 1997)

Följande problem är svårast att lösa under designarbetet:

- Att få information om användarnas behov och att förutse uppgiftens krav.
- Att utforma ett gränssnitt som är både lätt att lära och fungerar för experten.
- Att använda färger och fonter.

Ett dåligt gränssnitt kan medföra att användaren inte bryr sig om att använda alla delar av systemet. Om det dessutom tar längre tid än innan att genomföra en arbetsuppgift, ligger det nära till hands att användare går tillbaka till det sätt som förelåg det nya systemet.

4.3 Principer och riktlinjer för grafiska gränssnitt¹⁸

4.3.1 Allmänt om principer och riktlinjer

Skillnaden mellan principer och riktlinjer kan förklaras genom följande: En princip är ett mål, men talar inte om hur detta mål skall nås. En princip kan leda till flera riktlinjer som i sin tur är mer *specifika* mål.

En princip kan enligt J. S. Dumas m fl vara:

Be consistent in your choice of words, formats, graphics and procedure.

medan en riktlinje kan beskrivas som:

Be consistent in the way you have users leave every menu.

Det finns olika former av riktlinjer:

- *Högnivå* och generellt applicerbara designprinciper – används för att ge idéer och styra designen i rätt riktning.
- *Designregler* – används ibland för att instruera designern om hur man åstadkommer en design som är lämplig för det aktuella systemet.
- *Standards* – System bör anpassas till internationella, nationella och industriella standarder.

På liknande sätt kan man skilja på tre nivåer av regler och riktlinjer vid designarbetet.

- *Style Guides* som beskriver hur en speciell implementation är gjord och hur den skall användas. På denna nivå handlar det om objekt på skärmen. Exempel på detta är MS/Windows Style Guide.
- *Guidelines* är mer precisa i sin beskrivning av hur designen skall se ut och kan vara knutna till en speciell domän eller organisation, som t ex IFS Application GUI Design Guidelines 1.2.
- *Officiella standarder* som utgör ett slags fundament och klargör krav på en funktionell nivå. ISO 9241 är ett exempel på en sådan standard.

Dessa nivåer är avsedda att användas som komplement i designarbetet, och är i sig oanvändbara utan kunskap om design av användargränssnitt och användarens arbetsuppgifter. Riktlinjer och principer är ett hjälpmedel och en utgångspunkt som egentligen inte talar om hur en tillämpning skall formges. Problemet är att riktlinjerna är generella och inte anpassade för den specifika situationen. Ofta är riktlinjerna rent av omöjliga att följa i den aktuella designen eller i utvecklingsverktyget. Svårigheten i design ligger alltså inte i att hitta nya funktioner, utan att hitta de rätta funktionerna, dvs de som passar det aktuella gränssnittet. Det är därför viktigt att man som designer väljer ut och applicerar rätt riktlinjer vid rätt tillfälle och på ett intelligent och

¹⁸ B. Göransson och T. Sandbäck (1997), J. Löwgren (1993), J. Preece m fl (1993), B. Shneiderman (1992),

förnuftigt vis. Designarbetet skall helst utföras av experter inom området då lyckad design kräver djupare kunskaper kring människan och hennes beteende.

4.3.2 Åtta gyllene regler för dialogdesign¹⁹

Nedan presenteras ett urval av principer, riktlinjer och regler som man bör försöka tillämpa vid gränssnittsdesign. Dessa grundprinciper måste tolkas, förfinas och utökas för varje enskild miljö.

1. Sträva efter konsistens (likformighet). Samma sak görs på samma vis i likartade situationer. Använd identiska uttryck i menyer, hjälpfönster etc, t ex exit, end, quit – antingen eller!
2. Gör det möjligt för erfarna användare att använda sig av genvägar. Användaren blir efter ett tag säkrare och mer erfaren och vill då kunna använda sig av kortkommandon, reducera antalet interaktioner och ha snabbare svarstider.
3. Ge informativ feedback. Systemet skall ge feedback på alla handlingar som utförs.
4. Dela in dialogen i avslutande moment genom att ge feedback direkt, så att användaren verkligen förstår att en aktion är avslutad. Annars finns risken att användaren utför handlingen fler gånger än nödvändigt.
5. Erbjud en begriplig felhantering. Användaren skall inte kunna göra allvarliga fel. Vid "lindriga" fel skall lätthanterlig hjälp erbjudas, och användaren skall inte behöva göra om hela förloppet, utan bara den del som orsakade felet. Systemtillståndet skall bibehållas, eller så skall instruktioner ges om hur tillståndet återfås.
6. Tillåt användaren att ångra en handling. Det får användaren att känna sig "säkrare" om denne vet att fel kan åtgärdas.
7. Ge användaren kontroll och understöd dennes initiativ. Låt inte systemet överraska användaren.
8. Minska användarens belastning på korttidsminnet genom att skapa enkla skärmsidor.

Rekommendationerna kan tyckas självklara för den som skall utforma ett gränssnitt för ett givet system. Ändå stöter de flesta användare ständigt på tillämpningar som inte uppfyller dessa rekommendationer.

4.3.3 Hur bör ett gränssnitt se ut?²⁰

Ett gränssnitt är systemets ansikte utåt och skall vara tilltalande för de tilltänkta användarna. Följande avsnitt beskriver ingående några utvalda riktlinjer för utveckling av grafiska gränssnitt.

¹⁹ B. Shneiderman (1992)

²⁰ P. André m fl (1993), S. Lundhem, E. Nygren

4.3.3.1 Skärmdisposition

Ett stort problem vid gränssnittsdesign är att disponera skärmbilden på ett så effektivt sätt som möjligt. Användarna skall helst inte behöva scrolla eller byta skärmbild för att se information, samtidigt som de kan bli förvirrade om för mycket information visas på en gång. Om man måste byta skärmbild mitt i ett beslut och hålla kvar information i minnet till den nya skärmbilden, har undersökningar visat att man fattar sämre beslut, arbetar långsammare och upplever arbetet som mer stressat. Man måste därför göra en avvägning om hur stor informationsmängd som skall visas på skärmen. Det finns forskare som rekommenderar att skärmbilden inte skall fyllas ut till mer än 25 %. Andra menar att några sådana rekommendationer inte kan ges då det är mer avgörande hur informationen *presenteras* än hur mycket av skärmbildens yta som fylls. Undersökningar har emellertid visat att vissa typer av arbetsuppgifter kräver tillgång till mycket information samtidigt. Därför skall man inte vara rädd att visa för mycket information på en gång då situationen kräver det. Ju mer van användaren blir vid systemet, desto bättre blir dennes avläsningsstrategier.

Problemet med att användare ser för lite information löser många designers genom att öppna nya fönster, s k ”fönstersjuka”. Genom att klicka och få upp ett nytt fönster kan användarna se den information de är intresserad av. I många fall innebär det att flera nivåer måste passeras innan den aktuella informationen visas, vilket leder till att skärmytan tas upp av en massa fönster. Användaren får därefter flytta fönster, scrolla, förstora och förminska för att se informationen som krävs. Det är i detta fall bättre att låta fönstren innehålla mycket information då bl a korttidsminnet belastas mindre, än att dela upp informationen på flera små fönster.

4.3.3.2 Layout

Målet för utformning av layout är läslighet, klarhet och konsekvens. En skärmlayout bör disponeras så att informationen tydligt och klart uppfattas av användaren. Layouten skall vara konsekvent i systemet, vilket kan göras genom att man t ex alltid låter OK-knappen vara placerad längst ner till höger.

4.3.3.3 Meddelanderutor

Meddelanderutor används när systemet skall informera användaren om någonting. Det kan vara informations-, varnings- eller felmeddelanden. Exempelvis kan en varningsruta informera användaren om denne försöker stänga ett program utan att ha sparat sina ändringar.

En meddelanderuta måste avslutas innan något annat kan göras i systemet. Informations- och felmeddelanden skall alltid avslutas med OK, medan ett varningsmeddelande även bör ha en Cancel-knapp för att kommandot skall kunna avbrytas. Ett meddelande skall placeras mitt i det fönster som meddelandet avser.

En meddelanderuta förses ofta med en ikon (t ex ett utropstecken) för informationsmeddelanden. Ikonen skall vara den samma på alla liknande meddelanderutor för att inte förvirra användaren.

4.3.3.4 Feedback

Har användaren utfört ett kommando bör systemet ge ett svar på att kommandot tagits emot, d v s ge feedback. Detta kan exempelvis göras genom att det syns på knappen att den trycks in.

Vidare bör feedback ges om en process kommer att ta en stund. Det är en god idé att använda två markörsymboler om svarstiderna varierar mycket. Om processen beräknas ta mindre än fem sekunder räcker det med att ändra muspekaren till ett timglas. Annars är det bra att visa hur lång tid det är kvar av processen, allra helst angivet i tid (s).

Ett system bör vara feltolerant, d v s kunna upptäcka fel och informera användaren om dessa via väl genomtänkta felmeddelanden. Ett fel kan informeras för användaren i form av en meddelanderuta, eventuellt i kombination med en ljudsignal. Ett felmeddelande bör informera om när felet uppstod, vilken typ av fel det är, samt möjliga sätt att korrigera det.

4.3.3.5 Generella koder

Generella koder kan användas på egenskaper som återkommer i program efter program. Exempelvis kan man i en meny indikera om ett alternativ är valbart eller ej. Det kan göras genom att låta det icke valbara alternativet vara gråtonat medan det valbara är svart. Man kan även låta alternativet som skall väljas ändra färg när musen rörs över ordet. Detta är en form av feedback som meddelar användaren vilket alternativ denne är på väg att välja.

4.3.3.6 Visuella koder

Att använda sig av visuella koder är mycket användbart då de i regel inte tar mer plats på den disponibla skärmytan. Röd text tar inte upp mer plats på skärmen än vad svart text gör. Genom att använda sig av ikoner, knappar, färger, fonter m m kan informationen presenteras på ett mer lättläst sätt och hjälpa användaren att minnas och gruppera information.

Ett exempel på visuell kodning kan vara att alla avvikelser och negativa siffror markeras med rött. Detta medför att användaren inte behöver kontrollera alla siffervärden utan kan koncentrera sig på värden som inte är normala.

Visuella koder skall ha samma innebörd applikationen igenom för att inte förvirra användaren. Man bör dock tänka på att visuella koder bara skall användas när det är befogat. Om en applikation förses med för mycket visuella koder kan användaren tröttna efter ett tag, vilket leder till att poängen med visuella koder går förlorad. Det kan vara lämpligt att göra en lista över alla de koder som förekommer i gränssnittet. Listan kan exempelvis innehålla:

- Alla olika färger som används och den betydelse de har.
- Alla olika fonter som används och den betydelse de har.

- Alla ikoner som används för att betyda något särskilt.
- Alla olika typer av inramningar och deras betydelse.

Kontrollera att koderna är konsekventa och att de inte säger emot varandra.

4.3.3.7 Ikoner

Fördelen med att använda ikoner²¹ är att de förstärker metaforer. De är lätta att känna igen och minnas och gör dessutom gränssnittet lite roligare. Det är viktigt att tänka på att ikoner kan ha olika innebörd för olika människor. Därför är det svårt att för en viss funktion utforma en entydig ikon som betyder samma sak för olika individer.

En ikon bör inte ha för hög detaljeringsgrad. Människan skiljer lättare mellan ikoner som är av karikatyrstil än mellan de som är fotografiskt lika med den tilltänkta funktionen.

Att förstå en ikons innebörd är mest intressant första gången man kommer i kontakt med den. Andra gången är ikonens funktion snarare att påminna om dess innebörd än att förklara den. Efter några gångers användande får ikonen rollen att skilja funktionen från andra funktioner. Därför är det bra att låta ikoner med olika betydelser ligga skilt från varandra.

4.3.3.8 Knappar

Det skall synas tydligt på skärmbilden vad som är knappar. Knapparna skall vara placerade i den sekvens de troligen kommer att användas i. Knapparna bör inte vara alltför små eller ligga för nära varandra. Då är det lätt att användaren ”missar” knappen, speciellt om operationen utförs ofta. Därför skall man tänka på att inte placera en knapp som ger allvarliga konsekvenser omedelbart under en knapp som kommer att användas ofta.

4.3.3.9 Färger

Färger kan användas för att förstärka eller försvaga något. När man t ex väljer en bakgrundsfärg bör man tänka på att den inte skall vara tröttande att titta på under långa arbetspass. Dessutom bör man tänka på att färger är laddade med psykologiska effekter. Exempelvis kan blått och grönt verka lugnande medan rött verkar tröttande och väcker aggressiva känslor. Gult verkar stimulerande på kreativitet och fantasi.

Det kan vara mycket tidsödande att välja passande färger och därför är det bra att utforma en mall att gå efter. I en skärmvisning bör man arbeta med ett urval av färger och färgkombinationer. Man kan t ex bilda en ”familj” bestående av 6-8 färger där varje färg finns representerad i varianterna:

- Bakgrundsfärg

²¹ En ikon är en grafisk bild som kan symbolisera t ex ett program, en fil eller en händelse.

- Förgrundsfärg
- Särskilt nedtonad bakgrundsfärg

4.3.3.10 Fonter

Fonten (bokstavsstilar) kan också bidra till att en text blir mer eller mindre lättläst. Dessutom har kodning med fonter den fördelen att även färgblinda personer kan dra nytta av den. Man bör dock försäkra sig om att fontstilarna är lätta att läsa och att inte alltför många olika typsnitt förekommer i gränssnittet.

4.4 Webb-gränssnitt²²

Att bygga en webb-klient för ett system innebär en del nya problem som tidigare inte funnits. Många av de problem som uppstår rör tekniska begränsningar som kan lösas med hjälp av nya avancerade tekniker. Vi kommer inte att gå in på några sådana tekniker. Istället vill vi peka på vissa saker som man som designer bör tänka på vid utformning av en webb-klient.

