


Usability Matters!

NUMMER 4
HÖSTEN 1996

Datasystem är inte användbara när de konstrueras. Användbarhet uppstår i mötet mellan system, användare och arbetsuppgift. Systemutvecklingen måste syfta till att skapa förutsättningar för användbarhet. Många olika yrkesroller kan och bör bidra. I det här numret beskriver vi några av dessa roller och deras betydelse. 

Inledning . . 3 Utvecklingsarbetet . . 4 Anskaffning . . 10 När systemet används . . 12

Nyhetsbladet utges av forskargruppen Usability Matters (UM) vid institutionen för datavetenskap, Linköpings universitet. Ansvarig utgivare är Jonas Löwgren.

UM består av gruppledaren Jonas Löwgren, doktoranderna Torbjörn Näslund, Mikael Ericsson, Stefan Holmlid och Martin Howard, samt Pär Carlshamre, WM Data. Vår adress är:

Institutionen för datavetenskap, Linköpings universitet, 581 83 Linköping.

Telefon (vx): 013 - 281000. Fax: 013 - 142231.

Email: {jlo, tor, miker, steho, marho}@ida.liu.se; par.carlshamre@stoc.wmdata.se

WWW: <http://www.ida.liu.se/labs/aslab/groups/um/>

Om du känner någon som vill ha Usability Matters! i fortsättningen är vi tacksamma för att få veta det. Hör också av dig om du inte själv vill ha flera nummer.

Skicka Usability Matters! till nedanstående person.

Jag har bytt adress. Den nya står nedanför.

Jag vill inte ha fler nummer av Usability Matters!

Namn: _____

Adress: _____

**Faxa till Torbjörn Näslund, 013 - 142231
eller skicka email till tor@ida.liu.se**

Användbara datasystem är inte användbara redan när de konstrueras. Användbarhet är en kvalitet som uppkommer *i mötet* mellan ett datasystem, dess användare, och den arbetsuppgift som användaren försöker utföra. Under systemutvecklingsarbetet måste man försöka skapa datasystem som *har förutsättningar att bli användbara*; datasystem som när de används erbjuder relevanta tjänster, är lätta att förstå, effektiva att använda, och som uppfattas behagliga eller till och med roliga att använda. Användbarhet är därför inte något som kan uppnås enbart med hjälp av skickliga systemutvecklare. Det finns många olika yrkesroller som kan bidra till att datasystem blir användbara.

Användbarhetsarbete i fyra faser

Användbarhetsarbete kan alltid utföras. De viktigaste rollerna för användbarhetsarbete skiftar dock mellan dessa olika typiska faser:

- **Systemutveckling:** tonvikten ligger här på att skapa ett datasystem som är så väl designat att det har förutsättningar att bli användbart när det används.
- **Anskaffande av datasystem:** beställaren eller köparen av ett datasystem har stor makt att påverka vilka kvalitetsegenskaper som betonas. Beställaren/köparen påverkar därför medvetet eller omedvetet den framtida användbarheten.
- **Systeminförande:** vid införande av ett nytt datasystem skapas förväntningar och farhågor som påverkar hur datasystemet uppfattas av de blivande användarna.
- **Användning:** samspelet mellan datasystem, användare och arbetsuppgifter är dynamiskt. Det både kan och bör vidareutvecklas kontinuerligt för att vidmakthålla och förbättra användbarheten.


Inledning

Torbjörn Näslund

Användbarhetsarbete görs
under alla faser i ett
systems liv. De berörda
yrkesrollerna varierar.

Detta temanummer ägnar vi åt att illustrera yrkesrollernas stora betydelse för användbarhet. Vårt syfte är att ge idéer. Däremot presenterar vi inga färdiga lösningar för samspelet mellan dessa roller. Varje organisation är unik, och måste utgå från sina egna förutsättningar. Vi gör inte heller några anspråk på att vara uttömmande — det finns många fler roller, som vi inte tagit upp.

Vi har valt att uttrycka oss i form av fiktiva intervjuer med företrädare för olika yrkesroller som har viktiga uppgifter i de olika faserna. Intervjuerna bygger på våra erfarenheter och vetenskapliga studier, men de är, som sagt, fiktiva.

