



LUNDS
UNIVERSITET

INSTITUTIONEN FÖR PSYKOLOGI

***Användbarhetstest av Beräkningsverktyg för Användning vid
Design och Konstruktion av Motorer***

Patrik Andersson

Kandidatuppsats ht 2010

Handledare: Magnus R Larsson

Sammanfattning

Denna studie beskriver ett test av användbarheten hos beräkningsverktyget PC-OSKAR. Detta är ett av företaget ABB egenutvecklat datorprogram för design och konstruktion av motorer. Syftet med testet var att identifiera användbarhetsproblem hos PC-OSKAR. Testet bestod i att fem stycken deltagare fick sju stycken representativa arbetsuppgifter. Uppgifterna utvärderades med Performance Measurement Method (PMM), och en operationalisering av nyckelbegreppen i definitionen av användbarhet i ISO 9241 kapitel 11, dvs. *Effectiveness*, *Efficiency* och *Satisfaction*. PC-OSKAR uppvisade ett väldigt bra resultat när det gällde beräknad *Effectiveness*. Med några undantag var poängen hög. Uppgifterna genomfördes fullständigt och korrekt i mer än 60 % av fallen, och vid tolv tillfällen (av 35) var resultatet helt korrekt. Beräknad *Efficiency*, som kunde varit bättre, återspeglade att deltagarna fick ägna mycket tid åt de olika parametrarna i PC-OSKAR. Det framgick relativt tydligt att PC-OSKAR ifråga om deltagarnas attityd, dvs. *Satisfaction*, vilket mättes med frågeformuläret Software Usability Measurement Inventory (SUMI), lämnade en del att önska. Intervjuerna av deltagarna visade ändå att deltagarna fann PC-OSKAR vara ett stort framsteg jämfört med andra system. Studien visade att framtida utveckling bör fokusera på hanteringen av parametrar, generell layout och interaktion, men framförallt på att få till stånd en användarcentrerad utvecklingsprocess.

Nyckelord: Användbarhet, beräkningsverktyg, *Effectiveness*, *Efficiency*, ISO 9241, ISO 13407, Performance Measurement Method, *Satisfaction*, SUMI

Introduktion

Syftet med denna studie var att testa användbarheten hos beräkningsverktyget PC-OSKAR. PC-OSKAR är ett av ABB egenutvecklat datorprogram. PC-OSKAR används av ingenjörerna på ABB vid design och konstruktion av motorer. Ingenjörerna som använder PC-OSKAR är inte alltid i första hand professionella datoranvändare utan snarare experter på ett arbete som kräver datorstöd. På samma sätt som ett verktyg som t.ex. en hammare måste utformas olika beroende på om det skall användas till silversmide, järnsmide eller finsnickeri, bör ett verktyg i form av ett datorprogram också utformas olika beroende på användarna av programmet, uppgift och sammanhang.

Vad är användbarhet?

Användbarhet är ett begrepp som inom Human Factors forskningen är kopplat till användarcentrerad design och utveckling (Bailey, 1989). Vad begreppet användbarhet innebär finns det många olika definitioner på. Nielsen (1993) definierar användbarhet i termer av att det skall vara effektivt att använda, lätt att lära, lätt att komma ihåg, få fel och subjektivt tilltalande. Schneiders (1998) definition liknar Niensens (1993) och skiljer sig egentligen bara åt när det gäller ordvalet. Schneiderman (1998) definierar användbarhet i termer av effektiv användning, tid att lära, komma ihåg över tid, antal användnings fel, och subjektivt tilltalande. Den internationella standarden ISO 9241 Ergonomic requirements for office work with visual display terminals (VDTs) definierar i kapitel 11 användbarhet på följande sätt:

"Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (SIS, 1999, s.2)

Gulliksen och Göransson (2002) menar att definitionen i ISO 9241 kapitel 11 är tillämpbar i många projekt eftersom den är konkret och den definierar användbarhet som en mätbar storhet där det går att uttrycka i kvantitet hur mycket någonting är användbart. Definitionen är allmänt erkänd och gör det därför också möjligt att diskutera användbarhet med en gemensam förståelse för begreppet. Definitionen av användbarhet i ISO 9241 kapitel 11 kan konkretiseras och göras mätbar med avseende på nyckelbegreppen i definitionen, dvs. *Effectiveness*, *Efficiency* och *Satisfaction*. Värt att notera är att ISO 9241 kapitel 11 lägger stor tonvikt på produktens egenskaper, men även på den kontext som produkten kommer att användas i (SIS, 1999).

Kostnaden för dålig användbarhet

Bias och Mayhew (1994) menar att kostnaden för dålig användbarhet hos de system som används ute på företagen uppgår till ansenliga belopp. I denna kostnad ingår inte enbart formellt redovisade kostnader för t.ex. omarbetningar, kompletteringar mm. utan även de oredovisade kostnader, vilka kan vara betydande, som en följd av att chefer, utvecklare, utbildare m.fl. får ägna en hel del tid åt verksamhet som egentligen har sin grund i bristande användbarhet t.ex. diskussioner med kunderna, ändring av design och konstruktion, sammanträden. Sparre och Andersson (1994) anser att det finns ett samband mellan pris och kvalitet och att priset för dålig kvalitet är på mellan 10-30 procent av försäljningsintäkten. Sparre m.fl. (1994) menar också att bättre kvalitet inkluderande användbarhet, bl.a. leder till att ett högre pris kan tas ut, och att företag med bättre kvalitet och större marknadsandel ger en vinst som är ungefär fem gånger större än företag med sämre kvalitet och mindre marknadsandel.

Exempel på bra och dålig användbarhet

Det finns många exempel på att det lönar sig för företag att aktivt arbeta med användbarhet. Nielsen (1993, s.3) beskriver hur ett datorföretag sparade 42 000 dollar första dagen genom att de designade om hur man loggade in på ett visst system. Det finns också många exempel på företag som inte satsat på att arbeta med användbarhet och vad det fått för konsekvenser. Preece, Sharp, Denyoun, Holland & Carey (1994) beskriver hur ett indiskt flygplan 1990 kraschade pga. att flygbolaget menade att det var piloterna som skulle anpassa sig efter systemet och inte tvärtom. Kontogiannis och Embrey (1997) skriver i en analys av datorbaserade processsystem att mänskliga fel under användningen av systemen var associerade med nästan 60 % av de olyckor som inträffade. Det konstaterades att en av de mest signifikanta orsakerna till detta var att processinformation inte visats på ett adekvat sätt. Sparre m.fl. (1994) beskriver hur man inom ABB har gjort ett stort antal användbarhetstest och inspektioner av en mängd olika typer av system för att försöka identifiera problemen innan de blir till verkliga problem, se figur 1. I genomsnitt fann man vid användbarhetstesten 70 olika problem vid 230 olika observationer. Vid inspektioner då man utifrån ett antal heuristiska principer som kännetecknar en design med hög användbarhet fann man i snitt 80 problem vid varje inspektion. Enligt Nielsen (1993) används inspektioner oftast som ett komplement till användbarhetstest för att identifiera mer lokala och ”kosmetiska” problem som t.ex. felaktiga färger och svårförståeliga ikoner.

ABB Usability Tests 91-95

Robot panel prototype
Display Builder
Display Builder B
Robot panel
Display Builder C Engineering Tool
Maintenance Tool
Maintenance Tool & Panel Operator station
Motor Design Tool

Average of 230 observations of 70 problems

ABB Heuristic Inspections 91-95

Offshore Application Display Builder A Visual Language Tool
Maintenance Tool
Formtek (PDM) Nuclear Fuel Design Project Support System

Average 80 findings/inspection

Figur 1. *Exempel på användbarhetstest och inspektioner vid ABB (Sparre m.fl., 1994, s.3).*

Dålig användbarhet, varför?

Vad är då anledningen till dålig användbarhet och att många företag gör fel vid utvecklingen av olika typer av system? Ofta beror dålig användbarhet hos ett system på att en del företag har fel fokus, sätter inte tidigt upp användbarhetsmål, eller har en bristande kompetens när det gäller behovet av användbarhet och hur man integrerar användbarhet i sin utvecklingsprocess.

Fel fokus

Företag är ofta mer fokuserade på teknik och nya funktioner än på användbarhet när det tilldelas resurser i form av bemanning osv. (Gulliksen m.fl., 2002; Kontogiannis m.fl., 1997). Många företag ser inte användbarhet som en del av den totala systemkvaliteten tillsammans med prestanda osv. Användbarhet menar företagen är något som de kan lägga tid och resurser på om det blir tid och pengar över. Gulliksen m.fl. (2002) studie visar att när det gäller budgeten för användbarhetsrelaterat arbete är andelen oftast inte mer än en procent av den totala utvecklingsbudgeten. Att det blir så här beror ofta på prioriteringar hos ledningen, men också på bristande kunskap om hur man bäst arbetar med användbarhet, t.ex. att man tidigt i utvecklingsprojekt bör sätta tydliga och mätbara användbarhetsmål.