Det finns inte så mycket skrivet om gränssnittsdesign för webb-klienter. Den information som finns att få tag på, rör främst design av s k hemsidor. Vi har ändå valt att sammanställa de riktlinjer vi stött på för utformandet av ett webb-gränssnitt.

4.4.1 Framträdande skillnader mellan webb- och windows-gränssnitt

De främsta och mest tydliga skillnaderna i utseende mellan webb- och windows-gränssnitt kan delas upp enligt följande:

- *Menyer* används flitigt i traditionella windows-program, men sällan eller aldrig i Webb-baserade lösningar.
- *Dialogrutor* används ofta i windows, men sällan i webb-program.
- *Länkar* används inte i windows, men ofta i webb-program.
- *Knappar* finns i båda lösningarna, men har ofta en mer framträdande roll i webb-program.

4.4.2 Viktigt vid webb-design

Ett av de största problemen med webb-klienter gäller svarstider. Det tar en stund innan en sida laddas om vilket kan leda till att användaren blir irriterad. Den typ av feedback som ges till användaren i traditionella windows-klienter kan inte ges till webb-klientens användare. Därför måste man tänka på att minimera svarstiden och helst disponera sidan så att den inte behöver laddas om. Samtidigt måste man tänka på att en användare inte vill scrolla för att se information. Dessutom bör stora bilder som tar tid att ladda ner undvikas.

²² L. Danielsson (1997), S. Lundhem, R. Rydberg

Det skall vara lätt att navigera sig genom applikationen. Ett alternativ kan vara att låta en *header*²³ följa med på alla sidor. Om länkarna är få är det bra om de presenteras i form av ikoner eller bilder. Om många länkar skall ingå i headern blir det mest överskådligt att använda ord. Det är också viktigt att man som användare vet *var* man befinner sig. Genom positionsangivelser kan man underlätta navigering och pedagogik och det är lättare att förstå informationen om man ser en förklarande rubrik.

Ett webb-gränssnitt innehåller i regel många länkar, antingen som ankare (länkar inom samma sida) eller länkar till nya sidor. Man bör vara konsistent i sitt sätt att ange vad som är en länk och vad som inte är en länk. Följ den vanliga standarden att ett understruket ord alltid är en länk. Länkar som inte är klickade på bör vara blå medan besökta länkar bör vara lila. Det skall synas om en länk är valbar eller ej.

Det är mycket viktigt att under utvecklingen testa klienten i olika webb-läsare, som t ex Netscape och Explorer. Det är inte alls säkert att utseendet och funktionaliteten är lika i de två olika läsarna.

4.5 Sammanfattning

Ett dåligt system kan inte gömmas av ett bra gränssnitt, men ett bra system kan förstöras av ett dåligt gränssnitt. För att en applikation skall bli lyckad krävs ett samspel mellan båda delarna och således ett samspel mellan slutanvändare och systemutvecklare.

Ofta är flera programmerare med och utformar olika delar av gränssnittet. Det kan då uppstå problem om inga riktlinjer har specificerats. I så fall finns risken att var och en t ex väljer olika visuella koder, se avsnitt 4.3.3.6, *Visuella koder*, vilket kan leda till att koderna blir mer störande än stödjande.

Nedan följer några konkreta riktlinjer vid utformandet av ett gränssnitt:

- Minimera mängden presenterad information genom att endast visa information som är nödvändig för användaren.
- Gruppera informationen på ett sätt som ökar läsbarheten. Var noga med en fix placering då dålig linjering mellan grupper av data irriterar användaren i onödan.
- Använd visuella koder för viktigt information. Välj kod med omsorg och se till att avbildningen mellan egenskap och uttrycksmedel blir konsistent.
- Presentera texten på ett lättläst sätt.
- Standardisera skärmlayouten så att användaren lättare hittar information på skärmen.
- Försök begränsa antalet nivåer av fönster som en användare måste arbeta med vid frekvent förekommande uppgifter.

²³ En header kan beskrivas som en rad med länkar, antingen i form av ord eller ikoner, som underlättar navigeringen på en webb-sida.

- Finputs inmatningsrutiner genom att minimera antalet knapptryckningar och växlingar mellan mus och tangentbord för en sammanhängande inmatningsfrekvens.
- Ge feedback till användaren.
- Upprätta ett dokument över de riktlinjer som skall användas i systemet.

5 Undersökningar

5.1 Fältstudie - Gullspång

I följande kapitel redogörs för de djupintervjuer som gjorts med Gullspånga användare av IFS/Time & Attendance webb-klient. Till att börja med ges en kort historik, samt en beskrivning av hur undersökningen gått tillväga. Resultatet av undersökningen presenteras sedan, och läses lämpligen med bilagorna 1a och b som komplement. Kapitlet avslutas med ett antal förändringsförslag och slutsatser.

5.1.1 Metod

Som en uppföljning av designarbetet kan, som vi tidigare talat om, ingå att göra en användbarhetsanalys. För att få en inblick i hur detta kan tänkas gå till har vi själva gjort en liten användningsundersökning. Undersökningen kan sägas motsvara en tidigare beskriven *Fältstudie*, se avsnitt 3.4.2, *Analysmetoder*. Då våra resurser var begränsade fanns ingen möjlighet att göra undersökningen så omfattande som vi önskat. Vi fick ändå en god inblick i hur viktigt det är att göra denna typ av utvärdering med användarna.

Gullspång, som är ett av Sveriges största kraftföretag, finns på flera orter i landet och har ca 1500 anställda. Sedan 1 april -98 har ett antal anställda kört IFS/Time & Attendance webb-klient. Vi har varit på Gullspång i Kungsbacka, där ungefär 60 personer börjat eller skall börja använda webb-klienten för tidrapportering, och gjort intervjuer angående klientens gränssnitt med fem användare. Man har även implementerat windows-klienten för IFS/Time & Attendance hos vissa användare, men ingen av de personer som ingick i vår undersökning hade använt windows-klienten ännu. Då de flesta användarna på Gullspång inte kört systemet i mer än en vecka var syftet med undersökningen att fånga in några användares första intryck av, och synpunkter på gränssnittet. Vi satt med användaren vid dennes arbetsplats och hade gränssnittet framför oss så att vi på ett naturligt sätt kunde diskutera kring detta. Vi hade i förväg ställt samman ett tjugotal frågor som vi ställde till varje användare. Dessa finns i detalj i bilaga 1b. Det fanns också utrymme för användaren att komma med spontana kommentarer och synpunkter.

Då vi utgår ifrån en så pass liten grupp användare, kan vi inte med säkerhet säga att de slutsatser vi kan dra utifrån undersökningen är helt reliabla. Vi försökte ändå fånga en samling användare som kan tänkas motsvara olika användarprofiler inom företaget. De fem män vi intervjuade är i åldrarna 24, 31, 31, 45 och 54 år och ansåg sig alla ha mycket datorvana. Med datorvana menas här daglig kontakt med datorer, men ingen programmeringserfarenhet. För vidare uppgifter om respektive användare se bilaga 1b.

Med användare avses i det följande dessa fem personer som deltog i intervjuerna. För bättre förståelse av följande sammanfattning finns en kortfattande beskrivning, samt bilder av skärmutskrift i bilaga 1a.

5.1.2 Sammanfattning av intervjusvar

Innan man övergick till IFS/Time & Attendance webb-klient utförde användarna sin tidrapportering i ett Unix-program de själva utvecklat. I samband med införandet av PC-miljö valde man dock att lämna det gamla systemet. Fyra av användarna tyckte inte att det fanns några likheter i sättet att rapportera tid mellan den gamla miljön och IFS/Time & Attendance. En användare kände igen en del av upplägget. Användarna hade vid intervjutillfället kört webb-klienten olika länge, allt ifrån en vecka till en månad. Fyra användare genomgick en kort introduktion eller utbildning vid olika tidpunkter inför användandet av webb-klienten. Ett par av dessa hade dock testat webb-klienten själva innan någon utbildning gavs. Två av de användare som kört webb-klienten i en månad tyckte att utbildningen var dålig och hölls för tidigt (innan systemet var färdigtestat), medan en annan användare upplevde den som mycket givande. Hjälpen som finns i webb-klienten visade sig vara ganska dåligt utnyttjad. Två av användarna hade inte ens sett att den fanns.

Två användare tyckte att det var ganska lätt att lära sig systemet, och de andra tre tyckte att det var mycket lätt. Man tyckte även att det var relativt lätt att genomföra tidrapporteringen, även om vissa moment upplevdes som lite ”knöliga”. Bl a tyckte några att komboboxarna innehöll för många alternativ, och att innehållet var svårt att tyda. Dessutom tyckte man att det var irriterande med långa väntetider för omladdning av sidorna. Ingen hade lagt ner speciellt mycket tid på tidrapporteringen, det rörde sig om antingen några minuter per dag eller en halvtimme i veckan. Det gamla systemet innehöll inte bara tidrapportering utan även möjligheten att rapportera reseräkning, flerdagstraktamente och flex-tider. Dessa funktioner ingår inte i IFS/Time & Attendance. Bortsett från att några användare saknade en del funktioner upplevde man att arbetet med att rapportera tid effektiviserats jämfört med tidigare tillvägagångssätt.

Överlag tyckte man att gränssnittet var snyggt, och hade en positiv inställning till webb-lösningen. En av användarna som i ett tidigt skede började köra webb-klienten upplevde den som mycket långsam. Nämnas bör dock att systemet blivit snabbare sedan dess.

Vid frågan om man såg all information man behöver på de olika skärmbilderna nämndes bl a att det saknades någon form av totala summeringar på första sidan, samt information om vilken lön rapporteringen resulterar i efter attestering. En användare saknade information på attesterings-sidan, för att kunna se om det finns något att attestera. En användare ansåg att onödigt mycket information visades på översikt-bilden.

Samtliga användare tyckte att ikoner (bilder) och färger var bra. Ikonernas innebörd var inte helt självklara, och kanske lite förvirrande vid första anblicken. En användare påpekade att det såg ut som om ikonerna hämtats från Internet, vilket så också var fallet, och inte var framtagna speciellt för webb-klienten. Tack vare den underliggande texten hade man dock inga större problem med att förstå ikonernas innebörd. Eftersom gränssnittet kompletterats med ett antal funktioner som inte används av alla användare, finns det ikoner (t ex *Op stämplingar*, *Op resultat*) som inte har någon mening för just dessa användare.

Användarna förstod, efter att ha prövat sig fram, vilka bilder som var länkar, även om *Ny-länken* på översikt bilden inte var en självklar sådan för alla. En användare påpekade att han förväntat sig att IFS logotype skulle vara en länk till vidare information om företaget med bl a telefonnummer att ringa om problem uppstår.

Ingen av användarna hade några problem med tidsmässig navigering, men behovet av att förflytta sig mer än två veckor framåt eller bakåt i tiden varierade. Tre av användarna uppgav att de ibland går tillbaka allt ifrån en till tre månader för att kontrollera mot lönespecifikationen. Det fanns även önskemål om att kunna gå tillbaka i tiden för att skriva ut tidrapporter för en månad, ett halvår eller ett år.

När det gällde hur gränssnittet passar på skärmen, tyckte alla användarna att de fick scrolla onödigt mycket. Man hade bl a synpunkter på att IFS logotype tar upp alldeles för stor plats, vilket gör att det översta utrymmet på sidan är outnyttjat. Dessutom ansåg någon att det på översikt bilden fanns många fält som inte används, och att det kändes avigt att behöva scrolla sig längst ner på sidan för att t ex godkänna hel vecka.

5.1.3 Övriga reflektioner

Många synpunkter som användarna spontant förde på tal handlade om funktionalitet i systemet, och inte så mycket om designen. Vi tar ändå med en del av dessa kommentarer, för att ta del av användarnas helhetsintryck.

Alla hade en positiv inställning till att kunna göra tidrapporteringen via Intranet. Mycket på grund av att man inte är beroende av några speciella maskiner eller särskild programvara. Användarna tyckte att gränssnittet var överskådligt och man gillade översikt bildens schema-struktur. De tyckte att det var bra att tiden registrerades automatiskt om man t ex inte hade någon övertid eller frånvaro att rapportera, men sämre att behöva gå in och godkänna sin tid även om man jobbat helt enligt schemat. En användare påpekade att det går mycket snabbare och är roligare att rapportera, jämfört med det gamla rapporteringssättet. En användare ansåg att länkarna för stämpling och tillägg borde ligga intill varandra överst på sidan, lätt åtkomliga, då dessa används mest frekvent av de flesta användare. Som det är nu måste man scrolla för att se ikonerna för tillägg. När det gällde tidnavigeringen fanns det önskemål om att gällande veckonummer skulle vara markerat (t ex rött) så att man snabbt hittar tillbaka.

Generellt sett tyckte man att det inte alltid var helt klart *vad* som skulle fyllas i, och *var*. Det var t ex oklart om bilersättning skulle rapporteras i mil eller kilometer. Dessutom upplevde användarna listorna i komboboxen för lönearter (på inmatningssidan *Nytt tillägg*) som krångliga att förstå sig på då de innehåller lönearter för hela koncernen. De var dock medvetna om att det kanske rör sig om en utbildningsfråga. Ett önskemål angående dessa listor var att exempelvis kunna klicka på varje alternativ i listan för att få mer information. Man önskade också någon slags indelning av inmatningssidan för ny stämpling, så att varje ny stämpling är uppdelad i olika typer, t ex restid, normalt看 och övertid. Mycket information bör dock övervägas mot snabbhet, det är viktigt att det går fort. Det som upplevs ta mest tid är omladdningen av sidorna.

Användarna tog även upp saknade funktioner, såsom möjlighet att rapportera flerdagstraktamente och flex, och att kunna göra utskrifter av tidsrapporter. Att man inte kan ändra sitt eget schema upplevdes också som en stor nackdel. Detta kan vara nödvändigt om man t ex skall byta dag med någon annan anställd.