Vi hoppas stimulera våra läsare att se sig om i sin egen organisation. Kanske upptäcker du att det finns outnyttjad kompetens i ditt eget företag. 

Utvecklings- arbetet

Det är nog vanligast att man förknippar användbarhetsarbete med systemutvecklingen. Här finns flera centrala roller.

Många gånger försöker man låta en **generell systemutvecklare** behärska alla moment i systemutvecklingen. Detta är dock en spännvidd i kompetens som blir allt svårare att upprätthålla. Mer specialiserade roller med tonvikt på användbarhetsarbete är t.ex. **interaktionsdesigner**, **grafisk designer** (speciellt för informationspresentation och multimediatillämpningar), **användbarhetsutvärderare**, och **teknikinformatör**.

Projektledare har en nyckelroll för att prioritera och fördela arbetet med aspekt på olika kvalitetsegenskaper, där framtida användbarhet givetvis är mycket viktig. **Verksamhetsexperten** och **verksamhetsrepresentanter** i systemutvecklingsarbetet bidrar med kunskap om arbetsuppgifter och användare. De kan också arbeta med anpassning av verksamheten parallellt med design av systemet.

En användbarhetsutvärderare och en interaktionsdesigner

Anna: ”Mitt företag har satsat en hel del pengar på ett användbarhetslab, där jag jobbar. Vi testar prototyper och färdiga system genom att låta användare lösa testuppgifter. Ofta videofilmar vi deras arbete och analyserar det efteråt, för att ta reda på var användbarhetsproblemen finns i systemet.

Själva labbet är egentligen inte så viktigt för användbarheten, utan vi skulle nog kunna göra våra studier med enklare utrustning. Däremot tror jag att det är viktigt att vi kan visa på en konkret satsning. Vi har ofta kunder på besök i labbet. En annan fördel med labbet är att de som utvecklade systemet kan sitta med och se testerna. Ofta tror de oss inte när vi berättar hur svårt testanvändarna har med deras lösningar, men när de fått se ett test själva är det mycket lättare att tala med dem efteråt.

Vårt största problem? Definitivt att vi kommer in för sent i projekten. Det händer ganska ofta att vi testar ett system som ska släppas om ett par veckor. Om vi då hittar användbarhetsproblem finns det inte tid att ordna dem, utan det blir i bästa fall i nästa version. Det känns inte så roligt.

Helst skulle vi vilja vara med från förstudien och framåt, för att tidigt få studera de blivande användarna och deras arbete. Dels är det lättare för oss att göra bra tester om vi vet vilka arbetsuppgifter systemet är tänkt att stödja för vilka användargrupper. Dessutom finns det utvärderingsmetoder som fungerar bra redan innan systemet är implementerat. Man kan användartesta pappersprototyper, eller inspektera designen med hjälp av tumregler eller psykologiska teorier.”

Utvärderaren granskar designförslag och prototyper ur användbarhetssynpunkt.

- Empiriska utvärderingar studerar typiska användares arbete med prototyperna.
- Analytiska utvärderingar granskar designförslagen med hjälp av expertkunskap.

Utvärderingarna gör att man kan åtgärda användbarhetsbrister i tid.

Interaktionsdesignern

utformar systemets utseende och beteende ur användarnas synvinkel. Typiskt vill designern tidigt ta reda på så mycket som möjligt om de situationer där systemet ska användas, för att sedan utforska ett antal alternativa utformningar.

Skisser, prototyper och andra tekniker används för att kommunicera designidéer och slutligen komma till ett beslut.

Bengt: ”Där jag jobbar har vi inga som du, som sysslar med formell användbarhetsutvärdering. Jag sysslar med interaktionsdesign: att utforma systemet som användaren kommer att uppleva det. För att kunna göra det måste jag veta mycket om vilka som ska använda systemet och till vad. Under förstudien intervjuar jag en massa blivande användare och besöker deras arbetsplatser om jag hinner. Sedan gäller det att bestämma vilka tjänster systemet ska erbjuda och hur användargränssnittet ska vara. Jag ritlar mycket för hand, skriver användningsscenarier och bygger små demonstrationer i olika prototypingverktyg.