Användbarhetsmål

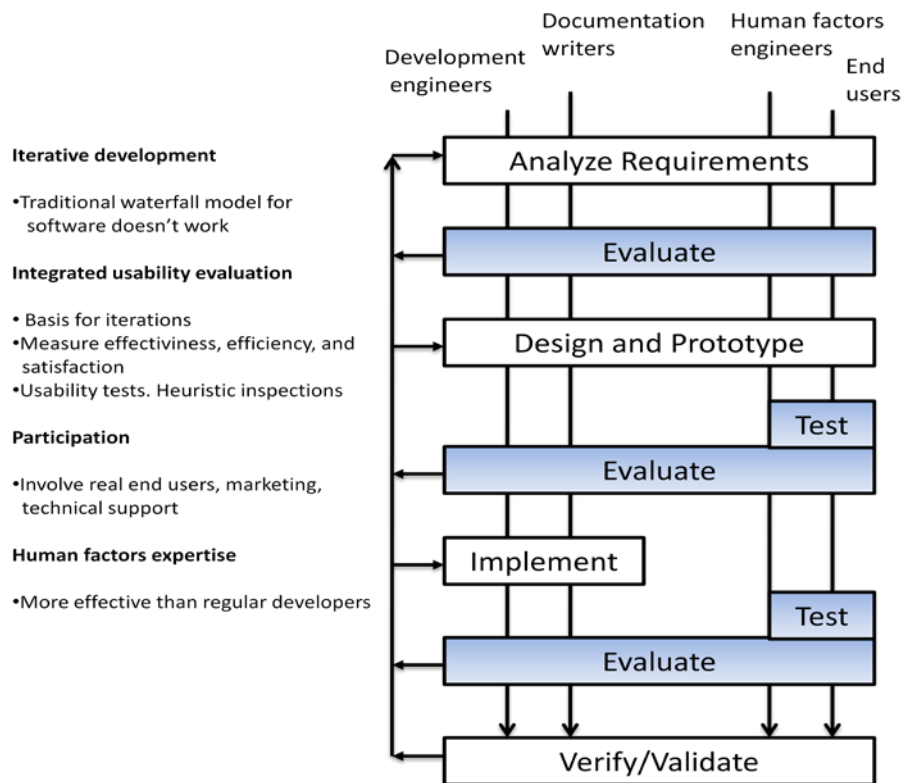
Enligt Rexfelt och Rosenblad (2006) finns det många studier som visar att det är en nödvändig förutsättning i utvecklingsprojekt med klara och välformulerade användbarhetsmål för att kunna utveckla användbara datorsystem men ändå saknas ofta dessa. Gulliksen m.fl. (2002) argumenterar för att det är möjligt att använda ISO 9241 kapitel 11 för att bestämma användbarhetsmål. Genom att specificera användbarhetsmål tidigt i utvecklingsprocessen är det möjligt för utvecklarna av systemet att få vägledning vid design av systemet. Men vad är då ett användbarhetsmål? Några exempel på användbarhetsmål är att systemet skall vara enkelt att använda, systemet skall vara effektivt att använda, osv.

Bristande kompetens

Gulliksen m.fl.(2002) skriver i sin studie hur vanligt det är att beställaren under upphandlingen av ett nytt system inte alltid inser sin bristande kompetens speciellt när det gäller användbarhet och hur man integrerar användbarhet i utvecklingsprocessen. Ofta förstår inte beställaren heller behovet av att använda sig av experter på användbarhet. Detta leder till att det är oftast snarare är en regel än ett undantag att olika typer av IT-system under leveransen inte motsvarar kundens förväntningar (Nielsen, 1993).

Vad är lösningen på dålig användbarhet?

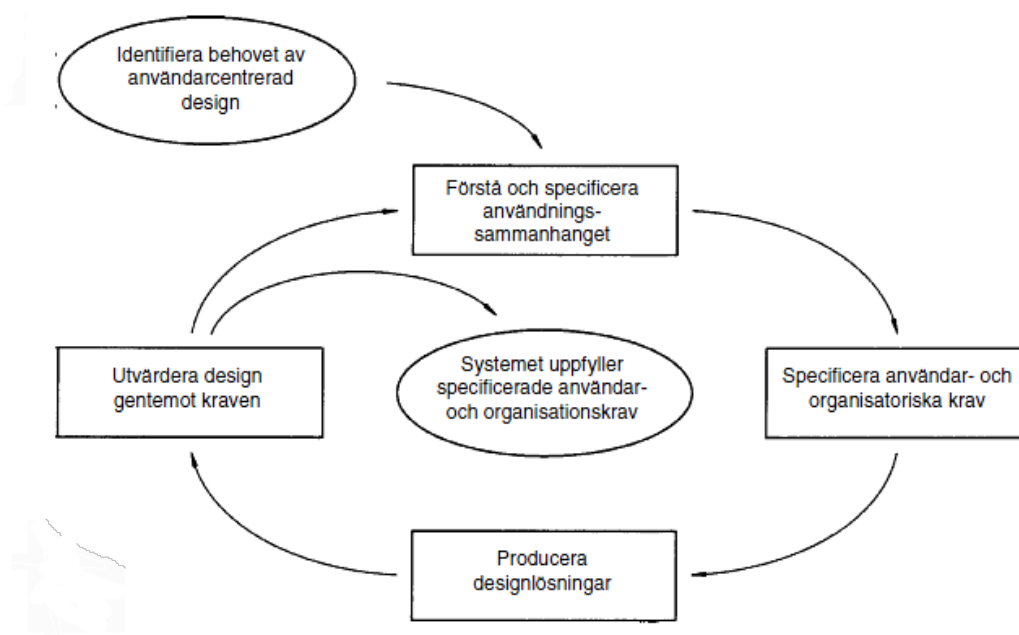
Idag förespråkar de flesta experter en användarcentrerad utvecklingsmodell vid design och utveckling, se figur 2. I den användarcentrerade utvecklingsmodellen är användarna i fokus. Användarna arbetar aktivt från början i utvecklingsprocessen tillsammans med olika experter på t.ex. användbarhet. Ett iterativt arbetssätt och utvecklandet av prototyper vilka regelbundet testas och med kontinuerlig återkoppling till utvecklarna är av största vikt vid en användarcentrerad design och utveckling. (Bias m.fl., 1994; Ehrlich, Butler & Pernice 1994; Gulliksen m.fl., 2002; Gould & Lewis, 1995; Ottersten & Berndtsson, 2002)



Figur 2. Generell användarcentrerad utvecklingsmodell (Sparre m.fl., 1994, s.5).

Användarcentrerad design

Den processororienterade standarden ISO 13407 innehåller flera användarcentrerade design aktiviteter som kan användas som en grund i utvecklingsprojekt som har som mål att arbeta fokuserat på användaren (SIS, 1999). De användarcentrerade design aktiviteterna bör börja i de tidigaste stadierna i ett utvecklingsprojekt och repeteras iterativt till dess att systemet uppfyller kraven, se figur 3. ISO 13407 beskriver fyra punkter som ligger till grund för definitionen av användarcentrerad design; aktiv involvering av användare och en tydlig förståelse av användarens och uppgiftens krav, en lämplig allokering av funktion mellan användare och teknik, iterering av designlösningar, och tvärdisciplinär design (SIS,1999). Earthy, Jones & Bevan (2001) sammanfattar ISO 13407 som ett ramverk som förser utvecklarna med bl.a. riktlinjer för hur användarcentrerad design och utveckling bör bedrivas. Abran, Khelifi, Suryan & Seffah (2003) utvecklar detta och menar att ISO 13407 beskriver användarcentrerad design som en multidisciplinär aktivitet som förenar mänskliga faktorer och ergonomi.



Figur 3. Användarcentrering enligt ISO 13407 (SIS, 1999, s.10).

Aktiv involvering av användarna

Gulliksen m.fl. (2002) menar precis som ISO 13407 att användarna kan ge feedback väldigt tidigt i utvecklingsprocessen bara de tillåts vara aktiva och får vara med från början i utvecklingsprocessen. Preece m.fl. (1994) anser att ett aktivt deltagande av användarna under utvecklingsprocessen kan minska gapet mellan den mentala modellen hos dem som designar systemet och användarna. För att lyckas utveckla användbara system är det således av största vikt att användarna hela tiden är i fokus. Benyon, Turner & Turner (2005) utvecklar resonemanget och menar att användarna skiljer sig åt med avseende på både fysiska, psykologiska, och konceptuella egenskaper och genom att ha tidig och aktiv medverkan från användarna av systemet uppnås en bättre förståelse om hur teknologin kan designas för att täcka dessa olika egenskaper.