5.1.4 Slutsatser

Det visade sig under intervjuernas gång att de spontana synpunkter användarna hade främst handlade om funktionalitet, vilket tyder på att designen var bra och diskret. Ett par av de som använt webb-klienten längst verkade ha fått ett mer negativt intryck av webb-klienten. Detta beror antagligen på att systemet inte var helt färdigt när det implementerades, varför utbildningen av användarna kanske skulle ha senarelagts.

5.2 Laborietvärdering förstagångs användare

Följande kapitel beskriver användarundersökningen vi gjorde med förstagångs-användare av IFS/Time & Attendance. Vi har undersökt två olika gränssnitt, SQL/windows och webb, för att reda ut vilket av dessa som är snabbast att rapportera i för en förstagångs användare. Kapitlet innehåller tidsstudier, kommentarer och slutsatser. En detaljerad beskrivning av gränssnitten återfinnes i bilaga 2a. I bilaga 2b finns de instruktioner om uppgifterna som gavs till testpersonerna, samt övriga frågor och instruktioner.

5.2.1 Testscenario

Vi har gjort en Laborietvärdering, se avsnitt 3.4.2, *Analysmetoder*, med förstagångs användare som inte varit i kontakt med vare sig webb- eller windows-gränssnittet för modulen IFS/Time & Attendance tidigare. Undersökningen ägde rum i en uppbyggd testmiljö i IFS:s lokaler. Genom att mäta tidsåtgång hos gruppen ville vi få fram vilket av de båda gränssnitten som är enklast att hantera och således lämpar sig bäst för en förstagångs användare. Däremot har vi inte räknat felfrekvens hos användarna, då det är svårt att definiera vad som skall, och inte skall, räknas som ett fel. Vi har dock noterat de problem som våra testpersoner stött på och tagit del av deras synpunkter som kom fram under undersökningen.

I undersökningen skulle användarna genomföra fyra olika moment i tidsrapporteringen. Dessa var *rapportering av arbetad tid*, *rapportering av frånvaro*, *godkännande av rapporterad tid* och *ta bort rapporterad tid*. Testpersonen fick en kort genomgång av modulens funktioner, samt uppgifter om vilka data som skulle matas in, se bilaga 2b. De fick även information om windows-klientens kommandometoder och ikonernas betydelse. Som hjälpmedel fick användarna ha en musmatta med windows-klientens ikoner och kortkommandon. Sedan fick testpersonen i lugn och ro utföra uppgifterna medan vi tog tid och noterade de fel som gjordes.

Vår testgrupp bestod av sju personer varav fem går sista året på ADB-programmet Dessa fem anser vi ha stor datorvana, både vad det gäller olika system och

systemutveckling. De andra två anser vi ha mindre datorvana, eftersom de ”bara” är bekanta med Word och Internet. Alla testpersoner är vana vid PC, Microsoft-miljö och Internet, vilket gav dem en fördel när de skulle rapportera i IFS/Time & Attendance. Som förstagångsanvändare av IFS/Time & Attendance kategoriserar vi vår testgrupp som noviser, se avsnitt **3.1, Användarkategorier**. För att lättare visa vilket gränssnitt som gick snabbast att rapportera i har vi valt att presentera tidsstudierna i excel-diagram. Testpersonerna är indelade i två grupper; dels de som började rapportera i webb-miljön, grupp 1, och dels de som började rapportera i windows-miljön, grupp 2. Grupp 3 avser den totala tidsåtgången för grupp 1 och grupp 2.

5.2.2 Undersökningsresultat

För varje uppgift redogör vi först för hur det skall gå till att genomföra uppgiften i respektive klient (SQL/windows och webb). Därefter berättar vi kortfattat om de problem våra testpersoner stötte på, samt deras kommentarer.

För att slippa upprepa på vilka sätt man kan utföra ett kommando i windows-miljön menar vi i fortsättningen med formuleringen *kommando* att man antingen väljer ett kommando i en meny, trycker på respektive knapp i knappraden eller använder sig av kortkommandon.

5.2.2.1 Rapportering av arbetad tid

Uppgift 1 innebar att testpersonen skulle rapportera arbetad tid för ett visst datum mellan 08:00 och 16:00. Tidtypen för både in- och utstämpling skulle vara *normal*.

- Webb-klienten:
Genom att klicka på en liten ikon med texten *New* kommer man till en skärmbild med inmatningsfält. Här matar man in tid för in- och utstämpling och väljer tidtyperna i komboboxar. Därefter trycker man på save-knappen för att komma tillbaka till översiktsskärmbilden där stämplingen nu syns.

New-ikonen ställde till problem för flera testpersoner och många menade att det vore bättre om hela rutan var en länk. New-ikonen är alldeles för liten och man måste röra med musen precis över för att se att den är en länk. Några hade därför problem med att hitta länken och försökte bli gå in på ikonerna *Schedule* och *Normal Clockings* innan de hittade rätt. En person förstod inte, när han kommit till inmatningsformuläret, hur han valt rätt datum och ansåg det mest vara en lyckoträff att han hamnat på rätt ställe. När testpersonerna väl kommit till inmatningsformuläret var det inga problem att mata in rätt värden och trycka på save.

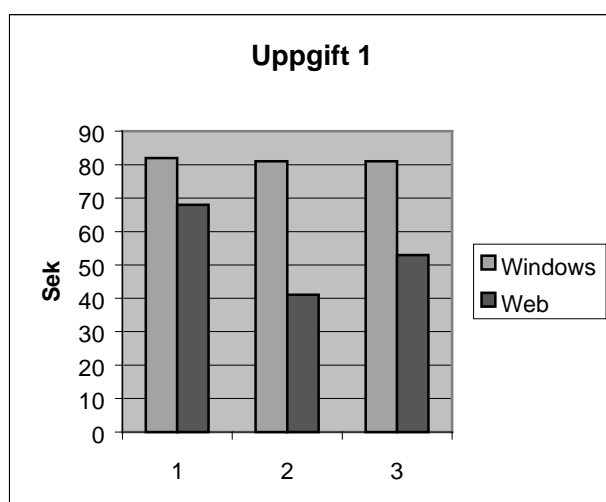
- Windows-klienten:
Det första man skall göra i windows-gränssnittet är *populate*, för att koppla rapporteringen till sitt anställningsnummer. Därefter måste man stå i tabellen och utföra kommandot new för att initiera att en ny stämpling skall göras.

Sedan rapporterar man sin tid direkt i tabellen. Man avslutar genom att välja kommandot save.

Några av personerna hade problem med göra populate. De sökte först i komboboxen med emp_no, vilken visade sig vara tom. Flera av våra testpersoner fick fråga innan de kunde utföra kommandot. Det kan ha berott på att vi inte gav dem tillräckligt tydliga instruktioner inför uppgiften. Många hade problem när de skulle göra sin första stämpling. Ett par av våra testpersoner ville mata in *Temp Day Type* i huvudet, andra tog fel på populate och new. Ett par av våra testpersoner fick fråga efter new medan andra sökte efter new på musmattan. När allt väl var inmatat sparade alla korrekt.

- Tidsstudie

Uppgift 1, se *Figur 10*, utfördes i båda fallen snabbast med webb-klienten. De testpersoner som började rapportera i windows-miljön var de som rapporterade snabbast i webb-miljön. Det beror till stor del på att terminologin och arbetsgången kändes igen. De som började rapportera i webb-miljön verkade tidsmässigt däremot inte ha lika stor nytta av dessa kunskaper när de genomförde rapporteringen i windows-miljön. Vi kan dra slutsatsen att det är lättare att gå från windows-miljö till webb-miljö än tvärt om.



Figur 10 Tidsstudie - Rapportering av arbetad tid

1. Började med webb-gränssnittet
2. Började med windows-gränssnittet
3. Totalt

5.2.2.2 Rapportering av frånvaro

Uppgift 2 gick ut på att rapportera frånvaro mellan klockan 14:00 och 16:00 samma datum som ovan. Det är inte tillåtet med överlappningar, vilket innebär att man först måste ändra dagens tidigare stämpling till 08:00 och 14:00, för att sedan rapportera frånvaro mellan 14:00 och 16:00. Tidtypen för frånvaron skulle vara *absence*.

- Webb-klienten:

För att rapportera sin frånvaro skall man först ändra dagens tidigare stämpling genom att klicka på tidlänken för aktuell dag som finns i raden clockings. Här skall man ändra out-stamp till 14:00 och trycka på save-knappen. Frånvaro rapporterar man genom att klicka på new-ikonen för den aktuella dagen på absence-raden. Sedan matar man in frånvarotiden och väljer tidtyper i komboboxar och trycker på save-knappen. Dagens rapporterade tid resulterar således i två stämplingar.

Flera användare hade problem med att förstå vilken länk som skulle leda dem till sidan för ändringar. Några av testpersonerna använde new-länken för clockings, vilket resulterar i en överlappning. Andra ville rapportera sin frånvaro innan de ändrade den gamla stämplingen, vilket också resulterar i överlappning. Ett par personer ”missade” ändrings-länken och klickade på new av misstag. De två länkarna ligger alldeles för nära varandra. När våra testpersoner väl kommit fram till inmatningssidan var det inga problem att mata in rätt tid och rätt tidtyp.

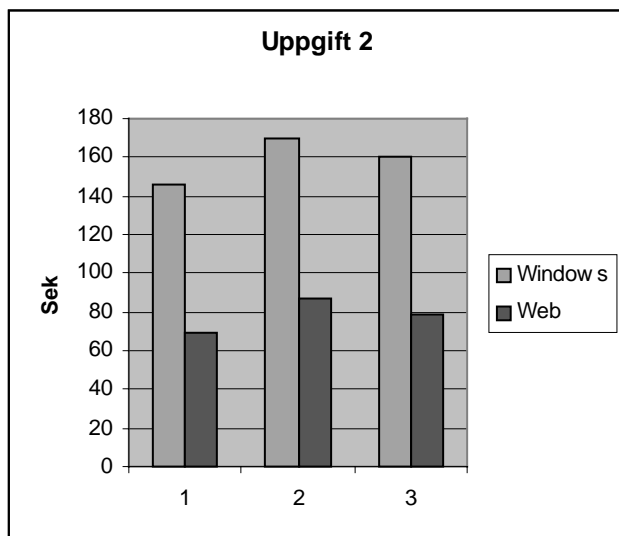
- Windows-klienten:

Windows-klienten skiljer sig något från webb-klienten då det är mer information som skall matas in, t ex värden för *wage-code* och *wage-type*. Dessa värden skall väljas i de värdelistor som finns. Annars liknar arbetsgången webb-klienten förutom att alla stämplingar görs på samma sida. Rapporteringen skall resultera i två stämplingar.

Att ändra den tidigare stämplingen var inga problem. Däremot ville ett par personer göra populate istället för new när en ny rad skulle läggas till vid rapporteringen av frånvaron. Det uppstod svårigheter när hela tabellen inte syntes och man var tvungen att scrolla för att nå fram till wage-code och wage-type. De testpersoner som inte hade någon datorvana visste dessutom inte vad scrolla var. Vi fick påminna flera personer om värdelistornas funktioner. Några valde att dubbelklicka i värdelistan för att välja värde medan andra valde genom att trycka på OK-knappen.

- Tidsstudie

Uppgift två, se *Figur 11*, genomfördes också snabbast i webb-miljön. Det är intressant att notera att de som började i windows-miljön rapporterade långsammast i båda miljöerna. Det kan bero på att man väntade sig att uppgiften kunde lösas på samma vis i webb- som i windows-miljö.



Figur 11 Tidsstudie – Rapportering av frånvaro

1. Började med webb-gränssnittet
2. Började med windows-gränssnittet
3. Totalt

5.2.2.3 Godkännande av rapporterad tid

Uppgift 3 innebar att testpersonen skulle *godkänna* (Confirm) sin rapporterade tid för att den skall kunna gå vidare till lönesystemet.

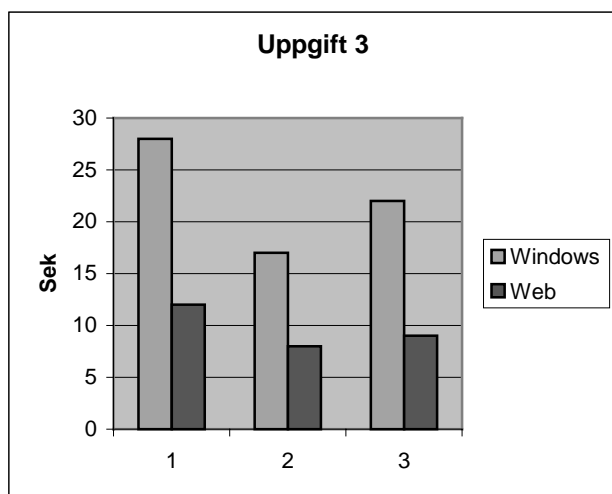
- Webb-klienten
Genom att klicka på länken *Confirm* för den aktuella dagen kommer man vidare till en sida där det bara är att trycka på ok-knappen för att godkänna dagen.

Alla våra testpersoner godkände sin tid helt felfritt.

- Windows-klienten
För att godkänna sin tid i windows-miljön kan man antingen välja *Confirm* i menyn *Commands* eller använda sig av höger musknapp.

Flera av våra testpersoner ville kryssa i *Confirm*-rutan som syns i huvudet. Den rutan är dock bara till för att visa om dagen redan är godkänd. De flesta hittade funktionen i menyn *Commands*. En person försökte hitta *Confirm* som en snabbfunktion på musmattan.

- Tidsstudie
Även uppgift 3, se *Figur 12*, utfördes snabbast i webb-miljön, mycket på att kryssrutan *confirmed* i windows-miljön förvirrade.



Figur 12 Tidsstudie - Godkännande av rapporterad tid

1. Började med webb-gränssnittet
2. Började med windows-gränssnittet
3. Totalt

5.2.2.4 Ta bort rapporterad tid

Uppgift 4 innebar att alla stämplingar för det aktuella datumet skulle tas bort.