Det känns ibland som att man förväntar sig att jag ska hitta den ’rätta’ designen direkt, men jag tycker inte att det fungerar så. Jag måste pröva så många idéer som möjligt och utforska olika möjligheter. Ibland är det knepigt att fatta besluten. Då vore det skönt att få hjälp med tidig utvärdering.

I en del projekt har jag varit den ende som sysslar med interaktionsdesign. Det är svårt, tycker jag. Många projektledare vill dela upp arbetet i funktionella delsystem, men ofta är det ju så att den uppdelningen inte stämmer med användarnas upplevelse av systemet. Då gäller det för mig att utforma och sprida en helhetsdesign snabbt, innan användargränssnittet till de olika delsystemen blir alltför olika. En enkel prototyp eller en serie skisser fungerar bättre än att skriva en style guide, tycker jag.”

— Passar era områden ihop?

Anna & Bengt: ”Definitivt! Det är nödvändigt att utvärdera designidéer, men det räcker inte. Det kostar mycket pengar att iterera och det gäller att hamna ganska rätt redan första gången. Då krävs det folk som har designkompetens och kan sätta en bra inriktning på arbetet under de tidigare faserna.”

Anna: ”Sedan händer det ofta att jag påtalar några användbarhetsproblem och får frågan om hur de ska åtgärdas. Sådant måste jag ju svara på. Det är nog faktiskt så att utvärdering och design måste hänga nära ihop, för att det ska bli ett tillräckligt bra resultat.”

Bengt: ”En annan sak som är bra med er utvärderare är att ni ofta har väldigt bra koll på de situationer där systemet ska användas. Metoder för användarstudier och sådant har ni också till hands. Men samtidigt tycker jag att ni blir lite för låsta till stöd för användarnas befintliga arbete. Ibland gäller det att hitta idéer som användarna inte ens kunde föreställa sig, men som visar sig vara bra i efterhand. Då måste man jobba tidigt och parallellt med många idéer. Sådant är vi duktiga på.”

Två teknikinformatörer

Carl driver ett eget företag som skriver dokumentation och utbildningsmaterial för datasystem. ”Jag får mina uppdrag från olika systemutvecklingsföretag”, berättar han. ”Ofta har dessa företag tidigare låtit systemutvecklarna själva skapa dokumentationen, men har nu insett att tekniker sällan kan skriva begripligt. Självt är jag utbildad informatör, och har arbetat både med utbildning och som skribent i många olika organisationer.

Jag har ett stort teknikintresse, men jag ser det faktiskt som en fördel att jag inte kan programmera. Jag bryr mig inte om hur datasystemen fungerar tekniskt, utan fokuserar på den roll som datasystemen kan ha i en verksamhet. Flera gånger har jag blivit uppringd av systemutvecklare som till en början har varit upprörda, och påstått att jag har missförstått deras datasystem. När vi sedan har pratat om det ett tag brukar vi ofta hamna in i en situation där systemutvecklaren utbrister i ett förvånat ’Jasså — är det så man brukar göra?! Jaa... i så fall kan man ju faktiskt använda datasystemet så som du beskriver det... Men det var inte så som jag hade

Teknikinformatörer är specialiserade på information till användare. De utformar ofta handböcker av olika slag, men kan också syssla med annan information i systemet: hjälptexter, ledtexter och felmeddelanden.

Det är fördelaktigt att integrera detta arbete med det övriga designarbetet.

tänkt!'. Jag ser på min roll som att skapa en brygga mellan teknikerns sätt att tänka om datasystemet, och hur användarna måste tänka för att få någon nytta av det.”