Iterativ utvecklingsprocess

Traditionell utveckling följer i regel en vattenfallsmodell där processen är linjär vilket innebär att de olika stegen påbörjas när föregående steg är avslutad. Preece m.fl. (1994) menar att ett problem med vattenfallsmodellen är att den är strikt och det finns ingen återvändo från ett steg till det föregående. Det avklarade steget kanske behöver ändras och förbättras, men eftersom steget är klart är förändringen någonting som måste göras i utvecklingsprocessens slutfas, och ändringar i ett projekts slutfas innebär i regel enorma kostnader jämfört med om dessa ändringar gjorts tidigare i

utvecklingsprocessen. Den iterativa utvecklingsprocessen som rekommenderas bl.a. i ISO 13407 är däremot cyklisk och inte linjär (SIS, 1999). Mandel (1997) menar att alla framgångsrika designprocesser måste vara iterativa vilket innebär att det aldrig går att producera framgångsrika system utan att med jämna mellanrum gå tillbaka och kontrollera med användarna. Att iterera betyder således att göra någonting om och om igen, se figur 2.

Prototyping

Innan utvecklandet av det skarpa systemet kan prototyper användas för att uppnå användbara system genom att man med hjälp av prototypen kan göra en utvärdering av användbarheten och snabbt göra nödvändiga förändringar utan att kostnaden för detta behöver bli avsevärd. Preece m.fl. (1994) beskriver olika metoder, bl.a. low-fidelity och high-fidelity prototyping. En low-fidelity prototyp ser inte ut som en slutgiltig produkt, vilket ofta är en fördel utan dessa prototyper är oftast gjorda med hjälp av enkla ritningar på papper. ISO 13407 beskriver inte explicit prototyping som en grundsten för användarcentrerad design (SIS, 1999). Gulliksen m.fl. (2002) menar trots detta att prototyping är en av de tolv nyckelprinciper som användarcentrerad design baseras på:

”Prototyping - tidigt och kontinuerligt skall prototyper användas för att visualisera och utvärdera idéer och designlösningar med slutanvändarna.” (Gulliksen m.fl., 2002, s.111)

Benyon m.fl (2005) skriver att prototyper kan användas i utvecklingsprocessen på olika sätt. Ett sätt är att prototyper fungerar som kravanimationer, vilket är ett begrepp som beskriver användandet av prototyper för att visualisera och verifiera krav på det färdiga systemet. Martin och Eastman (1996) anser att med prototyper kan användarna tidigt i och under hela utvecklingsprocessen ge sina åsikter om produkten och på så sätt kan de ge förslag på förbättringar. Preece m.fl. (1994) menar att det är viktigt att man i början undviker dyra IT stöd för att göra datorbaserade prototyper utan istället börjar med s.k. low-fidelity prototyper på en låg nivå, t.ex. med papper och penna och på det sättet uppmuntrar kreativiteten istället för att begränsa den.

Syfte

Syftet med studien som presenteras i denna uppsats var att utvärdera användbarheten hos ett av ABB egenutvecklat och använt beräkningsprogram PC-OSKAR. PC-OSKAR används vid design och konstruktion av motorer.

Metod

Metoden som användes vid användbarhetstestet av PC-OSKAR var Performance Measurement Method (PMM). PMM utvecklades ursprungligen i ett EU-project som heter MUSIC (Measuring Usability of Systems in Context, ESPIRIT Project 5429-Metrics for Usability Standards in Computing). PMM underlättade väsentligen mätningen av prestationsrelaterade data och gav möjlighet att utvärdera PC-OSKAR användbarhet genom att observera och analysera hur framgångsrikt deltagarna i testet utförde olika arbetsuppgifter med PC-OSKAR. PMM gav i detta testet ett underlag för designförbättringar av PC-OSKAR, men PMM kan även användas för att jämföra användbarheten hos PC-OSKAR med konkurrerande beräkningsverktyg, och olika versioner av PC-OSKAR. (Rengger m.fl., 1995)

Undersökningsgrupp/deltagare

Utifrån kontextanalysen, vilket är en del av PMM, valdes fem stycken deltagare till användbarhetstestet. Kontextanalysen bestod i att olika faktorer i PC-OSKAR:s kontext (sammanhang) identifierades och beskrevs, t.ex. typ av användare, arbetsuppgifter, miljö, etc. Fördelen med att göra en systematisk kontextanalys i enlighet med PMM var flera, bl.a. minimerades risken att få med helt fel deltagare, realistiska testuppgifter kunde skapas och så mycket som möjligt av PC-OSKAR:s kontext kunde återskapas i testsituationen. Deltagarna som valdes ut var alla män i åldern 35-55 år, och med en medelålder på 42 år. Alla deltagare var högkvalificerade civilingenjörer på ABB. Samtliga hade en genuin erfarenhet av att designa motorer. Deltagarna hade en varierande tidigare erfarenhet av beräkningsprogrammet PC-OSKAR och Windows miljön. Några av deltagarna hade inte svenska som modersmål utan kommunicerade på engelska.

Material

För att kunna utvärdera PC-OSKAR:s användbarhet användes PMM. Utifrån PMM:s definitioner operationaliserades nyckelbegreppen i ISO 9241 kapitel 11, dvs. *Effectiveness*, *Efficiency* och *Satisfaction*. I PMM definierades *Effectiveness* dimensionen som i vilken utsträckning ett mål eller en uppgift var uppnådd. *Efficiency* dimensionen definierades som den grad av ansträngning som krävdes för att slutföra och uppnå målet eller uppgiften. *Satisfaction* dimensionen definierades som graden av tillfredställelse och positiva känslor som PC-OSKAR frambringade då PC-OSKAR användes.

Deltagarna fick sju stycken representativa uppgifter. Syftet med dessa var att med hjälp av uppgifterna utvärdera de tre dimensionerna ovan. Samtliga uppgifter var skrivna på engelska.

Engelska är ABB:s koncernspråk vilket alla tjänsteman förutsätts kunna. Uppgifterna var representativa i den bemärkelsen att uppgifterna speglade de dagliga uppgifterna med verktyget. Uppgifterna togs fram utifrån kontextanalysen av testledaren tillsammans med den ansvarige för PC-OSKAR.

I uppgift 1 fick deltagarna i uppgift att från en diskett installera en ny version av PC-OSKAR. I uppgift 2 fick deltagarna som uppgift att utifrån ett testexempel modifiera en design. I uppgift 3 skulle deltagarna med hjälp av ett testexempel få PC-OSKAR att skriva ut ett diagram samt beräkna starttid och temperaturhöjning vid en 20 % höjning av uteffekten. I uppgift 4 skulle deltagarna förklara olika alternativ i en dialogruta. I uppgift 5 fick deltagarna utifrån ett antal förutbestämda specifikationer designa en motor. I uppgift 6 fick deltagarna analysera totala förluster vid användning av en konverterare. I uppgift 7 fick deltagarna välja en motor från en diskett med flera färdiga exempel och utifrån dessa exempel finna nödvändiga tekniska data. Se bilaga 1 för mer detaljer kring uppgifterna.

För att kunna mäta *Effectiviness* och *Efficiency* dimensionerna användes ABB:s portabla användbarhetslaboratorium innehållande kameror, monitor, ljudupptagningsenheter och videobandspelare. Programmet Diagnostic Recorder for Usability Measurement (DRUM) vilket är en del av PMM användes på en Mac för loggning, administration och analys av prestationsrelaterad data. För att mäta *Satisfaction* dimensionen användes den engelska versionen av frågeformuläret Software Usability Measurement Inventory (SUMI). Deltagarna ombads i SUMI formuläret svara på 50 påståenden och välja mellan; *Agree*, *Undecided* och *Disagree*. Några exempel på påståenden från SUMI:

- I would recommend this software to my colleagues.
- It takes too long to learn the software commands.
- The way the information is presented is clear and understandable.

För att analysera de data som samlats in med SUMI användes analysprogrammet SUMISCO. Analysen i SUMISCO resulterade i en global skala och fem specifika subskalor på hur användbart PC-OSKAR var. De olika subskalorna var:

- *Global*: Hur användbart deltagarna fann att PC-OSKAR var. Skalan är mest användbar när det gäller att jämföra t.ex. olika versioner, eller system.
- *Efficiency*: Deltagarnas känsla av att de kunde utföra sina uppgifter snabbt, effektivt och ekonomiskt, eller motsatsen att de fann PC-OSKAR som ett hinder.