- Webb-klienten

För att ta bort en stämpling måste man gå via de tidlänkar som finns för respektive stämpling och trycka på remove-knappen.

I webb-klienten var det flera som tog bort godkännandet (remove confirmation) innan de raderade sina stämplingar. På skärmbilden för webb-klienten syns en länk där det står *Remove Confirmation*. Det faktum att länken syns gör det naturligt för användaren att ta bort godkännandet. Vi vill påpeka att ingen tog bort godkännandet i windows-klienten, vilket vi tror beror på att funktionen inte är lika lättillgänglig där. En annan teori är att användarna letade efter en funktion som hette Remove Clocking och då fann Remove Confirmation.

- Windows-klienten

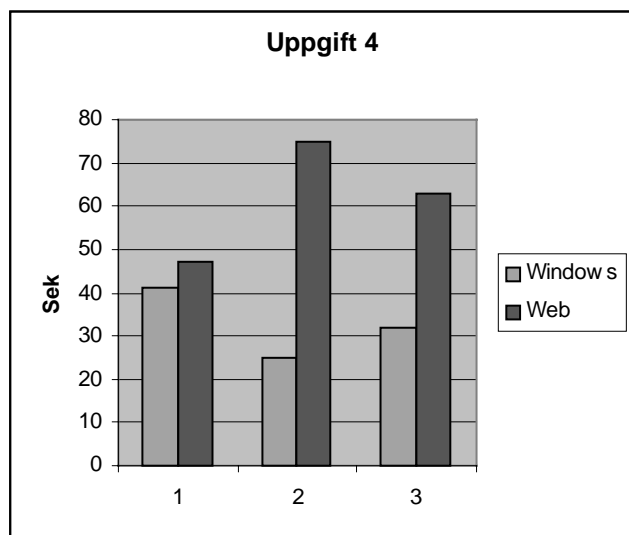
Genom att stå på den rad som skall raderas och utföra kommandot ta bort raderas en rad. Raden försvinner dock inte från skärmbilden förrän man tryckt på save-knappen.

Flera testpersoner letade ett tag efter rätt funktioner. Några valde att spara mellan borttagandet av de två stämplingarna medan andra markerade båda raderna innan de sparade.

- Tidsstudie

Det är intressant att notera att raderingar, se *Figur 13*, genomfördes snabbast i Windows-klienten. Mest markant är skillnaden för de som började rapportera i

windows-miljön. Det beror förmodligen på att svarstiderna i webb-klienten blir väldigt långa då bara en stämpling i taget kan raderas, samt att det tog tid innan de fann rätt länk för att ta bort. Dessutom hade testpersonerna mot slutet av undersökningen börjat vänja sig vid windows-miljön.

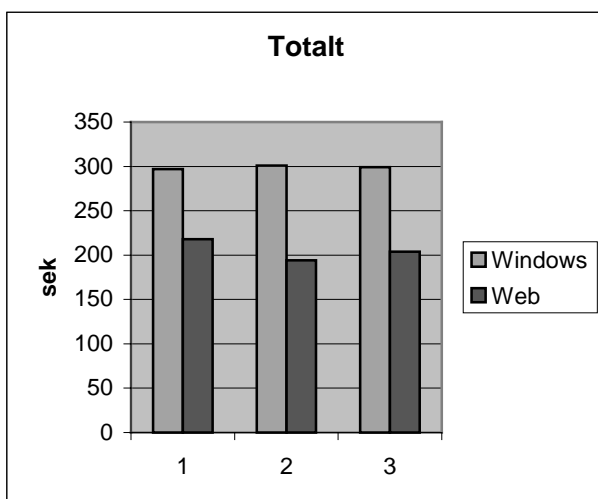


Figur 13 Tidsstudie - Ta bort rapporterad tid

1. Började med webb-gränssnittet
2. Började med windows-gränssnittet
3. Totalt

- Total rapporterad tid

Vi kan slutligen konstatera att det gick snabbast för våra testpersoner att rapportera i webb-klienten, se *Figur 14*. Vi märker inga tydliga skillnader beroende på i vilket gränssnitt testpersonerna börjat rapportera sin tid i.



Figur 14 Mätning av tid, totalt.

1. Började med webb-gränssnittet
2. Började med windows-gränssnittet
3. Totalt

5.2.3 Övriga reflektioner

Utöver vår egentliga undersökning framkom en del andra synpunkter på gränssnitten. Exempelvis tyckte en av våra testpersoner att webb-gränssnittet var alldeles för ”plottrigt” med för små bilder och länkar. Samma person kommenterade att det var förvirrande att t ex ikonen ”tummen upp” återfanns på två ställen. På det ena stället är ikonen en länk, medan den på det andra stället bara är en bild.

5.2.4 Slutsatser

Vi är väl medvetna om att en testgrupp om sju personer knappast kan räknas som en vetenskaplig grund att dra klara slutsatser ifrån. Vi vill dock hävda att vår undersökning ger tydliga indikationer på hur en förstagångsanvändare av IFS/Time & Attendance interagerar med detta system. Vi anser också att de kommentarer som kom fram under vår undersökning är av stor vikt för vår undersökning och för IFS.

Undersökningen gav följande indikationer:

Webb-klientens största nackdel är att den har långa svarstider. Exempelvis uppstår lätt irritation när man har tryckt på save eller cancel och sedan måste vänta på att sidan skall laddas om. En testperson tryckte på save-knappen flera gånger innan hon uppfattade att systemet tagit emot hennes kommando. Användaren väntade sig någon slags respons, *feedback*. Feedback skulle exempelvis kunna ges genom att låta muspekaren ta form av ett timglas under omladdningstiden.

Andra problem rörde sig mycket kring att hitta rätt länk för ändamålet och förstå vilka bilder som var länkar. Vi tänker främst på new-ikonen för t ex new clockings, samt tidlänken till sidan där ändringar görs.

Den främsta fördelen med webb-klienten är att den är roligare att arbeta med. Dessutom är det enkelt att förstå arbetsgången. Våra testpersoner rapporterade snabbast i webb-klienten och många uppfattade den som mest överskådlig. Testpersonerna föredrog att arbeta med webb-klienten men flera kommenterade att windows-klienten förmodligen känns mycket smidigare och snabbare när man arbetat med den ett tag, speciellt om det skall rapporteras in mycket tid på en gång.

5.3 Övriga undersökningar

Följande undersökningar av IFS/Time & Attendance respektive IFS/Project Reporting webb-klienter kan jämföras med en fältstudie, då de utförts i användarnas arbetsmiljö. Undersökningarna har gjorts av Saab och Ericsson och avser att utreda om webb-klienterna lämpar sig i deras organisationer.

5.3.1 Saab²⁴

Saab i Linköping har över 6 000 anställda som varje vecka lägger ner ca fem minuter var på att rapportera sin arbetade tid. Tillsammans lägger man alltså ner ungefär 500 timmar i veckan på detta. Därför söker man nu ett sätt som kan effektivisera rapporteringen. För detta ändamål har man testat IFS/Time & Attendance webb-klient för att undersöka om denna lösning har rätt funktionalitet för Saab. Undersökningen har gjorts med sju deltagare som fått svara på ett antal frågor under tiden de testat klienten.

Resultatet av undersökningen var mestadels positivt. Sammanfattningsvis tyckte man att webb-lösningen var mycket bra, delvis p g a att den innehåller standardiserad webb-funktionalitet som man snabbt lär sig att känna igen. Användarna tyckte att gränssnittet var mera lätt- och snabbarbetat än det system de kör idag. Nedan följer ett antal sammanfattande punkter över några kommentarer ur rapporten.

- Man saknade överblick över dagens stämplingar när man skall skapa en ny stämpling, och önskade en möjlighet att kunna skapa alla stämplingar från en och samma sida. Dessutom tyckte man det var svårt att skapa en stämpling för en dag som redan har en stämpling, då stämplingarna måste delas upp för att överlappning skall undvikas.
- Det nämndes att resultatfältet på översikt bilden var svårläst, då för mycket text visades här. Alla de användare som deltog i undersökningen tog fel på länkarna för att göra ny stämpling och ändringar.
- Man saknar en tillbaka-knapp på felsidan. Dessutom ligger inte det inskrivna värdet kvar när man tryckt *back* (i webb-läsaren) från felsidan.
- Man föreslog att varje ikon som inte är en länk skulle leda till en förklaring i hjälpen.
- Användarna tyckte att hjälpfunktionen var bra, förutom att den saknar någon form av navigeringsinformation.

5.3.2 Ericsson²⁵

På Ericsson har en utvärdering av IFS/Project Reporting webb-klient gjorts. Detta är en modul för rapportering av projekttid och webb-gränssnittet är mycket likt gränssnittet för IFS/Time & Attendance webb-klient. I undersökningen har man enbart tagit fasta på krav och behov hos Ericssons användare och alltså inte tagit hänsyn till att det är en standardprodukt. Nämnas bör också att användarna inte fick någon introduktion i applikationen innan testet utfördes.

Utvärderingen gick till så att man lät användarna, som representerades av två grupper; administratörer och ingenjörer, utföra ett antal uppgifter. Där ingick bl a att rapportera projekttid och ändra en inmatning.

Undersökningen resulterade i en rapport där man bl a kom fram till följande.

²⁴ Saab (1998)

²⁵ Ericsson (1998)

Tidrapporteringsformuläret och ikonerna som motsvarar olika delar av applikationen upplevdes som någorlunda lätta att förstå och jobba med. Några utav de problem man tog upp presenteras nedan.

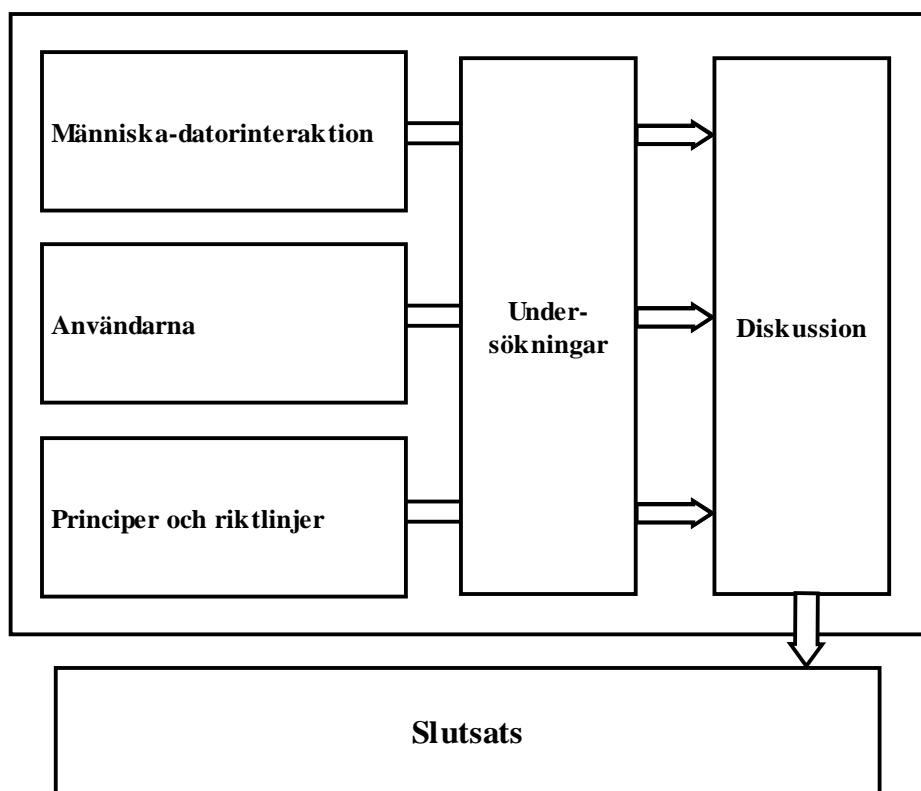
- För många dialoger – Användaren måste klicka sig förbi flera dialoger för att rapportera en dags projekttid. Detta gör att rapporteringen tar alldeles för lång tid. Här föreslog man att det vore bättre om det var möjligt att göra inmatningar och ändringar direkt på översiktsskärmen.
- Skärmstorlek och upplösning – Formuläret och dialogerna passade inte på skärmen, utan användaren måste scrolla för att se allt. Ett förslag var att dialogen skulle öppnas i en ny webb-läsare.
- Tangentbord kontra mus – En del användare vill eller kan inte använda sig av musen, vilket krävs här. Därför vill man att alla kontroller skall kunna nås i form av kortkommandon.
- Kopiera information – Man uttryckte ett behov av att exempelvis kunna kopiera information för att skapa en mall.
- Länkar kontra knappar – Användarna hade emellanåt svårt att hitta vissa länkar, på grund av att de inte utmärkte sig tillräckligt ifrån övriga kontroller på skärmen. Man tyckte att knappar vore ett bättre alternativ till många länkar, men att de måste vara synliga hela tiden utan att användaren skall behöva scrolla.

5.3.3 Slutsats

Enligt utvärderingarna kan vi utläsa att Saab var mer nöjda än Ericsson. För det första avsåg utvärderingarna inte samma klient, vilket naturligtvis måste tas i beaktande. Exempelvis saknade Ericsson anpassningar av webb-klienten i större utsträckning än vad Saab gjorde. Dessutom utfördes Ericssons utvärdering av en inhyrd konsult, speciellt inriktad på att utvärdera gränssnitt. Vi tror att detta kan vara en anledning till att Ericssons utvärdering var mer kritisk.

6 Diskussion

Följande kapitel avser att binda samman undersökningarna med de teorier som beskrivits i kapitel 1 – 3. Här diskuteras teorierna och undersökningarna i jämförelse med varandra utifrån vår synvinkel, se *Figur 15*. Beskrivning av gränssnitten, med bildutskrifter, finns i bilagorna 1a och 2a.