Debbie nickar instämmande när hon hör Carl berätta. ”Jag känner igen situationen, men vi har löst det på ett helt annat sätt. Jag är visserligen teknikinformatör, men jag är också en del av designgruppen här på SuperSoft. Man har lärt sig att min kompetens är viktig att ha med tidigt i systemutvecklingen. Jag måste ofta gripa in och förklara att det är skillnad mellan det som är enkelt och det som är flexibelt eller roligt. För användaren är det kanske inte så viktigt att det finns fyra sätt att göra samma sak. Omväxling kan också vara förvirrande. Min roll blir ofta att vara en moderator i designarbetet — att hejda teknikerna när de börjar skena iväg med allt mer fantastiska idéer, men ibland också att sätta fart på deras idéer genom att ge målade beskrivningar av vad vi sett under våra studiebesök hos tänkbara användare.

När vi slutligen enats i designgruppen går vissa vidare och arbetar med de tekniska detaljerna. Min roll är att skriva handböcker, ledtexter, hjälptexter och felmeddelanden. Men det är i de tidiga faserna som jag tycker att jag gör den stora nyttan! Att arbeta som fristående konsult och skriva en handbok i efterhand skulle inte passa mig.”

En projektledare

Erik: ”För mig är det inte någon tvekan om att man måste arbeta iterativt i användbarhetsorienterad systemutveckling. Man måste bygga prototyper och successivt lära sig allt mer genom att utvärdera dem. Vad jag är bekymrad över är att alla systemutvecklare som jag träffat på säger att de har förstått grundidéerna bakom prototyping, och att de är vana att arbeta med det. I

praktiken verkar det som om många arbetar helt planlöst, och ser prototyping som någon sorts ursäkt för att arbeta slarvigt och osystematiskt.

Som ett försök att stadga upp det iterativa arbetet har vi satt upp en stor whiteboardtavla i projektets konferensrum. På denna sammanfattar vi successivt vad vi har lärt oss av varje prototyp. Vad jag hoppas att åstadkomma är en balanserad syn på risktagandet i iterativ systemutveckling.

Å ena sidan är det viktigt att ta vara på det vi lär oss av varje prototyp. När vi utvärderar en prototyp kanske vi inser att vi är på helt fel spår. Då ska vi inte se det som ett misslyckande för våra idéer, utan i stället som att vi i tid har lärt oss något som annars kunde ha lett till en katastrof.

Å andra sidan vill jag undvika att vi sitter och bygger prototyper utan att ha någon nytta av dem. En prototyp som vi inte lär oss något av har varit ett slöseri med resurser. Regelbundet försöker jag också samla systemutvecklarna till diskussioner där vi ser tillbaka på processen. Den typiska frågan är 'OK, det här är vad vi har lärt oss hittills. Hade vi kunnat skaffat oss dessa lärdomar på ett enklare och snabbare sätt?' Många gånger inser vi att vi knappast hade kunnat det. I andra fall kommer det fram att vi hade kunnat skaffat oss precis samma kunskap med hjälp av en mycket enklare prototyp, som vi kunde ha skapat med papper och penna i stället för med två veckors programmeringsarbete. I andra fall kommer det fram att någon av oss faktiskt hade kunskapen som behövdes, men att vi inte frågade efter den utan satte igång att bygga i alla fall.

Om jag ska sammanfatta mina erfarenheter till en enda kort slogan får det bli något i stil med: 'Iterationer är nödvändigt, men det gäller att ha hård kontroll över varför vi itererar.' Glömmer vi bort det, tappar vi kontrollen över projekten. Det har jag sett alldeles för många exempel på."

Projektledaren förstår användbarhetsarbetet och behandlar det som en integrerad del av projektet. Det iterativa arbetssätt som präglar användbarhetsarbete kan kräva omprioriteringar och omprioriteringar.

Det är vanligt att prototyper och idéer måste revideras eller förkastas. Projektledaren måste kunna skilja dessa naturliga inslag i arbetet från signaler om att projektet har kört fast eller håller på att gå över styr.

Anskaffning

Man kanske inte tänker på det, men de som sköter anskaffandet av ett nytt system har mycket med användbarheten att göra.

Beställare av datasystem som skall utvecklas har stor möjlighet att betona vilka kvalitetsegenskaper hos datasystemet som är primära. Användbarhet är ett viktigt mål för datasystem som skall användas av människor. Det är viktigt att beställaren inser att man inte beställer sådana system på samma sätt som man specificerar tekniska system som endast skall samverka med andra tekniska system.