- *Affect*: En psykologisk term för emotioner. I denna kontext menas ifall deltagarna kände sig bra, varma, glada, eller motsatsen vid användningen av PC-OSKAR.
- *Helpfulness*: Deltagarnas upplevelse av PC-OSKAR:s kommunikation ifråga om hur hjälpsamt PC-OSKAR upplevdes och hur väl PC-OSKAR assisterat vid lösningen av problem. Generellt går det att säga att *Helpfulness* refererade till de meddelanden och varningar som getts av PC-OSKAR.
- *Control*: Deltagarnas känsla av att PC-OSKAR svarat på ett normalt och konsistent sätt på indata och kommandon. Det skulle inte vara svårt att få PC-OSKAR att fungera och PC-OSKAR:s sätt att fungera skulle vara lätt att förstå för deltagarna.
- *Learnability*: Deltagarnas känsla av hur lätt det är att lära sig och bli familjär med PC-OSKAR inkluderande hjälpfunktioner, träning, handböcker, osv.

För att mäta *Satisfaction* dimensionen intervjuade testledaren efter varje test som ett komplement till SUMI deltagarna, se bilaga 2. Dessutom observerades deltagarna kontinuerligt under tiden de utförde uppgifterna. Observationerna noterades på ett separat papper för varje deltagare och uppgift.

Procedur

Varje deltagare gavs sju olika uppgifter, representativa uppgifter som deltagarna konfronterades med i sitt dagliga arbete. Testet varade cirka fyra timmar per deltagare och inkluderade i kronologisk ordning följande aktiviteter:

- Introduktion (cirka 10 minuter)
- Testuppgifter och frågor på uppgifterna
- SUMI frågeformulär (cirka 15 minuter)
- Intervju (cirka 20 minuter)

Det var totalt fem sessioner vilka inkluderade en deltagare, en assistent (den ansvarige för PC-OSKAR), och en testledare som också agerade observatör. Assistenten var helt tyst och ryckte enbart in när deltagarna fastnat totalt vid någon uppgift. Deltagarna hade inte några hjälpmedel och kunde inte se testledaren. Deltagarna blev innan testet utfrågade om sin bakgrund och tidigare erfarenheter. Deltagarna blev sedan informerade om förutsättningarna för testet, att det inte var dem som testades utan PC-OSKAR, och att testet spelas in på band för senare analys. Vidare blev dem informerade om att det inspelade materialet bara var tillgängligt för testledaren, att allt material inkluderande det

inspelade materialet skulle lämnas över till den ansvarige för PC-OSKAR efter projektets avslut, och att dem hade rätt att avbryta sitt deltagande i testet när dem så önskade.

Det portabla labbet var uppställt i ett avskilt rum. Testledaren var i ett separat rum men kunde se deltagarna via ett fönster samt på flera monitorer. Varje testsession spelades in på ett VHS band. Samtidigt som skärmbilden spelades in tillsammans med klockslag var en kamera riktad mot deltagarna. Dessa bilder mixades till en bild tillsammans med ljudet från testerna. Deltagarna uppmuntrades att tänka högt när dem försökte lösa uppgifterna. Deltagarnas aktiviteter loggades av testledaren med DRUM för senare analys. Samtidigt gjorde testledaren skriftliga noteringar om vilka problem som observerades och uppfattades när deltagarna tänkte högt. Observationerna gjordes utan någon speciell kodningsmall. När deltagarna var färdiga med sina uppgifter intervjuades deltagarna av testledaren. Deltagarna fick även fylla i SUMI formuläret. Deltagarna tackades sedan och fick med sig en liten gåva.

Dataanalys

För att analysera de prestationsrelaterade data som låg till grund för beräkningen av *Effectiveness* och *Efficiency* dimensionerna användes DRUM, och resultatet visualiserades med Excel. För att beräkna *Effectiveness* dimensionen mättes om deltagarna utan assistans kunde slutföra de uppgifter som dem fått, och om resultatet var korrekt. De två olika feltyperna *Obstruction Errors* (deltagarna kunde inte slutföra sin uppgift) och *Completion Errors* (deltagarna kunde inte utföra en korrekt uppgift, eller det tog längre än 10 minuter att utföra en uppgift) sammanvägdes för att beräkna *Effectiveness* i procent. En fullständigt misslyckad uppgift gav 0 %, medan en perfekt gav 100 %. Utförda uppgifter men inte med helt korrekt resultat innebar ett procenttal någonstans mellan 0 – 100 %. Det exakta schemat för vilket procenttal som gavs berodde på hur allvarligt misstaget bedömdes vara. För att beräkna *Efficiency* dimensionen mättes den tid det tog för deltagarna att slutföra en uppgift. Utan *Effectiveness* fanns det inte någon *Efficiency*. *Efficiency* var inte acceptabelt ifall deltagarna inte klarade av uppgifterna utan hjälp. *Efficiency* definierades som skillnaden mellan *Effectiveness* och tiden det tog att utföra uppgiften, dvs. $Efficiency = Effectiveness / Tid\ spenderad\ på\ uppgiften$ (procent/minuter). Som komplement sammanfattades observationerna för varje uppgift. För att mäta *Satisfaction* dimensionen användes frågeformuläret SUMI samt intervjuer av deltagarna som sedan sammanfattades. Data hämtade från SUMI formulären konverterades av SUMISCO till lämplig subskala och jämfördes sedan med innehållet i en databas med data från tidigare gjorda användbarhetstester (sammanlagt 1000 svar från 120 testade system).

Resultatet från SUMISCO var standardiserat med medelvärdet 50 och standardavvikelsen 10. För varje subskala beräknade SUMISCO medianen och ett 95 % konfidensintervall.

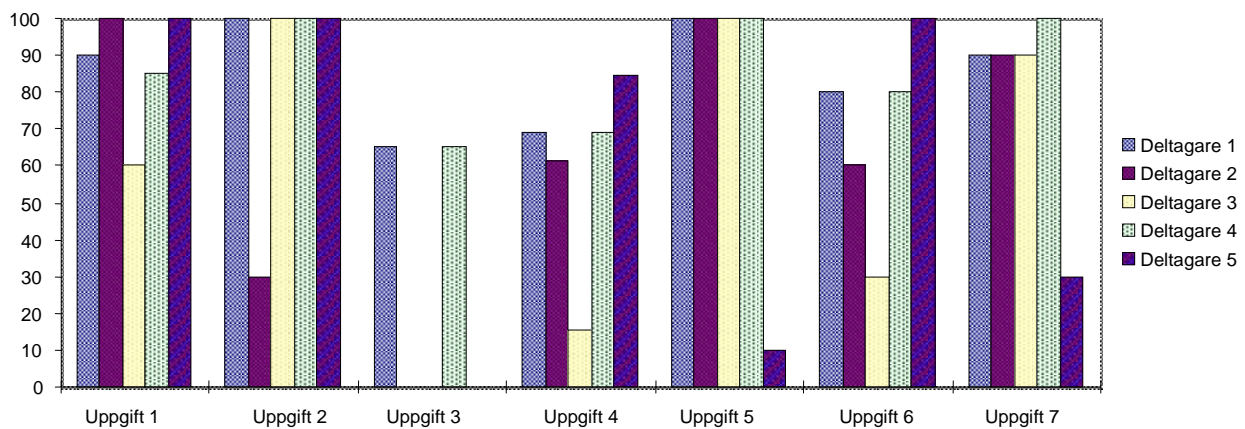
Resultat

Effectiveness

Vid flera tillfällen lyckades deltagarna inte:

- slutföra sin uppgift, t.ex. när deltagarna fastnade vid en uppgift, och var tvungna att få instruktioner om hur de skulle gå vidare, eller explicit hjälp för att kunna komma vidare (*Obstruction Errors*).
- utföra en korrekt uppgift, t.ex. när deltagarna inte lyckades göra en speciell uppgift, eller gjorde en felaktig slutsats när denne tog del av information från programmet, eller deltagarna tog för lång tid på sig på någon del av uppgiften. Längre än 10 minuter räknades som ett misslyckande (*Completion Errors*).