Figur 15 Uppsatsens disposition

6.1 Människa-datorinteraktion

I kapitlet om människa-datorinteraktion konstaterades att det är viktigt att ta hänsyn till människans minnespsykologi, dess kapacitet och begränsningar, vid utformning av gränssnitt. Människans minne är komplicerat och trots att långtidsminnet har obegränsade lagringsmöjligheter, kan hjärnan inte ta emot för mycket information på en gång. Minnets begränsningar gör att man vid utformning av gränssnitt bl a måste tänka på att inte visa för mycket information på skärmen samtidigt. I undersökningen med Gullspångs användare stämde denna teori till viss del. När användarna väljer löneart, när tillägg skall göras, visas på tok för mycket information samtidigt. Testpersonerna blev förvirrade och letade febrilt efter en kod de kände igen, då löneartslistan innehåller koder som gäller för hela Gullspångs koncern. Detta leder till att användaren lätt kan välja fel kod, vilket i sin tur bidrar till att systemets säkerhet minskar.

För att ytterligare minska minnets belastning bör användaren inte tvingas att komma ihåg information från en skärmbild till en annan. I IFS/Time & Attendance webb-klient måste användaren komma ihåg information om tidigare rapporterad tid när frånvaro skall rapporteras. Intervjuerna med Gullspångs användare visade dock att ingen tyckte att det var något problem att komma ihåg information från en sida till en annan. Det innebar heller inga problem för förstagångs användarna, vilket till stor del beror på att de hade tillgång till en "fusklapp" med uppgifter om de värden som skulle matas in. Ericssons utredning av IFS/Project Reporting visar däremot att det är ett problem. De föreslog att inmatningar skulle kunna göras i s k pop-up-fönster, så att den aktuella informationen hela tiden ligger synlig bakom. Detta är tyvärr inte möjligt att lösa med den teknik²⁶ som använts för att utveckla dessa webb-klienter.

Då Gullspångs föregående tidrapporteringsystem även innefattade funktionalitet som reseräkning och flerdagstraktamente förväntade sig flera användare att dessa funktioner skulle finnas i IFS/Time & Attendance. Detta är ett typiskt exempel på en negativ metafor, där användaren med hjälp av tidigare erfarenheter skapar förväntningar av det nya systemet. I Gullspångs fall har avsaknaden av dessa funktioner bl a resulterat i att man tvingats gå tillbaka till att rapportera sin reseräkning manuellt. Vi vill dock tillägga att reseräkning, inklusive flerdagstraktamente, är en fristående modul i IFS Applications, vilken inte implementerats hos Gullspång.

Övergången till det nya systemet medförde även en positiv metafor för Gullspångs användare, då de tyckte att det nya systemets gränssnitt utseendemässigt var bättre än föregångarens. Vi upptäckte också en positiv metafor i vår undersökning med förstagångs användare. De är alla vana vid Internet och visste således att blå, understrukna ord är länkar. Detta är av stor betydelse då hela webb-klienten är uppbyggd kring länkar. På liknande sätt kände testpersonerna sig hemma i PC- och windows-miljö. I Ericssons utredning hävdas att användarna oftast är mer vana vid PC/windows- och SUN/unix-miljö än vid Internet. Vi tror att dessa skillnader kommer jämnas ut, då allt fler människor får tillgång till Internet både på arbetsplatsen och i hemmet.

Gestaltlagarna har lärt oss att objekt som liknar varandra färg- eller formmässigt eller är grupperade antas höra ihop. I undersökningen med Gullspångs användare framgick att de funktioner som används mest frekvent ligger skilda från varandra, vilket skapade en viss förvirring. Vi syftar på *stämpling* och *tillägg*, se vidare bilaga 1a, där den senare är placerad flera rader under den föregående. Detta innebär också att ikonerna för tillägg inte är synliga på skärmbilden från början utan att användaren måste scrolla neråt för att se den.

I vår undersökning med förstagångs användare visade det sig att testpersonerna trodde att ikonerna i den vänstra kolumnen innehöll alla händelser för att utföra uppgifterna. Här försökte de flesta testpersonerna klicka för att hitta funktioner. Vi tror det kan bero på att Ny/New-ikonerna och tidlänkarna är små och svåra att upptäcka och att ikonerna, med förklarande text, i vänstra kolumnen drar uppmärksamheten till sig. Det kan även bero på att det inte syns om ikonerna är länkar. Av egen erfarenhet vet vi att en ikon kan markeras som länk genom att man lägger en blå ram runt den, vilket i detta fall inte gjorts, kanske på grund av att det ser snyggare ut utan. Då en text-länk i regel är blå och understruken är det lätt att skilja den från ord som inte är länkar. Därför tror vi att ovanstående problem kunnat undvikas om ikonerna togs bort och enbart de

²⁶ Frontpage, Oracle WebServer, PL/SQL

förklarande texterna visats - då hade användaren sett att det inte är en länk. Det här är inget bestående problem, utan uppstår bara vid första kontakten med gränssnittet.

I Saab:s undersökning föreslogs att var och en av ikonerna i den vänstra kolumnen skulle leda till en förklarande rubrik i hjälpen. Anledningen till ett sådant förslag tror vi grundar sig på windows-standardens F1-funktion, där hjälp kan fås genom att man markerar det aktuella objektet och trycker på F1-tangenten. Eftersom denna möjlighet inte finns i webb-klienten bygger alltså förslaget på en negativ metafor. Om detta förslag realiserats hade det medfört att alla förstagångs användare i vår undersökning ofrivilligt hamnat i hjälpen när de försökt göra en stämpling!

6.2 Användarna

Vid gränssnittsutveckling måste man som designer vara medveten om att alla människor är olika. Vi tänker på olika sätt, reagerar på olika sätt och handlar på olika sätt. Därför är det svårt att utforma ett gränssnitt så att det skall passa alla användare av ett system. Som marknadsförare lär man sig att dela in människor i olika grupper. Genom att kategorisera människor efter deras ålder, kön, intressen, arbete, sociala samhällsklass etc är det lättare att rikta rätt marknadskampanj till rätt målgrupp. På liknande sätt kan man gruppera ett systems slutanvändare genom att dela in dem i olika användarkategorier. Dessa kategorier kan t ex vara experter, medelgoda användare och noviser. Att utforma ett system som passar bara en av dessa kategorier innebär sällan några större problem. Svårigheterna uppstår när systemet skall passa flera användargrupper, vilket är vanligt i de flesta fall. Är användarna vana vid att jobba med exempelvis ett windows-gränssnitt ställs högre krav, då de t ex vill ha menyer, kortkommandon och snabba svarstider. Ett sätt att tillfredsställa flera användarkategorier kan vara att erbjuda användaren olika åtkomstmetoder för att nå ett kommando, som t ex kortkommandon och mus. Den teknik som användes vid utvecklingen av IFS/Time & Attendance webb-klient innebar vissa begränsningar, vilket bl a medförde att webb-klienten tyvärr inte kunde förses med kortkommandon. Därför tror vi att expertanvändaren föredrar windows-klienten framför webb-klienten. Undersökningen med förstagångs användare visade att några av de som började rapportera i windows-miljön saknade möjligheten att utföra kortkommandon i webb-miljön. Enligt teorin bryr sig noviser vanligen inte om att använda sig av kortkommandon, men i vårt fall hade användarna tillgång till en lathund med funktionstangenter, vilket gjorde att de förmodligen använde kortkommandon i större utsträckning än vad de annars skulle ha gjort.

Vid utvecklingen av webb-klienten fastställdes ingen direkt användarkategori, även om man misstänkte att majoriteten av slutanvändarna skulle vara ovana. Det gjordes heller ingen användbarhetsanalys, kanske p g a att man inte kände till slutanvändarna. Gränssnittet utformades för att bli så enkelt som möjligt att använda, då tidrapportering är en uppgift som görs snabbt och rutinmässigt. Vi anser därför att gränssnittet främst passar sällan användare och ovana användare. Vi tycker ändå att någon form av användbarhetsanalys borde ha gjorts, om inte annat för funktionalitetens skull. Webb-klienten implementerades nämligen hos Gullspång utan att några föregående tester gjordes. Då man inte hade några konkreta slutanvändare att testa på, kunde förslagsvis en expertutvärdering varit på sin plats. Trots att inga

användbarhetsanalyser gjordes, lyckades man dock designa ett gränssnitt som föll användarna i smaken.

Det är inte bara kategoritillhörighet och individuella faktorer som skiljer användarna åt, utan även deras ändamål med systemet. Vissa användare använder IFS/Time & Attendance för att enbart godkänna sitt ordinarie schema, medan andra har behov av att rapportera övertid, göra tillägg, attestera etc. Därför finns det funktioner i systemet som inte utnyttjas av alla användare, vilket kan leda till att många användare känner sig vilsna bland för dem överflödiga information. Ett exempel på detta är att webb-klienten kompletterats med funktioner som endast berör servicemän, knappt hälften av Gullspåns anställda.

Utifrån tidigare beskrivna teorier, se **Figur 8**, s. 22, kan vi undersöka hur användbarheten, och därmed produktiviteten, hos Gullspåns användare påverkades vid implementationen av IFS/Time & Attendance webb-lösning.

- *Användaracceptansen* är relativt hög, eftersom man hade en positiv inställning till webb-lösningen och var motiverad att använda den. De som var mindre motiverade skyllde främst på saknade funktioner och att klienten till en början var alldeles för långsam. Vi upplever den allra första kontakten med gränssnittet som avgörande för hur användaren kommer att förhålla sig till systemet. Även om gränssnittet är tilltalande får användaren en mer fientlig inställning om denne i ett tidigt skede stöter på brister i funktionaliteten, vilket leder till en låg användaracceptans. När dessa fel senare åtgärdats är det ändå den första "bilden" av användandet som etsar sig fast hos användaren.
- Trots att den inledande utbildning som gavs, enligt några av användarna, inte var tillräckligt bra var *användarkompetensen* hög. Det beror troligtvis på att webb-klienten är relativt självinstruerande och att personerna i undersökningen hade intresset att själva pröva sig fram.
- *Användarvänligheten* kan delas upp i flera faktorer. Den mest uppenbara fördelen är här, och för alla system med webb-klienter, att åtkomsten genast ökar eftersom systemet är tillgängligt via Internet/Intranet. Individualisering kan, som vi vet, vara svår att uppnå. I IFS/Time & Attendance har användaren möjlighet att individuellt välja språk, samt tid- och datumformat. I webb-klienten är det den enda möjlighet till individualisering som finns. I windows-miljön har man större valfrihet och kan även välja exempelvis färger och vilka kolumner som skall visas. Ur individualiseringssynpunkt är det också en nackdel att webb-klienten innehåller funktioner som inte är avsedda för alla användare. Användarvänligheten höjs i det faktum att åtkomsten ökar, men problemen med individualisering, mental förenlighet och hjälpresurser kvarstår ändå.

För att förstå hur användaren skall involveras i systemutvecklingen måste vi fråga oss varför vi överhuvudtaget vill bygga ett system. Varför är vi beredda att satsa både tid och pengar på att implementera ett nytt system i verksamheten? Svaren på ovanstående frågor är självklara. Att datorisera en arbetsuppgift som tidigare varit manuell innebär nästan alltid tidsvinster och därmed kostnadsbesparingar. Hur stora tidsvinster och kostnadsbesparingar blir, d v s hur mycket produktiviteten ökar, beror på systemets slutliga funktionalitet och användbarhet. Dessa begrepp kompletterar

varandra. Dålig funktionalitet kan inte döljas av ett bra gränssnitt medan bra funktionalitet däremot kan döljas av ett dåligt gränssnitt. Ett dåligt gränssnitt kan skapa förvirring, frustration och panik hos slutanvändaren. Detta kan leda till att användaren struntar i att lära sig systemet och återgår till de gamla rutinerna eller att användaren endast lär sig de enklaste funktionerna i systemet och struntar i de mer komplicerade. I så fall blir systemet underutnyttjat och datorinvesteringen misslyckad.

Ett exempel på hur produktiviteten kan öka visar Saab:s utredning av IFS/Time & Attendance. Saab i Linköping har över 6 000 anställda vilka i genomsnitt lägger ner fem minuter varje vecka på att rapportera sin tid. Det innebär att Saab:s personal tillsammans lägger ner ca 500 timmar per vecka på att rapportera sin arbetade tid. Om de anställda kunde minska sin rapporteringstid med en minut per dag skulle en besparing om 5 200 timmar om året kunna göras. Då kan man fråga sig hur stora tidsbesparingar som egentligen skulle göras. Ovanstående tidsvinst innebär att två heltidstjänster kan dras in, vilket förmodligen inte skulle ske. Tidsvinsten som görs kan lätt jämnas ut genom att de anställda tar någon minuts längre rast varje dag.

Vi kan konstatera att tids- och kostnadsbesparingar vid en systemimplementation till stor del beror på hur slutanvändaren uppfattar det nya systemet och utför sina arbetsuppgifter. Alltså är slutanvändarens intryck av systemet avgörande för hur lyckad systeminvesteringen skall bli.

6.3 Principer och riktlinjer

Vi har tidigare konstaterat att det är viktigt att involvera användarna i designprocessen. Detta gjordes inte vid utvecklingen av IFS/Time & Attendance webb-klient. Eftersom det inte fanns några riktlinjer för hur webb-gränssnittet skulle utformas, fick designern fria händer att utforma gränssnittet. Då webb-utvecklingen i framtiden kommer att spridas inom IFS håller utvecklingsbolaget²⁷ på att utforma riktlinjer för denna. Tanken är att webb-gränssnittet skall utformas så att det liknar windows-gränssnittet så mycket som möjligt. Detta kan tänkas vara en stor fördel för de kunder som redan kör SQL/windows-klienten, medan det är av mindre betydelse för övriga användare. Man har dock inte tagit reda på om det är ett windowsliknande gränssnitt som användarna vill ha.