Köpare av redan färdiga system har en mer indirekt påverkan. Tillsammans med goda **produktrecensenter** kan de visa **marknadsavdelningar** vilka produktsegenskaper som är värdefulla. Det krävs stor kompetens för att göra riktiga val och skicka riktiga signaler till produktleverantörer.

En köpare

Fredrik har för företagets räkning köpt ett nytt order/lager/faktureringsystem: ”Det lättaste var att göra det som står i böckerna; att ta reda på företagets behov av systemtjänster och att sedan studera marknaden och se vilka färdiga O/L/F-system som erbjuder de tjänster som vi behöver. Det gick snabbt att sälla fram de system som erbjuder tjänsterna vi behöver.

Men bara för att systemen erbjuder rätt tjänster så betyder det ju inte att det är användbart för oss. Det kan man inte få reda på genom att tala med en inställsam försäljare. För att försöka ta reda på det har vi ägnat mycket tid åt studiebesök och diskussioner med företag som använder systemen för att ta reda på vilka problem de har och har haft, och vad de faktiskt tycker om systemen. Vidare har vi bestämt oss för att först hyra det system som vi tror på, och ta det i provdrift i fyra månader i en del av verksamheten. Vi har också anlitat en konsult som skall hjälpa oss att anpassa våra arbetsrutiner så att de passar ihop med systemet som vi köper, och kommer att satsa mycket på internutbildning.


Alla dessa åtgärder kostar oss en hel del extra pengar. Bara att hyra ett system som vi sedan i alla fall tänker köpa kostar oss massor. Men dels ser jag det som en sorts försäkringspremie, dels ser jag det som en investering i användbarhet. Att bara köpa ett system som erbjuder rätt tjänster och sedan hoppas på att det skall passa in i vår organisation alldeles av sig självt känns lite naivt.”

En beställare

Gunilla har i stället beställt ett helt eget system genom en form av kontraktsupphandling med en konsultfirma: ”En stor del av vår beställning är en kravspecifikation som beskriver våra krav med systemet. Våra krav är formulerade som mål som vi skall uppnå med hjälp av systemet: organisationsmål och användbarhetsmål. Däremot har vi inte varit särskilt detaljerade med att beskriva hur själva datasystemet skall se ut. Jag ser inte det som vår uppgift — det är en uppgift för specialister, och det är också därför som vi lägger ut arbetet i stället för att göra det själva. Vi är till exempel tydliga på hur snabbt det skall gå för vår personal att lära sig använda systemet, och vilken frekvens av olika typer av felhandlingar som får förekomma för att man ändå ska kunna säga att de har lärt sig att använda det. I vilken utsträckning som konsulterna sedan vill lägga sin tid på att skapa ett intuitivt lättlärt system, och i vilken utsträckning de i stället vill göra bra handböcker och utbildningsmaterial till ett lite mer komplext system är till stor del deras beslut. Vi har också skrivit in långsiktiga användbarhetsmål; om dessa inte uppfylls kommer vi att kräva förändringar eller helt enkelt lämna tillbaka systemet.

Jag har fått vara ganska tuff för att driva igenom denna form av upphandling. Flera systemutvecklingsföretag har protesterat och sagt att ’detta är inte en kravspecifikation — ni måste i stället beskriva exakt vad

systemet ska göra och hur det skall se ut'. Dessa har jag resolut skickat iväg med kommentaren 'Det är ett systemutvecklingsuppdrag som jag har erbjudit, och om ni inte ens är kompetenta att skilja mellan systemutveckling och programkonstruktion får ni söka lyckan hos någon annan än oss!'

Visst har det blivit dyrare att göra på detta sätt, men jag tror att det betalar sig i längden. Jag tror dessutom att vi effektivt har kunnat sålla bort en del oseriösa konsulter genom att ställa den här typen av krav." 

När systemet används

Vi för ihop de två faserna införande och användning, eftersom det ofta är svårt att dra gränsen.