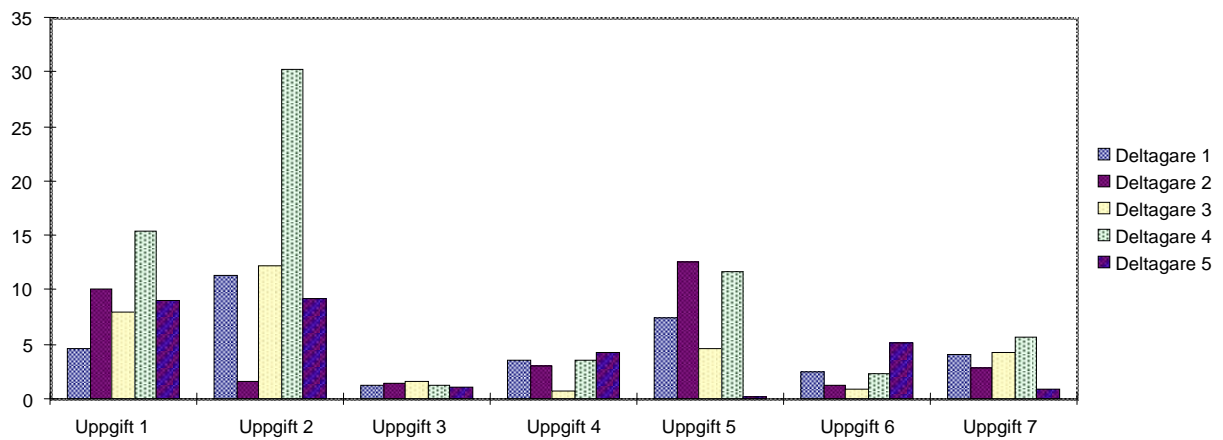
De två olika feltyperna *Obstruction Errors* och *Completion Errors* sammanvägdes för att beräkna *Effectiveness* i procent. En fullständigt misslyckad uppgift gav 0 %, medan en perfekt gav 100 %. Utförda uppgifter men inte med helt korrekt resultat innebar ett procenttal någonstans mellan 0 – 100 %. Det exakta schemat för vilket procenttal som gavs berodde på hur allvarligt misstaget bedömdes vara. Diagrammet i figur 4 ger en överblick över varje uppgifts beräknade *Effectiveness*. Generellt är det ett väldigt bra resultat. Med några få undantag är poängen hög. Uppgifterna genomfördes fullständigt och korrekt i mer än 60 % av fallen, och vid tolv tillfällen (av 35) är resultatet helt korrekt.



Figur 4. Beräknad *Effectiveness* för respektive deltagare och uppgift.

Efficiency

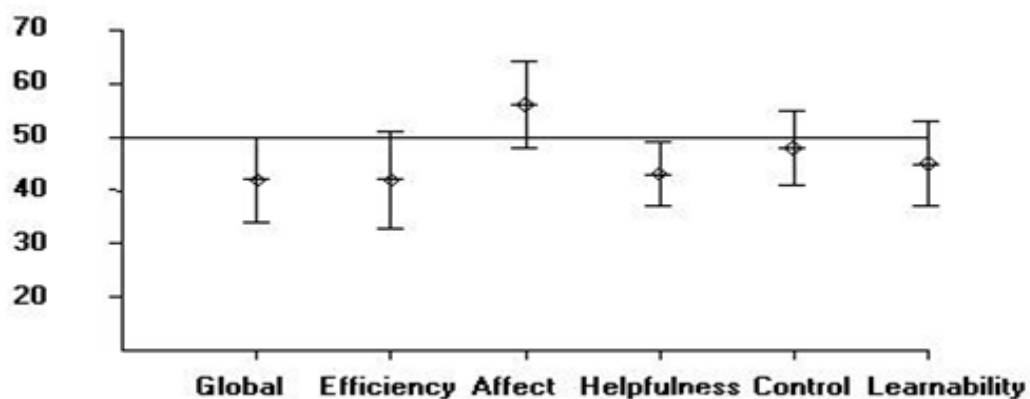
Diagrammet i figur 5 ger en överblick över varje uppgifts beräknade *Efficiency*. *Efficiency* definierades som skillnaden mellan *Effectiveness* och tiden det tog att utföra uppgiften, dvs. $Efficiency = Effectiveness / Tid\ spenderad\ på\ uppgiften$ (procent/minuter). *Efficiency* var inte acceptabelt ifall deltagarna inte klarade av uppgifterna utan hjälp. För deltagarna gällde det att för respektive uppgift få ett så högt beräknat *Efficiency* som möjligt. *Efficiency* dimensionen är inte särskilt användbar utan att ha något att jämföra med, t.ex. går det att använda *Efficiency* dimensionen för att jämföra två versioner av PC-OSKAR, när PC-OSKAR används av samma grupp, för samma uppgifter, och i samma kontext. I detta test gick det i viss mån att jämföra *Efficiency* dimensionen mellan två eller flera uppgifter, eller för att jämföra med framtida versioner av PC-OSKAR. Väljer man att jämföra uppgifterna i testet med varandra är det viktigt att notera att uppgifterna i testet skiljde sig en hel del åt när det gällde komplexitet. Resultatet av en jämförelse mellan uppgifterna kan då bli missvisande. Beräknad *Efficiency* kunde generellt varit bättre. Det var bara deltagare 4 som i uppgift 2 fick ett mer acceptabelt resultat. Beräknad *Efficiency* återspeglar de problem deltagarna hade med att få rätt på alla PC-OSKAR:s parametrar. Ungefär samma tid som det tog att få rätt på parametrarna ägnade deltagarna åt att studera sidorna som innehöll resultat från beräkningar odyl. I båda fallen handlade problemet om att finna nyckelinformation som förtydligade och förklarade både användningen av PC-OSKAR:s många parametrar såväl som vad det egentligen var för resultat som presenterades.



Figur 5. Beräknad *Efficiency* för respektive deltagare och uppgift.

Satisfaction

Med SUMI mättes deltagarnas attityd till PC-OSKAR vid utförandet av uppgifterna. Diagrammet i figur 6 visar att poängen för *Helpfulness* är under medelvärdet. Inget av de övriga värdena var lika tydligt vare sig över eller under medelvärdet, *Global*, *Efficiency*, *Control* och *Learnability* fick låga poäng från deltagarna medan *Affect* fick relativt höga poäng, dvs. över medelvärdet 50.



Figur 6. Diagrammet visar median och övre och undre 95 % konfidensintervall.

Intervjuer

Sammanfattningsvis visade intervjuerna att deltagarna fann PC-OSKAR vara ett stort framsteg jämfört med andra system. Flera deltagare tyckte att man med PC-OSKAR snabbt kunde göra förändringar på sin design och sedan göra en ny beräkning. Beräkningarna som PC-OSKAR gjorde ansågs vara korrekta och precisa. Flera av deltagarna efterlyste någon form av ”kokbok” som täckte de vanligaste uppgifterna, vilka beräkningar som gjordes, vilka parametrar som användes, och hur deltagarna skulle använda dessa parametrar. Deltagarna fann det generellt svårt att finna information, dels i hjälpfunktionen men också i PC-OSKAR:s gränssnitt då layouten uppfattades som otydlig och komplex, att det fanns för många parametrar, för dålig feedback, användarmanualen var för dålig och att den information som ändå tillhandahålls av PC-OSKAR, t.ex. via hjälpfunktionen inte var relevant eller för komplex. En deltagare påpekade att tyska användare kunde få problem med att PC-OSKAR enbart finns på engelska, och att det ofta är en engelska med många förkortningar.

Observationer

Uppgift 1:

Uppgift 1 bestod i att deltagarna fick i uppgift att från en diskett installera en ny version av PC-OSKAR. Om inte installationen av PC-OSKAR gjordes på rätt sätt gick det inte heller att lita på PC-OSKAR:s beräkningar. Flera av deltagarna fick problem med Dos-editorn. Anledningen till detta var att deltagarna hade begränsad kunskap om Autoexec.bat. Deltagarna förstod i regel inte vad som hände. Nästan alla deltagarna verkade osäkra på om de hade lyckats. En av deltagarna efterlyste någon form av bekräftelse från PC-OSKAR på att denne hade lyckats installera PC-OSKAR.

Uppgift 2:

Uppgift 2 bestod i att deltagarna utifrån ett testexempel skulle modifiera en design. Den här uppgiften kunde verka enkel och kunde också vara det ifall deltagarna valde att göra det enkelt. Antingen hade deltagarna erfarenhet av att använda PC-OSKAR, och då visste de exakt vilka parametrar som behövde sättas, eller så var deltagarna noviser på PC-OSKAR och insåg då inte komplexiteten hos PC-OSKAR. Flera av deltagarna insåg PC-OSKAR:s komplexitet, förundrade sig och fortsatte att söka efter information som kunde vara till hjälp.

Uppgift 3:

Uppgift 3 bestod i att deltagarna med ett testexempel skulle få PC-OSKAR att skriva ut ett diagram samt beräkna starttid och temperaturhöjning vid en 20 % höjning av uteffekten. Uppgiften

visade sig vara den svåraste för deltagarna. Flera av deltagarna fick svårigheter med att göra ett diagram. Flera av deltagarna hade inte använt PC-OSKAR för att göra ett diagram och förstod inte vikten av parametern "L PLOT". För att överhuvudtaget kunna göra några diagram var deltagarna tvungna att ge denna parameter ett värde. Deltagarna sökte länge efter information. Det fanns flera parametrar som ställde till problem för deltagarna, t.ex. STATUS. Det fanns information i hjälpfunktionen men denna uppfattades av deltagarna som ologisk och otydlig.