Som vi nämnt i teorin skall en skärmbild disponeras så att för mycket eller onödig information på skärmbilden undviks. En van användare av systemet kan dock hantera mer information på skärmen än vad en nybörjare kan. Det beror på att en van användare har lärt sig att selektera ut nödvändig information utan att störas av annan irrelevant information. I de fall det inte är av större betydelse hur mycket information som visas rör det sig snarare om en ren smaksak hos användarna. Exempelvis tyckte en av Gullspångs användare att det saknades viss information (totalt rapporterade timmar) på översiktsskärmen, medan en annan tyckte att onödigt mycket information visades här.

²⁷ IFS Research & Development

Om ett system inte är konsekvent rakt igenom kan förvirring lätt uppstå. Ett belägg för detta är om språkanvändningen skiljer sig i dialogerna så att det i ena fallet heter *delete* och i det andra *remove* när någonting skall tas bort. Ett annat exempel på inkonsekvens visar Saab:s undersökning där det påpekas att cancel-knappar saknas på felmeddelande-sidorna, något som annars finns på alla andra sidor i klienten.

Feedback är viktigt vid felhantering och när långa processer i systemet skall utföras. Användaren måste informeras om vad som händer och hur lång tid det kommer ta. Risken finns annars att användaren upprepar ett kommando i onödan. Detta visade sig stämma i vår undersökning med förstagångs användare, då de långa svarstiderna i webb-klienten fick en testperson att klicka på samma länk flera gånger innan sidan hann laddas om. Om muspekaren t ex tagit formen av ett timglas hade ovanstående problem kunnat undvikas. Det faktum att klienten är uppdelad på flera sidor gör att feedback ges efter varje avslutande moment genom att man kommer till en ny sida.

Idealet är att bygga ett system som är självinstruerande, så att användaren inte behöver läsa igenom någon användarmanual. För att uppnå målet med ett självinstruerande system kan man titta på hur användaren tidigare genomfört sina arbetsuppgifter. Om arbetsuppgifterna tidigare har skötts manuellt är det en bra idé att titta på de blanketter som tidigare använts. Kanske kan formuläret byggas så att det efterliknar blanketten så mycket som möjligt? Har användaren däremot tidigare använt ett helt annat system är det bra att titta på föregångarens layout och arbetsgång. Man bör då fråga användarna vad de tycker om det gamla systemet, vilka funktioner det har och vilka förbättringar som efterfrågas.

Det finns många fördelar med ett webb-gränssnitt. En av dessa är att man kan göra det lite ”roligare” än traditionella windows-gränssnitt. Det är dock viktigt att göra en delikat avvägning av var gränsen skall gå, så att gränssnittet inte blir tröttsamt att titta på i längden. En annan stor fördel med webb-gränssnittet är dess lättillgänglighet via Internet/Intranet.

6.4 Slutsatser

Syftet med uppsatsen var att utreda användarens betydelse vid gränssnittsdesign och att jämföra undersökningresultaten med beskrivna teorier. Vi kan konstatera att användaren har stor betydelse vid gränssnittsdesign. Vi kom fram till att de flesta teorier stämde i jämförelse med våra undersökningar.

Varje system är unikt beroende på ändamål, vilken verksamhet det avser, samt vilka de tilltänkta användarna är. Det är alltså omöjligt att rekommendera några specifika principer och riktlinjer utan att gå in på djupet i varje unik situation. Därför ger vi här några generella rekommendationer som, efter anpassningar, kan följas vid all gränssnittsdesign:

- Kunskaperna om MDI bör främst finnas hos de som utformar riktlinjer för design.
- Genom att ta hänsyn till människans minne, dess kapacitet och begränsningar, mentala modeller, metaforer etc, underlättas systemdesignerns uppgift att utveckla

ett användarvänligt gränssnitt. Ändå är det viktigaste inte att som designer vara expert inom MDI-området, utan att ha läggning för estetisk presentation vid val av färger, fonter etc.

- Ett konsekvent gränssnitt tillåter användarens automatiska nivå att generera en konsistent tolkning.
- Det räcker inte med att designern är duktig om man inte utgår ifrån användarna och deras behov.
- Ett bra gränssnitt kan uppfattas som dåligt om det hamnar hos fel slutanvändare.
- Första kontakten med gränssnittet kan vara avgörande för hur användaren kommer att uppfatta och förhålla sig till systemet.
- Slutanvändarens *intryck* av systemet är avgörande för hur lyckad systeminvesteringen skall bli.
- Det är viktigt att involvera användaren under hela systemutvecklingen, inte minst i början för att veta att man är på rätt spår.
- Ett dåligt gränssnitt kan leda till att felfrekvensen blir högre och att det tar längre tid att genomföra uppgiften än vad det borde göra.
- Ett gränssnitt som är byggt utifrån användarens behov och önskemål, d v s är användarvänligt, leder till att kostnaderna minskar medan effektiviteten och säkerheten ökar.

7 Vidare frågor

Under arbetets gång har vi stött på en rad frågor som kan utredas närmare.

- Hur bör hjälpfunktioner, användarmanualer och användarutbildningar utformas?
- Hur kan användbarhetstester göras för standardlösningar?
- Vad tjänar man, i tid och pengar, på att implementera ett affärssystem?
- Är webb-gränssnitt mjukvarumarknadens framtid eller bara en fluga?
- För vilka slags system är det lämpligt att utforma webb-klienter?

8 Källförteckning

8.1 Böcker

Allwood, C M (1991), *Människa-datorinteraktion – Ett psykologiskt perspektiv*, Studentlitteratur

Andrén, P m fl (1993), *Grafiska användargränssnitt – en utvecklingshandbok*, Studentlitteratur

Dumas, J S och Redish, J C (1993), *A practical guide to usability testing*, Ablex Publishing Corporation

Hägglund, S m fl (1989), *Dialogergonomi – Effektiv interaktion människa-dator*, Mekanförbundet

Lansdale, M W och Ormerod, T C (1994), *Understanding Interfaces – A Handbook of Human-Computer Dialogue*, Academic Press Limited

Löwgren, J (1993), *Human-computer Interaction – What every system developer should know*, Studentlitteratur

Norman D A, (1988), *The psychology of everyday things*, BasicBooks

Preece, J m fl (1993), *A guide to usability*, Addison-Wesley

Preece, J m fl (1994), *Human-computer Interaction*, Addison-Wesley

Shneiderman, B (1993), *Designing the user interface*, Addison-Wesley

8.2 Bibliografier

Seymor, D T (1992), *Marknadsundersökningar med kvalitativa metoder*, Probus Publishing Company

Wiedersheim-Paul, F och Eriksson, L T (1991), *Att utreda, forska och rapportera*, Liber-Hermods

8.2.1 Rapporter

Nygren, E (1991), *Utformning av användargränssnitt I – Erfarenheter, teorier och förslag till principer* (CMD-rapport nr 26/91), Uppsala universitet, Centrum för studier av människan och datorn

Nygren, E m fl (1991), *Utformning av användargränssnitt II – Förslag till en uppsättning av byggbara dialogelement. Implementationsmöjligheter och problem* (CMD-rapport nr 27/91), Uppsala universitet, Centrum för studier av människan och datorn

8.2.2 Opublicerat material

Boivie, I (1998), *Tida – Usability Evaluation Report*, Ericsson

Hoffman, E (1996), *Intresserad av gränssnittsutveckling?*, Institutionen för Informatik

Westerberg, S (1998), *Utvärdering Aida webgränssnitt*, Saab

8.2.3 Artiklar

Blomkvist, S (1995), *MDI i utvecklingen av ett grafiskt användargränssnitt för en läkemedelsdatabas*, www.csd.uu.se/~s92sbl/xjobb/xjobb2.html
Tillgänglig: 1998-04-03

Danielsson, L (1997), *Webben ger nya regler för gränssnitt*, Computer Sweden, Nr 54 1997

von Dorrien, C och Magnusson, L (1997), *Människa-Dator Interaktion – Föreläsningssanteckningar 2: Psykologiska teorier*, Chalmers Tekniska högskola och Göteborgs Universitet, Institutionen för datavetenskap, www.cs.chalmers.se/ComputingScience/Education/Course.../forelasning2.fm.html
Tillgänglig: 1998-03-30

Göransson, B (1997), *Användbarhetsutvärdering*, Technology Nexus AB, www.nexus.se/infosite/nexus-usability.html
Tillgänglig: 1998-04-03

Göransson, B (1997), *Användarcentrerad utveckling*, Technology Nexus AB, www.nexus.se/infosite/nexus-ucd.html
Tillgänglig: 1998-04-03

Göransson, B och Sandbäck, T (1997), *Användargränssnitt kräver bra design*, Technology Nexus AB, www.nexus.se/infosite/nexus-design.html
Tillgänglig: 1998-04-03

Lundhem, S, *Skärmtypografi: Färgtypografi*, CAP & Design, www.idg.se/capdesign/artikelsier/caphtmok/Farg.htm
Tillgänglig: 1998-03-30

Nygren, E (1997), *Grafiska användargränssnitt, några tips*, www.cmd.uu.se/papers/60/60.html
Tillgänglig: 1998-04-16

Rydberg, R (1997), *Slut med virrvarret på webben*, CAP & Design,
www.idg.se/capdesign/artikelsier/webdesign/webdesign1_97.html
Tillgänglig: 1998-03-30

Wiss, U (1996), *MDI och Distribuerad Multimedia – Föreläsninganteckningar*,
Lunds tekniska högskola, www.cdt.luth.se/pvt/courses/smd074/MDI/forelas.html
Tillgänglig: 1998-03-30

8.2.4 Intervjuer

Bergqvist, T (1998-04-08), Gullspång AB
Bryngelsson, C (1998-04-08), Gullspång AB
Grunander, H (1998-04-08), Gullspång AB
Larsson, M (1998-05-18), IFS AB
Molinder, M (1998-04-08), Gullspång AB
Olsson, I (1998-04-08), Gullspång AB

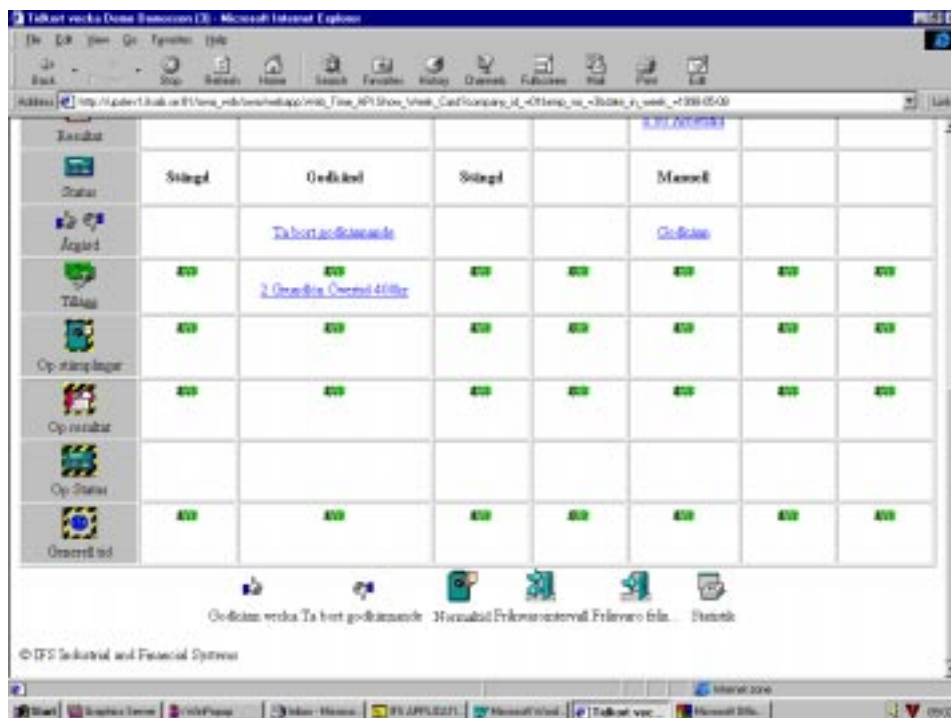
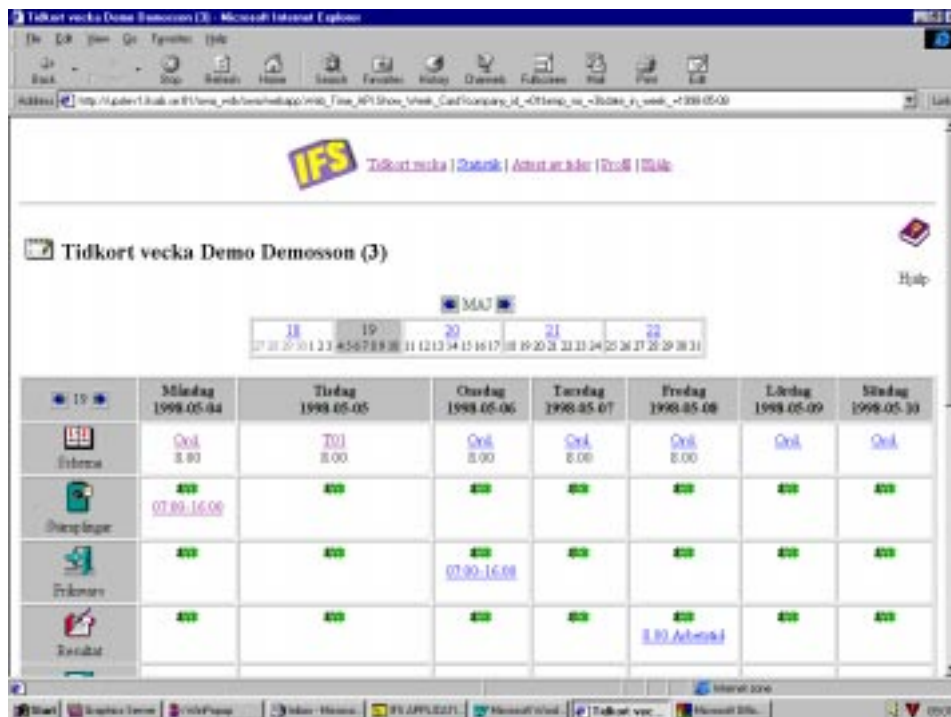
8.2.5 Testpersoner

Andersson, M (1998-04-07)
Carlsson, M (1998-04-07)
Hast, C (1998-04-07)
Liljedahl, H (1998-04-07)
Magnusson, F (1998-04-07)
Ringh, A (1998-04-07)
Samuelsson, M (1998-04-07)

Bilaga 1a - Gullspång

Beskrivning av IFS/Time & Attendance webb-klient för Gullspång.