Utbildare har en viktig roll för användbarheten, men det finns många ytterligare roller att uppmärksamma:

- **Användarna själva** har ett stort ansvar för användbarheten. Det är viktigt att reflektera över hur det egna arbetet utförs och om det finns alternativ.
- **Arbetskamrater** är en viktig källa för tips om hur arbetet kan utföras på andra sätt, och om hur datasystem kan användas så att de ger ett effektivt bidrag till verksamheten.
- **Chefer** måste stimulera reflektion, vidareutveckling och gruppdiskussioner kring arbetet och datasystemen. Utan sådan stimulans riskerar felaktiga arbetsätt att konserveras.
- **Tekniska resurspersoner** kan hjälpa till med att ställa in och anpassa datasystemen så att de verkligen blir användbara.
- **Verksamhetsmässiga resurspersoner** kan fungera som interna utbildare, skapa lathundar och instruktioner liksom specialanpassade kalkylark, dokumentmallar, register, makron osv.

Två utbildare

Hanna arbetar på ett utbildningsföretag med att ge kurser på standardprogramvara, som t.ex. ordbehandling och kalkylering. ”En väldigt vanlig situation är att jag får elever som är desperata efter att ha drunknat i information. I vissa fall berättar de om hur de har ägnat helgerna åt att plöja igenom tjocka handböcker, och hur hjälpsamma kolleger har försökt visa hur man ska använda programmet genom att med flinka fingrar hoppa fram och tillbaka mellan underliga kommandotangenter.

Vi glömmar ofta bort hur långt tid det tar att bygga upp en förståelse och en repertoar av fungerande handlingar när vi stöter på något nytt. Ett datasystem är oanvändbart när vi stöter på det första gången, men allteftersom vi lär oss det kan det bli mer och mer användbart för oss. Men det måste få ta sin tid! Dagens standardprodukter är ofta så kraftfulla att de många olika möjligheterna blir ett initialt hinder för användbarheten. Man måste gå steg för steg, och börja med att visa att systemet är användbart även med en mycket begränsad repertoar av handlingar, bara man har en någotsånär adekvat förståelse för vad det är som programmet gör. Då har man lagt grunden för att successivt bygga på förståelsen.

Tyvärr värderas vi utbildare sällan på rätt sätt. Jag får ofta kommentarer från erfarna besserwissrar som klagar över vår bristande kompetens — att vi lär ut omständliga omvägar i stället för att nämna att man kan få tjänsten utförd genom att trycka ’Control-Shift-Q Escape-vänster måsvinge’. Men vår kompetens är just att avgöra när det ökar användbarheten att nämna en viss funktion, och när det i stället ökar användbarheten att hålla tyst!

Ingvar: ”Det är exakt dina argument som vi på vårt företag använder för att förespråka en annan utbildningsform än den som du använder! Vi tar fasta på två

Rollen som **utbildare** kan variera kraftigt, till exempel:

- Utbildning kan ske initialt eller kontinuerligt.
- Utbildning kan vara anpassad till en specifik verksamhet eller generell.
- Utbildning kan vara individanpassad eller generell.


Gemensamt för all användarutbildning är att den kraftigt kan påverka användbarhet och nytta i användningssituationen.

saker: Att kompetens byggs upp successivt, och att kompetens byggs upp i ett sammanhang. Därför är det svårt att nå en tillräcklig kompetens genom att gå standardiserade kurser; det är på arbetsplatsen som kompetens — och därmed användbarhet — kan skapas.

Ett typexempel är kalkylprogram. Om man skickar personalen på en kurs i användning av kalkylprogram, så kommer de tillbaka fullärda i hur man skriver in formler, kopierar formler, löser ekvationer med hjälp av kalkylprogrammen, och så vidare. Men det stora steget måste komma på arbetsplatsen: 'Hur ska vi skapa de kalkylark och formler som vi behöver för vår verksamhet?' Det är först när man klarar av detta som datasystemen verkligen blir användbara.

Konsekvensen för oss är att satsa mycket lite på ordinarie utbildning, men i stället satsa på kunskapsöverföring och verksamhetsutveckling på arbetsplatsen. Alla får en mentor som skall hjälpa dem med användning av datasystemen, och dessutom satsar vi på utvecklingscirkel där vi ifrågasätter och vidareutvecklar verksamheten. Det är så som utbildning måste ske för att skapa verkliga effekter!"