Uppgift 4:

Uppgift 4 bestod i att deltagarna fick förklara olika alternativ i en dialogruta. Nästan ingen av deltagarna hade tidigare använt den aktuella dialogen. Deltagarna fann inte den särskilt användbar efter att de lärt sig mer om den. Huvudproblemet var att de termer som användes var för svåra och att layouten inte var särskilt intuitiv

Uppgift 5:

Uppgift 5 bestod i att deltagarna utifrån ett antal förutbestämda specifikationer skulle designa en motor. För en av deltagarna var det här helt nytt. Deltagaren tittade flera gånger i användarmanualen i hopp om att finna någon information som kunde vara till hjälp men blev besviken de flesta gångerna. Deltagaren hade väldigt svårt även vid sökning i hjälpen, det fanns ingen bra funktion för att söka på nyckelord.

Uppgift 6:

Uppgift 6 bestod i att deltagarna fick analysera totala förluster vid användning av en konverterare. I denna uppgift blev PC-OSKAR:s komplexitet uppenbar när en av deltagarna fick problem. Deltagaren hade inte erfarenhet sedan tidigare av denna typ av uppgift och var tvungen att använda sig av "trial and error". Hjälpfunktionen gav ingen som helst hjälp. Ett exempel som deltagarna fann när dem sökte i hjälpfunktionen om parametern LDUMP: *"Adds more details in output file "OSKAR.013" from inverter analysis. Only if LSPEKT is equal to 1!!!"*

Det gavs ingen information om vilka värden som förväntades ges till parametern LDUMP. Deltagaren som tog längst tid på sig för uppgiften fick flera felmeddelanden från PC-OSKAR som visade på att värdet på parametern LDUMP och flera till den relaterade som LSPEKT var fel. När deltagaren försökte ändra värdena så försökte deltagarna ha både loggfilen där felmeddelandena fanns öppna på skärmen samtidigt med dialogrutan där parametrarna fanns. Detta fungerade inte.

Deltagaren var säker på att denne behövde ändra på någon parameter men var inte säker på vad som skulle ändras. Deltagaren ändrade parametern BRST från 1 till 0. PC-OSKAR kraschade efter att ha följande meddelanden: ”*Winoskar. An error has occurred in your application. If you choose Ignore, you should save your work in a new file. If you choose Close, your application will terminate.*” Deltagaren valde *Ignore* varefter följde nästa meddelande: “*Application error. Winoskar caused a General protection Fault in module OSKAR.DLL at 0053:01:38*”. Det enda alternativet var att stänga ner meddelandet. När deltagaren gjorde detta stängde också PC-OSKAR ner, dvs. kraschade med dataförlust som följde.

Uppgift 7:

Uppgiften 7 bestod i att deltagarna fick välja en motor från en diskett med flera färdiga exempel och utifrån dessa exempel finna nödvändiga tekniska data. Vid flera tillfällen när deltagarna försökte använda en fil fick dem meddelandet: ”*Parameter not found !*” *NFOT = 4*”. Enda alternativet var att svara *Ok* och då fick deltagarna meddelandet: ”*Parameter was not found ? Do you wish to continue reading file, Yes or No*”. Meddelandena var inte bara kryptiska utan till synes redundanta. Det var intressant att notera på hur många olika sätt deltagarna löste uppgiften. Flera av deltagarna uttryckte också sin förvåning att PC-OSKAR kunde mer än vad de förväntat sig.

Diskussion

Syftet med denna studie var att testa användbarheten hos beräkningsverktyget PC-OSKAR. PC-OSKAR är ett av ABB egenutvecklat beräkningsprogram vilket används av ingenjörerna på ABB vid design och konstruktion av motorer. Metodiken som användes vid testet var Performance Measurement Method (PMM), och en operationalisering av nyckelbegreppen i standarden ISO 9241 kapitel 11, dvs. *Effectiveness*, *Efficiency* och *Satisfaction* (Rengger m.fl., 1995; SIS, 1998). Fem stycken högkvalificerade civilingenjörer valdes utifrån en kontextanalys, vilket ingick i PMM, till att delta i testet. Deltagarna hade i regel en väldigt lång erfarenhet av att konstruera motorer. Deltagarna hade en mer varierad erfarenhet av PC-OSKAR och Windows-miljön.

PC-OSKAR uppvisade ett väldigt bra resultat när det gällde beräknad *Effectiveness*, se figur 4. Med några få undantag är poängen hög. Uppgifterna genomfördes fullständigt och korrekt i mer än 60 % av fallen, och vid tolv tillfällen (av 35) är resultatet helt korrekt. Beräknad *Efficiency* kunde varit bättre men återspeglar att deltagarna fick ägna mycket tid åt de olika parametrarna, att förstå dem, att använda rätt parameter, och ge dem rätt värde, se figur 5. PC-OSKAR var inte speciellt transparent, dvs. de grunder som programmet använde för sina beräkningar, parametrar osv. var inte

tydliga för deltagarna. För att göra det ännu svårare var deltagarna för vissa parametrar tvungna att ange en kod som instruerade PC-OSKAR om hur PC-OSKAR skulle hantera parametrar som deltagarna till fullo kanske inte förstått. Deltagarna hade även problem med frånvaron av hjälp eller en väldigt otydlig hjälp när det gällde parametrarna, t.ex. kunde deltagarna läsa följande i hjälpen för parametern NP12P:

”NP12P Unisolated strand diameter for stator conductor group 2(mm). If DP11P is less than or equal to 0.1 OSKAR will disregard of DP12P and select wires according to DIMTAB”

De beräknade dimensionerna *Effectiviness*, *Efficiency* och *Satisfaction* skulle med största sannolikhet kunnat vara bättre om PC-OSKAR varit enklare, tydligare och mer strukturerat, och då inte bara när det gällde hjälpfunktioner, utan generellt när det gällde layout och interaktion (Mandel, 1997; Martin m.fl., 1996; Nielsen, 1993). Det var mer ett undantag än en regel att PC-OSKAR följde etablerad Windows standard, t.ex. vid möjligheten att spara då det fanns en uppsjö av olika möjligheter att i olika lägen spara inställningar odyl. Dem flesta deltagarna hade någon form av erfarenhet från Windows. Om PC-OSKAR följt Windows standard bättre skulle nog inlärningströskeln varit mindre (Nielsen, 1993).

Vid utveckling av framtida PC-OSKAR versioner bör fokus flyttas från att inkludera ännu fler funktioner, med ännu flera parametrar att ta hänsyn till för användarna, till användarna och deras uppgifter och kontext, t.ex. måste en grundläggande uppgift som installationen av PC-OSKAR bli mer effektiv. Att installera PC-OSKAR är det första en användare gör med PC-OSKAR, och det första intrycket de får av PC-OSKAR. Den attityd som användarna får till PC-OSKAR vid denna tidpunkt är väldigt svår att förändra (Nielsen, 1993). Ett sätt att lösa detta problem är att automatisera hela proceduren, att göra ett installationsprogram som installerar PC-OSKAR automatiskt. Ifall detta inte är möjligt måste instruktionerna bearbetas från grunden och kompletteras med vilken feedback som man som användare kan förvänta sig från systemet (Nielsen, 1993).

Det framgick relativt tydligt att PC-OSKAR ifråga om användarnas attityd, dvs. *Satisfaction*, till PC-OSKAR lämnade en del att önska, se figur 6. Den enda accepterbara subskalan var *Affect*, vilket utifrån intervjuerna att döma nog kan förklaras med att deltagarna finner PC-OSKAR snabbt att arbeta med, och att PC-OSKAR gav korrekta och specifika resultat. Det är värt att notera att bl.a. observationerna av deltagarna visade att flera av deltagarna hade svårt för att installera PC-OSKAR. Ifall inte installationen blev korrekt kunde man inte heller lita på PC-OSKAR:s beräkningar. Attityden hos dem ansvariga för PC-OSKAR är inget som kan utläsas av resultatet från SUMI, men också ett problem som inte bör förbises, och som kanske kan kategoriseras som kulturellt. PC-

OSKAR är nämligen designat och utvecklat av ingenjörer för ingenjörer, och som ingenjör anses det ofta att man skall klara av det mesta. Dessutom anses ofta användbarhet inte vara lika viktigt som häftiga funktioner som gör det dem skall snabbt och korrekt (Preece m.fl., 1994).