Nedan ges en övergripande beskrivning av webb-gränssnittet hos Gullspångs användare. Så här ser det ut när användaren t ex skall rapportera och godkänna sin tid.



Bilderna ovan visar översiktsskärmen för tidrapportering (*Tidkort*). Som synes får inte hela layouten plats på skärmen, utan användaren måste scrolla för att se resten av översiktstabellen. Precis ovanför denna tabell finns en tidsöversikt där den vecka man tittar på är gråmarkerad. Här kan man navigera sig tidsmässigt genom att klicka på veckonumren eller bläddra sig framåt eller bakåt per månad.

Ikonerna i tabellens vänstra kolumn (med grå bakgrund) visar respektive rads innebörd. Dessa ikoner är dock inte länkar. De fyra sista raderna i tabellen (Op stämplingar o s v) visar de funktioner som standard-klienten kompletterats med och som bara utnyttjas av vissa användare. Ikonerna som är placerade under tabellen är länkar. Här kan man t ex klicka på länken *Frånvarointervall* för att registrera frånvaro för en längre tid.

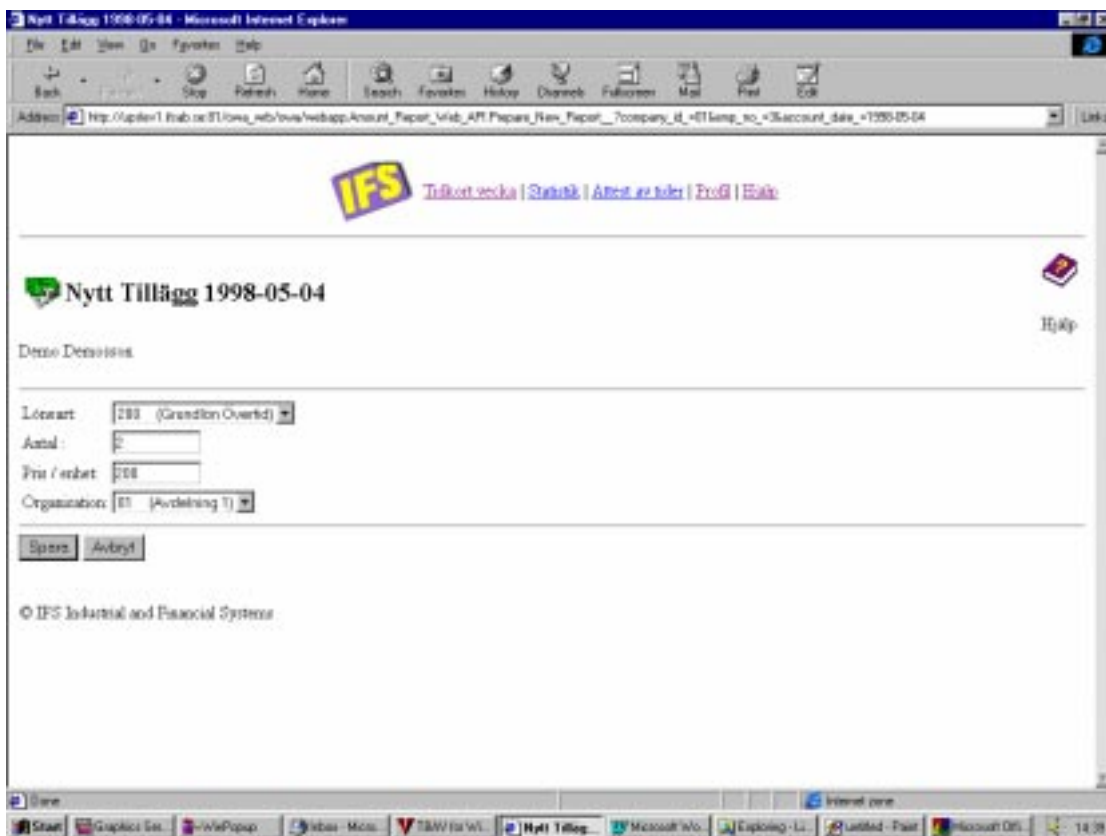


Länk till inmatningsdialog.

För att göra en ny registrering klickar man på en utav *Ny-ikonerna*, som är länkar, beroende på vad man skall registrera och för vilket datum. I det fall man väljer att göra en ny stämpling kommer man till följande inmatningsdialog:

The screenshot shows a web browser window with the title 'Ny stämpling 1998-05-04 - Microsoft Internet Explorer'. The address bar shows a URL starting with 'http://apiter1...'. The main content area displays the IFS logo and navigation links: 'Tidkort vecka | Statistik | Årstat av tider | Profil | Hjälp'. Below this is the title 'Ny stämpling 1998-05-04' and a 'Hjälp' icon. The form is titled 'Denno Dennotext' and contains the following fields:
- Instartstid: 1998-05-04 07:00
- Tidstyp: Normal
- Slutstämpling: 1998-05-04 16:00
- Tidstyp: Normal
- Organisation: E1 (Avtäckning T)
At the bottom of the form are 'Spas' and 'Avbryt' buttons. The footer of the page reads '© IFS Industrial and Financial Systems'. The taskbar at the bottom shows several open applications including 'Start', 'Graphics Set', 'WebPopup', 'Web - Mosa', 'T&W for W...', 'Ny stämpl...', 'Microsoft Wo...', 'Exploring - Li...', 'Unifed - Post', and 'Microsoft Off...'. The system clock shows 14:38.

Detta inmatningsformulär används alltså för att göra en ny stämpling. Här kan man inte ändra eller ta bort en stämpling, utan det görs på en liknande sida som man tar sig till via den tidlänk som visas under *Ny-ikonerna* på översiktsskärmen. Samtliga dialoger för inmatning ser ut på ungefär samma sätt, varför vi inte kommer att visa alla.



Ovanstående bild visar dialogen för inmatning av tillägg som exempelvis övertid eller OB-tillägg. I den översta komboboxen väljer man löneart för det tillägg man skall göra.

Gemensamt för alla skärmbilderna är att det längst upp till höger finns en ikon som är en länk till hjälpfunktionen. När man klickar på denna öppnas hjälpen i en ny webbläsare. En sk *header* återfinns dessutom överst på varje sida, som innehåller IFS logotype, samt några länkar till andra delar i webb-klienten.

Bilaga 1b - Frågeformulär

Gullspångs användare

Ålder:

Utbildning:

Tidigare arbeten:

Nuvarande arbetsuppgifter:

Datorvana: (lite) 1 2 3 4 5 (mycket)

1. Hur har tidrapporteringen skett förut?
2. Känner du igen upplägget (sättet att rapportera)?
3. Kör du windows-applikationen också?
4. Hur länge har du använt dig av den här web-klienten?
5. Vilken typ av utbildning/genomgång fick du inför användandet av web-gränssnittet?
6. Hur tyckte du var att lära sig systemet?
(lätt) 1 2 3 4 5 (svårt)
7. Vilka intryck fick du första gången du körde systemet? (adjektiv)
8. Hur tycker du det är att utföra rapporteringen?
(lätt) 1 2 3 4 5 (svårt)
9. I jämförelse med det tidigare rapporteringssättet – har din tidrapportering effektiviserats?
10. Vad anser du om hjälpen, är den lätt att navigera sig i och förstå?
11. Ser du den information du behöver på de olika skärmbilderna? Saknas något?
12. Vad tycker du om bilderna (ikonerna)?
13. Förstår du bildernas innebörd?
14. Förstår du vilka bilder som är länkar?
15. Hur upplever du färgerna?
16. Förstår du hur du navigerar tidsmässigt?
17. Finns det behov av att förflytta sig mer än två veckor framåt/bakåt i tiden?

18. Passar gränssnittet på din skärm? (Får du scrollera mycket)
19. För att logga in som en annan användare måste man stänga ner web-browser. Är detta ett problem?
20. Vad tyckte du var bäst?
21. Vad tyckte du var sämst?
22. Förslag till förändring/förbättring.

Användarprofiler:

Person 1:

Ålder: 54
Utbildning: El
Tidigare arbeten: Eldistributör
Nuvarande arbetsuppgifter: Säljare
Datorvana: (lite) 1 2 3 4 5 (mycket)

Person 2:

Ålder: 31
Utbildning: 4-årig teknisk linje
Tidigare arbeten:
Nuvarande arbetsuppgifter: Driftledare
Datorvana: (lite) 1 2 3 4 5 (mycket)

Person 3:

Ålder: 45
Utbildning:
Tidigare arbeten: Gullspång
Nuvarande arbetsuppgifter: Fakturering
Datorvana: (lite) 1 2 3 4 5 (mycket)

Person 4:

Ålder: 24
Utbildning: Elektroingenjör
Tidigare arbeten:
Nuvarande arbetsuppgifter: IT- och systemansvarig
Datorvana: (lite) 1 2 3 4 5 (mycket)

Person 5:

Ålder: 31

Utbildning: 4-årig teknisk linje + Extra ekonomistudier

Tidigare arbeten: Service mobiltelefoner, Planeringsansvar Regionnät

Nuvarande arbetsuppgifter: Planeringsansvar Lokalnät

Datorvana: (lite) 1 2 3 4 5 (mycket)

Bilaga 2a - Förstagångsanvändare

Beskrivning av IFS/Time & Attendance webb-klient - standard



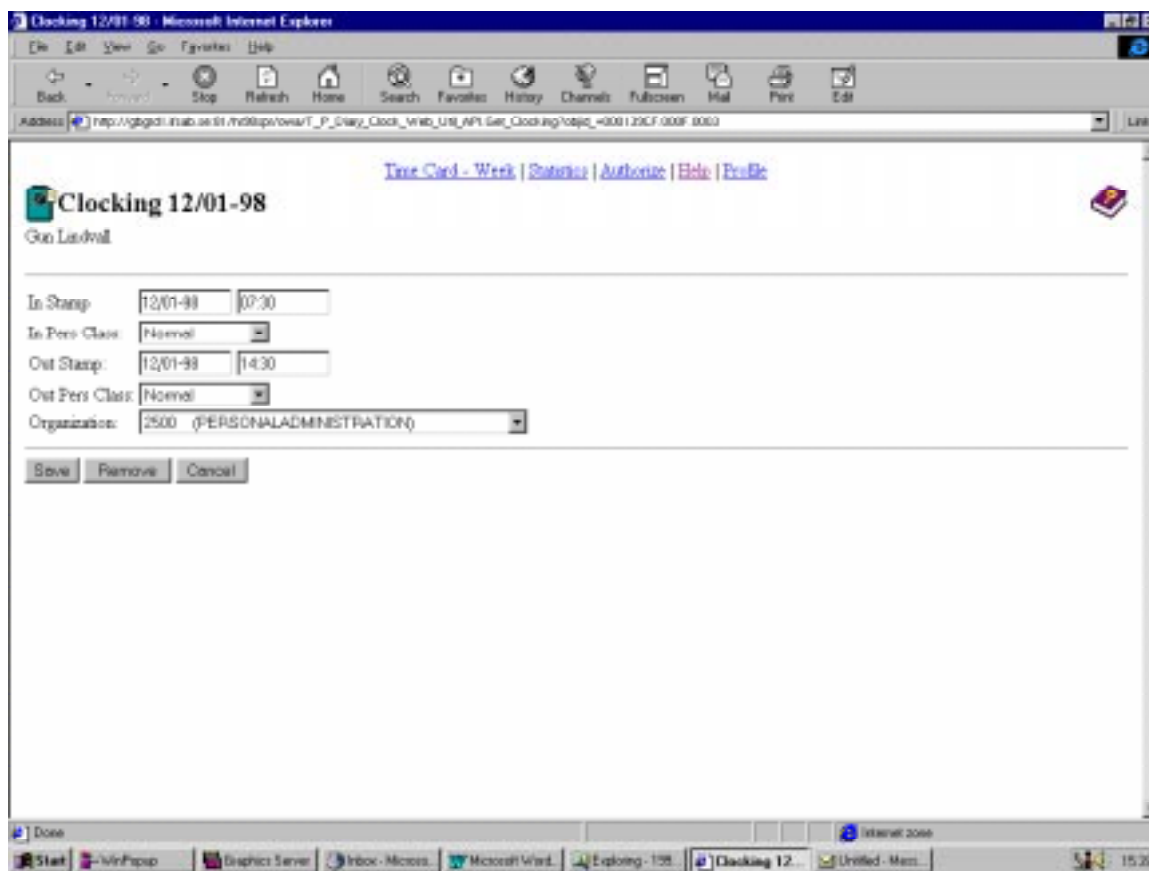
Användarna utgår från ovanstående översigtsbild när de skall rapportera sin arbetade tid. Överst finns en *header* med länkar till funktioner som statistik, attest, hjälp etc. Headern, liksom länken till hjälpfunktionen, finns på alla av webb-klientens sidor för att användaren snabbt skall kunna navigera sig genom gränssnittet. Headerns länkar och hjälpfunktionen kommer inte att förklaras närmare. Under headern visas användarnamnet på den person som är inloggad för tillfället. Därunder finns tidnavigatören där man väljer den månad och vecka tiden skall rapporteras på.