Hanna: "Jo, det låter ju verkligen fantastiskt! Säkert är det ett jättebra komplement till min typ av utbildning. Men det är ju inte riktigt traditionell utbildning. Är det inte snarare så att du faktiskt är systemutvecklare? Du vidareutvecklar ju verksamhetssystemet!"

Ingvar: "Det spelar nog inte så stor roll vad man kallar det. Jag betraktar det som utbildning, eftersom det handlar om kompetensutveckling. Men huvudsaken är väl inte vilken etikett som man sätter på arbetet, utan att det faktiskt ökar våra datasystems användbarhet." 

Carlshamre, P. och Tumminello, J. (1995). Technical communicators' views on usability and collaboration.

Fem teknikinformatörer i USA och lika många i Sverige intervjuades om sin arbetsituation, speciellt användbarhetsarbete och samarbete med systemutvecklare. Resultaten visar att teknikinformatörer är värdefulla användbarhetsresurser, men hindras av sin egen och kundernas organisationer. Situationen är intressant nog densamma i USA och i Sverige.

Carlshamre, P. (1994). A collaborative approach to usability engineering: Technical communicators and system developers in usability-oriented systems development.

En utförlig rapport från Delta-projektet, där vi studerade införande av användbarhetsorienterade tekniker och samarbete mellan systemutvecklare och teknikinformatörer i professionell systemutveckling. Pär studie visar på goda möjligheter till användbarhetsarbete i praktiken, diskuterar några kritiska problem och visar hur de kan undvikas.

Löwgren, J. (1995). Perspectives on usability.

I den vetenskapliga MDI-litteraturen kan man finna olika sätt att se på begreppet användbarhet. Rapporten går igenom några olika synsätt och visar vilka konsekvenser de får för systemutveckling.

Löwgren, J., Carlshamre, P. och Näslund, T. (1995). Developing and studying usability-oriented methods in professional contexts.

Den här artikeln är en sammanfattning av vår praktiska nära forskningsfilosofi med två exempel: samarbete mellan teknikinformatörer och systemutvecklare i Delta-projektet, samt hur användbarhetsinspektioner fungerar i iterativ utveckling.

(forts på pärmens baksida)

Mer att läsa

Vi dokumenterar naturligtvis vår forskning i vetenskapliga publikationer. Här är ett aktuellt urval.

Om du vill ha kopior av våra skrifter så kontakta respektive författare eller Birgitta Franzén, tel: 013 - 282692, email: birfr@ida.liu.se

Näslund, T. (1996). Seven traditions for information systems development.

För att förstå systemutvecklingslitteraturen måste man förstå den tradition som varje bok eller artikel är skriven inom. I den här artikeln sammanfattas sju olika traditioner: administrativ systemutveckling, software engineering, participativ design, MDI, expertsystem, beslutsstödsystem och datorstött samarbete.

Näslund, T. (1995). Computers in context — but which context?

Artikeln bygger på en fallstudie från ett systemutvecklingsprojekt i näringslivet. Fyra grupper av projektdeltagare identifieras, var och en med sitt eget sätt att se på systemutvecklingen. Skillnaderna innebär att deltagarna kan fokusera på de aspekter som de är speciellt kunniga på, men medför också en risk att missförstånd uppstår.

Näslund, T. (1994). "Usability is extremely important — but it's somebody else's job, I hope".

Läroböcker i människa-datorinteraktion poängterar ofta vikten av att datasystemen blir användbara. Däremot är bilden en annan inom systemutveckling och software engineering. Systemutvecklingslitteraturen förutsätter typiskt att användbarhet ska ordnas vid programkonstruktionen, men ger inte många råd om hur det ska gå till. I böckerna om programvaruproduktion förutsätts i stället att användbarhetsfrågorna är utredda innan produktionen startar.

Nästa nummer kommer under våren 1997.

Det kommer att handla om visualisering som interaktionsteknik.