Med fler och fler användare över hela världen, frekventa nya releaser av PC-OSKAR, och snabbare produktcykler när det gäller nya motorer, blir ett mer användbart PC-OSKAR för varje ny version än viktigare. Kostnaden för att inte tidigt i utvecklingsprocessen inkludera användbarhet och se användbarhet som en del av den totala systemkvaliteten kommer bara bli dyrare (Bias m.fl., 1994; Nielsen, 1993; Sparre m.fl., 1994). Dem ansvariga för PC-OSKAR måste för att undvika kostnadskrävande förändringar av PC-OSKAR i slutet av utvecklingsprocessen, eller behöva skapa en omfattande och dyr support, i förväg försöka identifiera de krav som olika användare och kontexter kommer ställa på PC-OSKAR (Rexfelt m.fl., 2006; SIS, 1999; Gulliksen m.fl. 2002; Ottersten m.fl., 2002) . De flesta experterna, t.ex. Gulliksen m.fl. (2002), menar att företag ofta är mer fokuserade på teknik och nya funktioner än på användbarhet. Användbarhet är något man kan tänka på i slutet av utvecklingsprocessen, om man hinner och har pengar över. Detta är ett faktum även när det gäller ABB och PC-OSKAR. De ansvariga för PC-OSKAR måste för att kunna uppfylla framtida användares krav, och för att kunna konkurrera, att i enlighet med ISO 13407 börja använda sig av ett användarcentrerat perspektiv i utvecklingsprocessen (Gulliksen m.fl., 2002; Kontogiannis m.fl.1997; SIS, 1999).

Testet av PC-OSKAR hade en kvantitativ ansats genom användningen av PMM. Fördelen med denna ansats var att användbarheten hos PC-OSKAR gjordes mätbar (Rengger m.fl., 1995). Samtidigt var den kvantitativa ansatsen begränsande då den bl.a. utgick från givna svarsalternativ, se t.ex. svarsalternativen i SUMI. Den kvantitativa ansatsen kompletterades därför med kvalitativa metoder som en intervju, såväl som att deltagarna observerades under utförandet av de sju uppgifterna. De kvalitativa metoderna var inte heller de utan begränsningar. Holme och Solvang (1997) skriver att några av de kvalitativa metodernas största brister är bristen på mätbarhet, reliabilitet, och att det är svårare att generalisera resultatet. Att i testet av PC-OSKAR komplettera den kvantitativa ansatsen som PMM innebar med kvalitativa metoder hade som fördel att testet av PC-OSKAR blev mer öppet för subjektiva tolkningar, t.ex. av hur deltagarna uppfattade PC-OSKAR, vilket i sin tur medförde ny förståelse såväl som kunskap om hur PC-OSKAR kan förbättras.

Eftersom det bara var fem stycken deltagare i testet var det svårt att generalisera resultatet från studien, t.ex. SUMI till att gälla alla möjliga användare av PC-OSKAR, se figur 6. Dessutom fanns

det inga kvinnor med i testet. Anledningen till detta var att män generellt är överrepresenterade inom ingenjörsvetenskapen. Det är viktigt att ha i åtanke att det inte är brukligt att ställa samma krav på ett användbarhetstest, fast användbarhetstestet till det yttre liknar ett klassiskt experiment, som på ett klassiskt hypotesprövande experiment, t.ex. ifråga om att kunna generalisera resultatet (Nielsen, 1993). I utvärderingen av PC-OSKAR blir bristen i att inte kunna generalisera resultatet till en annan organisation väldigt tydlig. Men det är viktigt att ha i åtanke att PC-OSKAR inte är tillgängligt på den öppna marknaden, och att det än så länge bara finns ett fåtal användare av PC-OSKAR. Resultatet från testet av PC-OSKAR kan med fördel användas för att kunna bilda sig en uppfattning om potentiella problem etc. när PC-OSKAR börjar användas av fler ingenjörer, på andra avdelningar, och kanske t.o.m. så småningom i andra organisationer.

Testet tog cirka fyra timmar per deltagare och den långa tiden kan onekligen ha påverkat hur alerta deltagarna var och detta i sin tur prestationen vid utförandet av olika uppgifterna. Detta upplevdes vid observation av deltagarna inte vara ett större problem. Närheten mellan testledare, den ansvarige för PC-OSKAR, och deltagarna, kunde även orsakat problem då alla involverade arbetade på samma företag. Deltagarna skulle t.ex. kunnat börja agera som de trodde förväntades av dem vid utförandet av uppgifterna. Ingetdera blev uppenbart vare sig i testsituationen, vid intervjuerna, eller vid analysen av mätdata. Men det är sannolikt att anta att denna påverkan trots allt fanns i mindre eller högre grad. Både testledare som den ansvarige för PC-OSKAR, som fungerade som assistent vid testerna, var måna om att försöka skapa en avslappnad atmosfär där deltagarna kände att de kunde vara naturliga och öppna sig genom att inta en lyssnande inställning gentemot deltagarna, att förhålla sig objektiva, och inte på något sätt vara värderande vare sig när det gällde uttalanden eller attityd. Det är svårt att göra ett helt perfekt test. Testet kanske hade fel fokus, fel dimensioner, fel uppgifter, osv. Men genom att göra en noggrann kontextanalys, och använda en beprövad metodik som systematiskt adresserar problemet med användbarhet, dvs. PMM, har testet försökt minimera chanserna för eventuella fel.

Framtida forskning och utveckling rekommenderas att fokusera på att försöka göra PC-OSKAR mer transparent. Alla parametrars funktion och användning bör göras tydligare genom en tydlig och välstrukturerad hjälp, kanske kompletterat med någon form av "kokbok" som beskriver parametrarna och hur de används. Alternativt bör fokus ligga på att göra parametrarna "osynliga" för användarna. Användarna skall inte behöva bry sig, utan istället bör det undersökas vad som går att automatisera i PC-OSKAR, t.ex. vid installation, hanteringen av parametrar, osv. Generellt bör den allmänna layouten, såväl som interaktionen (dialogboxar osv.) mellan PC-OSKAR och användare förbättras, och det bör därför undersökas på vilket sätt och hur. Framtida användare kan se

annorlunda ut och även ha andra krav än dagens. Det allra viktigaste för PC-OSKAR:s framtid är att de ansvariga försöker få till stånd en användarcentrerad utveckling, t.ex. med stöd av ISO 13407 (SIS, 1999). Slutligen bör de ansvariga för kommande versioner av PC-OSKAR tidigt i utvecklingsprocessen definiera klara och välformulerade användbarhetsmål för PC-OSKAR (Rexfelt m.fl., 2006).

Bilaga 1. Uppgifter

Task 1 Installation

- You will use a new PC-OSKAR release during the test.
- Insert the PC-OSKAR installation disk.
- Enter A:\SETUP A C (no colons!)
A is the floppy drive carrying the disk and C is the hard disk drive where PC-OSKAR will be installed.
- The installation procedure will create the following directories:
 - \OSKAR\BIN
 - \OSKAR\EXAMPLES
 - \OSKAR\TEMP
- After installing, make sure your path is set correctly and that the locking device is connected:
 - Modify the PATH command in \AUTOEXEC.BAT. Do the following:
 - C:\> EDIT AUTOEXEC.BAT
Add C:\OSKAR\BIN to the PATH (if the C-drive is used)
 - Execute the startup file \AUTOEXEC.BAT
 - Check if the Computer date is correct. (OSKAR may not run otherwise.) Do the following: C:\> DATE
 - Check if the locking device is connected to the parallel port.
 - Start Windows and add the PC-OSKAR icon to Windows. Do the following steps:
 - C:\>WIN
 - Go to the Program Manager window.
 - Select FILE, NEW from the menu.
 - Select Program Group, click OK.
 - Type as Group File: C:\OSKAR\BIMOSKAR.GRP