Tidrapporteringen visas veckovis med varje dags stämpling synlig. I den vänstra kolumnen finns ett antal ikoner som representerar varje rads funktion. Dessa ikoner är inte länkar. Vi koncentrerar oss kring *Clockings* (stämplingar), *Absence* (frånvaro) och *Operation* (bl a godkännande). Vi kommer inte att beskriva de ikoner som ligger långt ner i formuläret.

För att rapportera sin tid skall man klicka på *New-ikonen* i raden *Clockings* för respektive dag. Ikonen ser ut så här:

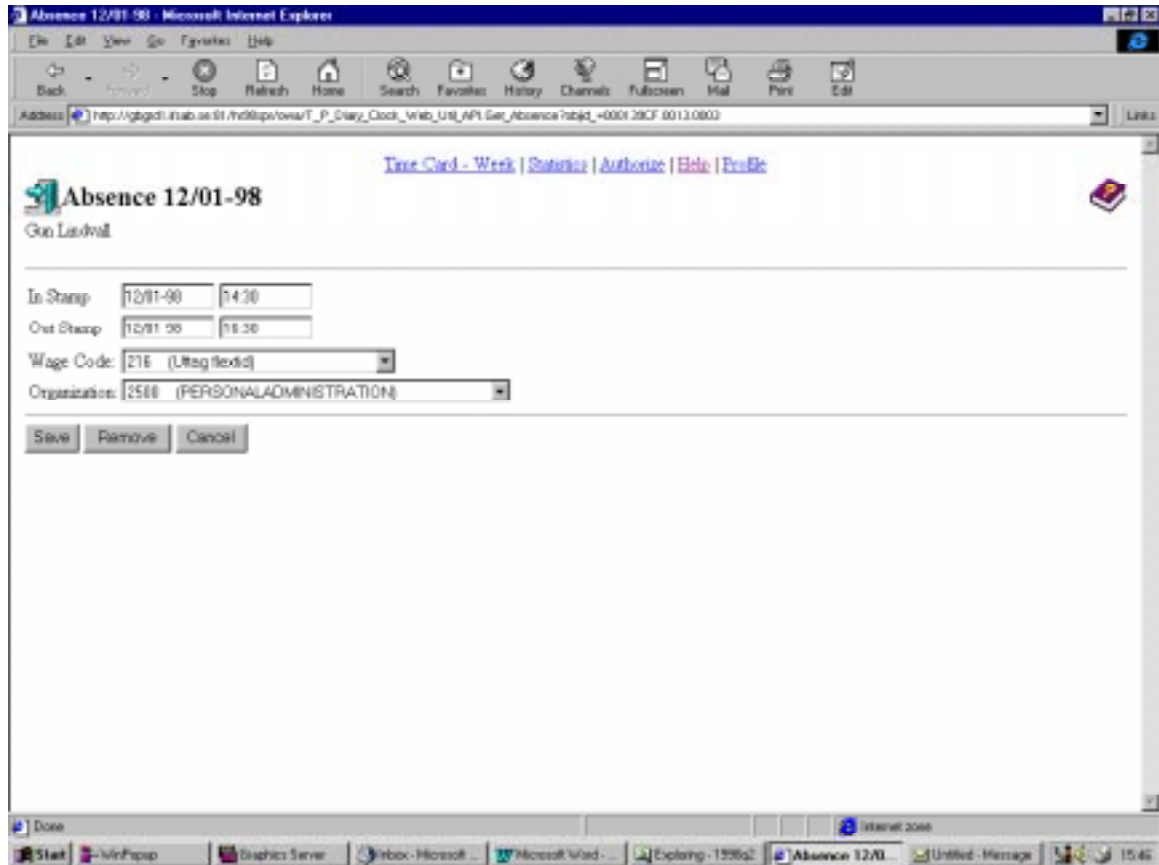


Ikonen är en länk till följande inmatningsformulär:



Här skall den arbetade tiden matas in. Under headern finns datumet för den aktuella dagen, samt användaridentiteten som en kontroll för användaren. *In Stamp* och *Out Stamp* är redan ifyllda med tider enligt användarens ordinarie schema. Om den arbetade tiden inte stämmer överens med schemat ändrar användaren till de korrekta värdena i inmatningsrutorna. Om tidtypen skall vara en annan än normal eller om organisationskoden skall vara en annan, väljer användaren de rätta värdena i komboboxarna. Användaren har tre valmöjligheter i knappraden, *Save* (spara), *Remove* (ta bort) eller *Cancel* (avbryt). Alla alternativ för användaren tillbaka till översiktsskärmen. Om det har gjorts en ny stämpling, eller någon ändring, visas den nu där.

När användaren skall rapportera sin frånvaro får inga överlappningar ske. Om dagens rapporterade tid är 07:30 till 16:30 och frånvaro skall rapporteras från och med 14:30, måste användaren först ändra sin tidigare Outstamp till 14:30. Ändringen av den tidigare stämplingen görs genom att användaren klickar på den tidlänk som finns i raden Clockings det aktuella datumet. Då visas samma bild som ovan och användaren gör här sina ändringar. Själva frånvaron rapporteras genom att användaren klickar på new-ikonen på raden för Absence. Då visas dialogen där frånvaro rapporteras. Dialogen ser ut så här:



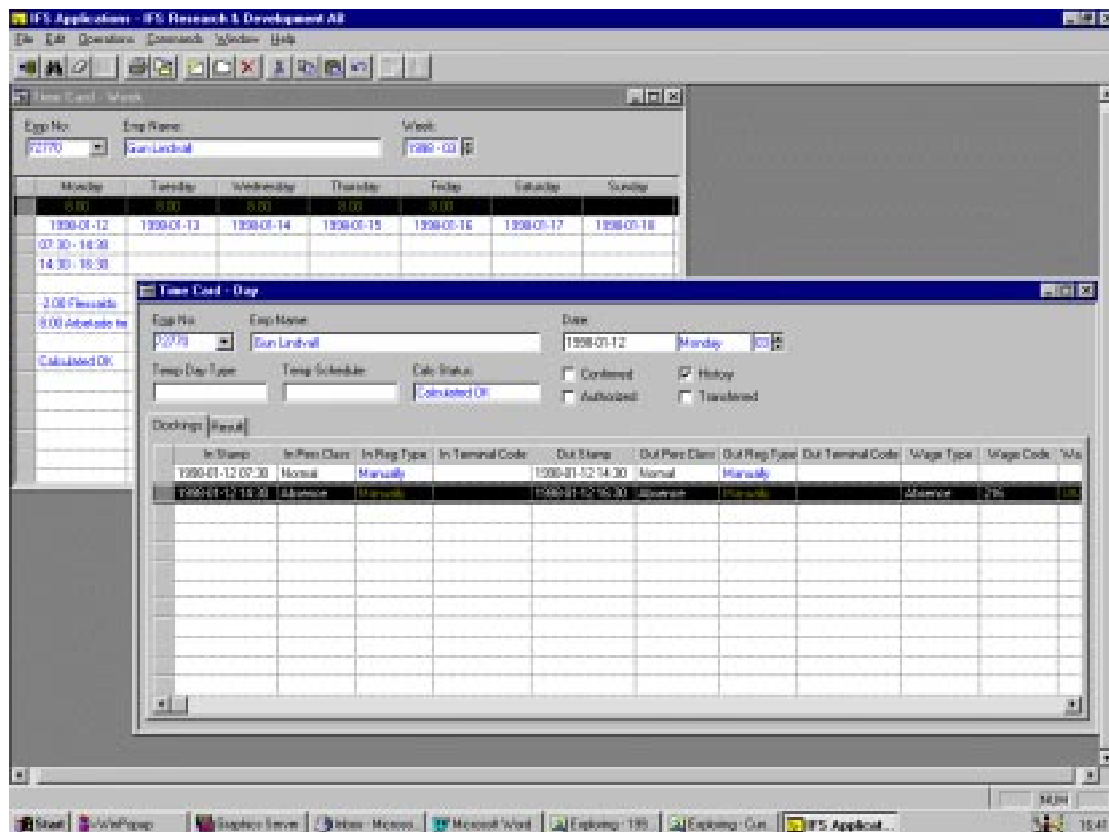
På liknande sätt som den första stämplingen gjordes matar användaren in sin frånvaro.

Vi kan konstatera att frånvaron resulterar i två stämplingar, dels Clockings med personens närvaro och dels Absence med personens frånvaro.

För att godkänna sin rapporterade tid skall användaren klicka på länken Confirm som finns i raden Operations. I dialogen som visas behöver användaren bara trycka på OK-knappen för att godkänna sin tid.

För att ta bort sina stämplingar skall man från översiktsskärmbilden gå in på respektive tidlänk och trycka på knappen Remove. Ta bort görs således i samma dialogruta som stämpling och ändring.

Beskrivning av IFS/Time & Attendance SQL/windows-klient



Ovanstående skärmdokument visar Tidkort vecka och Tidkort dag. Tidkort vecka är en översiktsbild som visar en hel veckas tidrapportering. Genom att klicka på respektive dag, kommer man till Tidkort dag där arbetad tid matas in. Man kan även välja Tidkort Dag direkt i *Navigatorn*, jämför Windows Explorer (utforskaren). Överst på Tidkort dag finns ett *huvud* som innehåller den anställdes namn, anställningsnummer, tidnavigator etc. Huvudet finns hela tiden synligt när man rapporterar sin tid.

För att förstå arbetsgången är det nödvändigt att veta innebörden av vissa ikoner och dess funktioner. Ett kommando kan nås via menyer och de mest frekventa även via toolbaren. Vissa kommandon kan också utföras via snabb- och funktionstangenter eller höger musknapp.

Det första användaren måste göra är att hämta sina uppgifter. Det görs för att personen skall kunna se sina inmatade uppgifter. Ikonen för att hämta uppgifter ser ut så här:



Om en ny rad skall läggas till initieras detta genom att användaren klickar på New-knappen som ser ut så här:



Vissa värden hämtas via en s k värdelista. En värdelista innehåller de fördefinierade alternativ som kan väljas i vissa av inmatningskolumnerna. Genom att stå i rätt fält och klicka på följande ikon,



kommer det fram en lista med de för fältet godkända inmatningsvärden. En sådan lista kan se ut så här:

Wage Code	Wage Name	Def Wage Type	Transfer
005	tom 960930 Extra tillägg	Increment	No transfer
007	tom 960930 Månadslön	Normal	No transfer
015	tom 960930 Utryckning = AV1	Increment	No transfer
052	tom 960930 Ord semester	Absence	No transfer
053	tom 960930 Sparad semester	Absence	No transfer
070	Arbetade timmar	Normal	No transfer
075	Arbetade timmar	Normal	No transfer
101	OB-Tillägg 1 (16-22)	Increment	Transfer grouped
102	OB-Tillägg 2 (22-0030)	Increment	Transfer grouped
103	OB-Tillägg 3 (00-06)	Increment	Transfer grouped
121	tom 951231 OB-sjuk 1 (4-14)	Increment	No transfer
122	tom 951231 OB-sjuk 2 (4-14)	Increment	No transfer
131	OB-Tillägg 1 sjuk (2-28)	Increment	Transfer day by d
132	OB-Tillägg 2 sjuk (2-28)	Increment	Transfer day by d

Användaren godkänner sina uppgifter genom att antingen via höger musknapp eller rullgardinsmenyn Commands välja Confirm. Om en dag är godkänd visas det i en kryssruta i headern:

<input checked="" type="checkbox"/> Confirmed	<input type="checkbox"/> History
<input type="checkbox"/> Authorized	<input type="checkbox"/> Transferred

För att ta bort en inmatad rad skall användaren markeraraden och klicka på ta-bort-knappen:



Genom att klicka på Save-knappen sparar användaren alla sina uppgifter och ändringar:



Bilaga 2b - Instruktionsformulär

Test av tidrapportering – användarnas exemplar

Förklaring av menyn och ikoner

Kommandon kan du finna antingen i menyn, höger musknapp eller via knapparna i toolbaren.

Hämta alla (*populate*), för att få fram personens tidsuppgifter.

Värdelista (*list of values*), för att välja ett värde ur en lista.

Bortse från fliken Result.

Tänk på...

Efter varje rapportering/ändring måste man spara.

Tänk på att stå med markören i rätt fält för att få fram rätt menyval.

Uppgift

1. Gör en ny stämpling (*clocking*) den 6/4 mellan 08.00 och 16.00.
In Stamp (Time Type In) = normal
Out Stamp (Time Type Out) = normal
2. Ändra stämplingen du nyss gjorde till outstamp = 14:00. Rapportera in frånvaro (*absence*) mellan 14.00 och 16.00.
Out Pers Class (i windows) = normal
Wage Type = hämta *absence* i värdelistan
Wage Code = hämta *216* i värdelistan
Organization Code = hämta *2500* i värdelistan
3. Godkänn stämplingen (*confirm*).
4. Ta bort stämplingen – både rapporteringen och frånvaron (delete alt. *remove*).

Test av tidrapportering – utvärderarnas exemplar

Login: test3, test4

Databas: test520

Förklaring av menyn och ikoner

Kommandon kan du finna antingen i menyn, höger musknapp eller via knapparna i toolbaren.

Hämta alla (*populate*) - för att få fram personens tidsuppgifter.

Värdelista (*list of values*) – välj ett värde här när du står i ett fält och inte vet vad du skall fylla i.

Bortse från fliken Result.

Tänk på...

Efter varje rapportering/ändring måste man spara.

Tänk på att stå med markören i rätt fält för att få fram rätt menyval.

Frågor till användaren

1. Vilken miljö uppfattar du som
 - a) enklast
 - b) snabbast
 - c) mest överblickbar
2. Vilken miljö föredrar du?
3. Varför då?

Kolla upp:

1. Vilka fel gör användaren?
2. Hur lång tid tar rapporteringen?
3. Vilka svårigheter stöter användaren på?