Task 2 Modification of winding

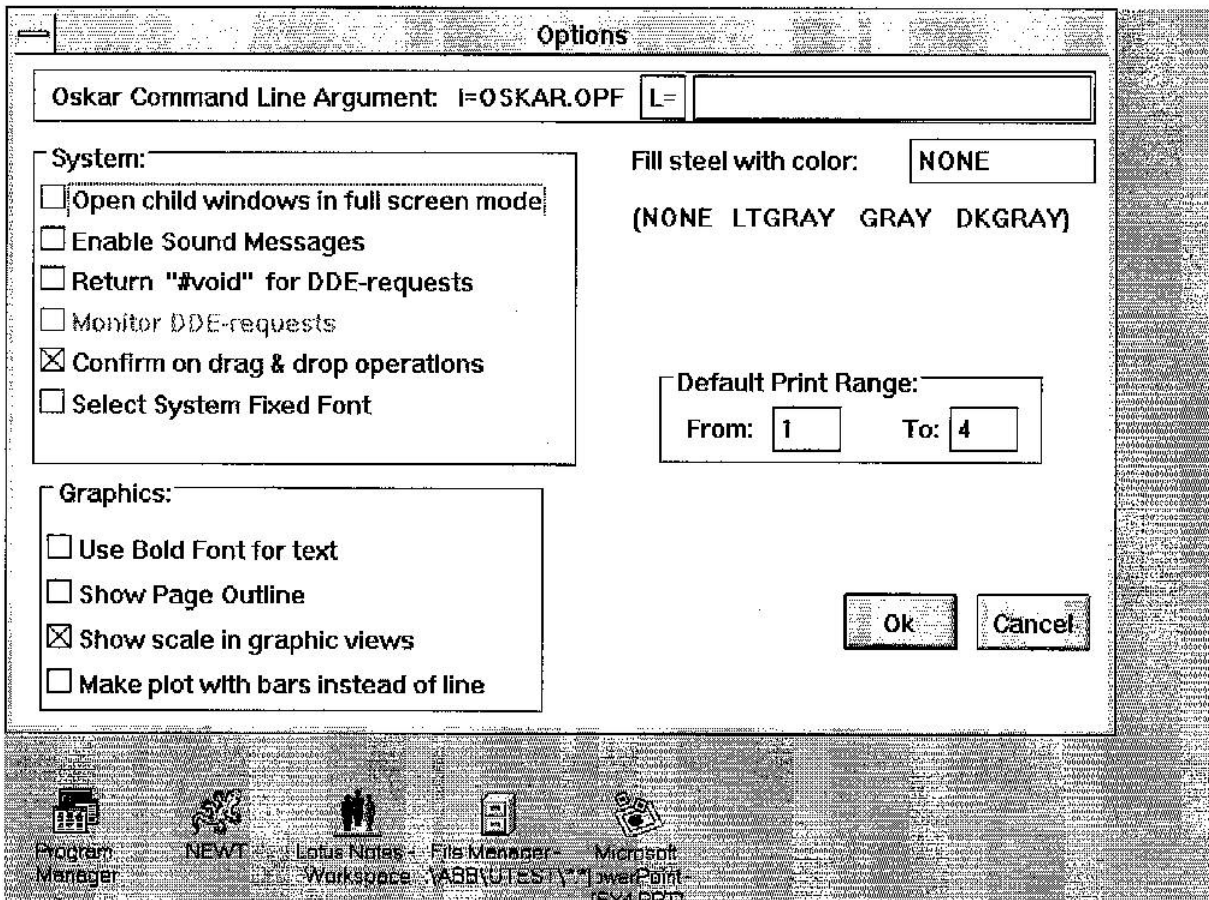
- Use the PC-OSKAR example E32.OPF.
- Modify the winding for 5000 V.
- Document your design (on paper).
- Print the view of all scalar parameters (in alphabetic order).

Task 3 Starting time and temperature rise calculation

- Use the 355_2 motor from the examples diskette.
- Plot the torque-speed curve (in D-connection).
- Calculate the starting time and temperature rise in starting for a cold machine. Set $J_{load} = 5 * J_{motor}$, and use a square bad torque curve.
- Document your results (on paper), and circle the answers.
- Print the view of all scalar parameters (in alphabetic order).
- Calculate the stationary temperature rise in the stator winding with 20% increased output power.
- Document your results (on paper), and circle the answer.
- Print the view of all scalar parameters (in alphabetic order).

Task 4 Program options

- Describe the various options in the “Set program options...” dialog.
(Try to answer without using the User’s Manual.)



Task 5 Design according to technical requirements

- Use the 355_6 motor from the examples diskette and design a motor with the following requirements:
 - Shaft height 355 mm
(in M2000 dimensions this means $DY1 = DY = DY2 = 600$ mm, $D = 420$ mm, $DAXEL = 130$ mm).
 - 6poles
 - Output power $P2 = 250$ kW
 - Stator voltage 400 V
 - Starting torque $Ts/Tn > 2.5$
 - Starting current $Is/In < 9.0$
 - Maximum torque $Tmax/Tn > 3.0$

- Document your design (on paper), and circle the requirements listed above.

- Print the view of all scalar parameters (in alphabetic order).

Task 6 Inverter analysis

- Use the 355_4 motor on the examples diskette and make an analysis of total losses on the following industrial type converter:
 - Subharmonic PWM method
 - Fundamental frequency for inverter analysis, 67 Hz
 - Reference value for electrical torque, 1910 Nm
 - Inverter switch frequency, 700 Hz

- Document your results (on paper), and circle the answer.

- Print the view of all scalar parameters (in alphabetic order).

Task 7 Quotation

- Find technical data for a quotation, i e
 - motor size
 - winding - could be special if advantageous
 - rated current
 - p.f
 - efficiency
 - full load starting current
 - Y/D starting current
- 100 motors/year, 52 kW, 2-pole, 400 V +1- 10%, Cl B temp-rise, direct start and Y/D-start possible, starting current preferably < 845%.

Resistive torque during start:

Speed 0 300 600 900 1200 1500 1800 2100 2400 2700 3000 rpm

Torque 0.42 0.37 0.35 0.36 0.37 0.39 0.43 0.47 0.53 0.63 0.74 p.u.

Load inertia 0.11 kgm²

- Document your design (on paper), and circle the requirements listed above.
- Print the view of all scalar parameters (in alphabetic order).

Bilaga 2. Intervjuguide

Your immediate reaction to

- the test

- the tool

What do you like the most?

What do you dislike the most?

How would you compare this environment to other tools you know?

Can you think of any function that is lacking, which would speed up/simplify work?

Will this be an efficient tool for those who will use it?

What was it like to work with English text in menus and titles?

Did you find the Help facility? Did you find the information useful?

Where did you have most difficulties using the tool?

Which is the most urgent thing to change in or add to the system?

Were the tasks realistic?

Any further comments on

- the tool

- the test

Referenser

- Abran, A., Khelifi, A., Suryan, W., & Seffah, A. (2003). Usability meanings and interpretations in ISO Standards, *Software Quality Journal*, 11, 325-338.
- Bailey, W.,R. (1989). *Human performance engineering: Using human factors/ergonomics to achieve computer system usability*. Englewood Cliffs, N.J.: Prentice Hall.
- Benyon, D., Turner, P., & Turner, S. (2005). *Designing interactive systems: people, activities, context, technologies*. Essex: Pearson Education.
- Bias, R.H. & D.J. Mayhew (1994). *Cost-Justifying Usability*. London: Academic Press.
- Earthy, J., Jones, B. S. & Bevan, N. (2001). The improvement of human-centered process - facing the challenge and reaping the benefit of ISO 13407, *International Journal of Human-Computer Studies*, 55, 553-585.
- Ehrlich, K., Butler, M.B. & Pernice, K. (1994). Getting the whole team into usability testing, *IEE Software*, 89-93.
- Gulliksen, J. & Göransson, B. (2002). *Användarcentrerad design*. Lund: Studentlitteratur.
- Gould, J.D. & Lewis, C. (1985). Designing for usability – key principles and what designers think of Them. *Communications of the ACM*, 28(3), 300-311.
- Holme, I.M., & Solvang, B.K. (1997). *Forskningsmetodik: Om kvalitativa och kvantitativa metoder*. Studentlitteratur: Lund.
- Kontogiannis, T. & Embrey, D. (1997). A user-centred design approach for introducing computer based process information systems, *Applied Ergonomics*, 28, 109-119.
- Mandel, T. (1997). *The elements of user interface design*. New York: John Wiley & Sons, Inc.
- Martin, A. & Eastman, D. (1996). *The user interface design book*. Chichester: John & Sons Ltd.
- Nielsen, J. (1993). *Usability Engineering*. London: Academic Press.
- Ottersten, I. & Berndtsson, J. (2002). *Användbarhet i praktiken*. Lund: Studentlitteratur.
- Preece, J., D, Sharp, H., Denyoun, D., Holland, S., Carey, T. (1994). *Human-Computer Interaction*. Wokingham: Addison-Wesley.
- Rengger, R, Macleod, M., Bowden, R, Drynan, A., & Blayney, M. (1995). *Performance Measurement Handbook Version 3*. Middlesex: NPL Usability Services.
- Rexfelt, O. & Rosenblad, E. (2006). The progress of user requirements through a software development project, *International Journal of Industrial Ergonomics*, 36, 73-81.
- Schneiderman, B. (1998). *Designing the User Interface*. Wokingham: Addison-Wesley.

SIS. (1998). *ISO 9241 Ergonomic requirements for office work with visual display terminals (VDTs) – part 11*. Stockholm: STG.

SIS. (1999). *ISO 13407 Human-centered design processes for interactive systems*. Stockholm: STG.

Sparre, E., & Andersson, P. (1994). *Metodik för utvärdering av gränssnitt (Intern rapport)*. Västerås: ABB.

