

Usability Matters!

NUMMER 1 1994

Att utveckla användbara datasystem är svårt. Genom det här nyhetsbladet vill vi skapa ett intresse för användbarhet och visa att vetenskapen kan bistå med mycket nyttig kunskap om användbarhetsorienterad systemutveckling. ✍️ Vi är en forskargrupp vid Linköpings universitet. Ett av våra mål är att bidra till mer användbara datasystem i samhället. Detta försöker vi uppnå genom att studera och lösa användbarhetsrelaterade problem i industriell systemutveckling, i samarbete med företagen. ✍️ Det här första numret har vi skickat till personer som vi tror kan vara intresserade. Om du vill diskutera något vi har skrivit, känner någon som också vill ha nyhetsbladet, eller kanske vill meddela att det inte var något för dig, så hör av dig!

Användbarhet. Självklart? . . 3 Omvärdera gränssnittsbegreppet! . . 7

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

UM består av gruppleadaren Jonas Löwgren och doktoranderna Torbjörn Näslund, Pär Carlshamre och Mikael Ericsson. Vår adress är:

Institutionen för datavetenskap, Linköpings universitet, 581 83 Linköping.

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

Email: {jlo, tor, parca, miker}@ida.liu.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

Det sunda förnuftet säger oss att ett datasystem bör vara till nytta för dem som använder det. Det bör också vara så lätt att lära sig och att använda att det är värt besväret. Systemet bör med andra ord vara *användbart*. Om ett företag kan utveckla användbara system utan att ge avkall på andra kvaliteter kan de räkna med nöjda kunder och mera affärer.

Det finns nog ingen som kan säga emot om man uttrycker sig på det här sättet. Men samtidigt är det svårt att hålla med utan vidare. Det behövs "hårdare" argument, och speciellt är det intressant att veta om användbara system kan ge direkta ekonomiska vinster. Ett känt exempel är systemet som användes av kundtjänsten vid Australiens televerk för att handlägga kundärenden. Vid en utvärdering av systemet 1992 fann man att en investering i användbarhet på 1,8 miljoner kr skulle återbetala sig på 15 dagar och därefter generera 120 000 kr om dagen i ren vinst (se exemplet på nästa sida).

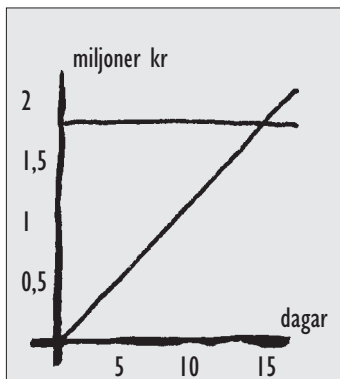
Det finns också andra goda exempel. Forskarna Lederer och Prasad gjorde 1992 en enkätundersökning av hur systemutvecklingsprojekt klarade av att hålla sin budget och vilka skäl som fanns till eventuella avvikelser. 115 projektledare svarade och det visade sig att 63 % av alla projekt gick kraftigt över budget. Av de tjugofyra skäl som uppgavs för detta hade de fyra vanligaste alla med användbarhet att göra. Användarna begärde ändringar; man hade förbisett vissa av användarnas uppgifter; användarna förstod inte sina egna krav; förståelsen mellan användare och utvecklare var bristfällig.

Typiska problem

Om det nu finns goda skäl att utveckla användbara system, hur kommer det sig då att det finns så många onyttiga, svårlärda och svåransända system? Vi tror att det finns många olika faktorer som samverkar.

Användbarhet. Självklart?

Det behövs forskning om användbarhetens roll och möjligheter i professionell systemutveckling. Vi beskriver här vår målsättning och våra pågående projekt.



En räkneövning

Det kostade ca 420 000 kr att hitta användbarhetsproblem, som sammantaget fördröjde användarnas arbete med 85 minuter per dag. Att åtgärda dem skulle kosta ca 1,4 miljoner till, och man bedömde att användarna skulle kunna använda hälften av den intjänade tiden — 42 minuter — till nyttigt arbete. Dessa 42 minuter är värda 120 000 kr per dag, om vi räknar med 2 500 tjänstemän och en årslön på 130 000 kr. Investeringen betalar sig alltså på femton arbetsdagar.

- Efterfrågan på användbara system är förvånansvärt låg. Upphandlare av kontraktprojekt saknar ibland insikt om de blivande användarnas arbete och behov. Då är det svårt att formulera krav på användbara system. Kunder som köper hyllprodukter har inte alltid den kunskap som behövs för att kräva användbarhet. Andra faktorer, som t ex tidigare erfarenheter, får styra köpet. Eftersom det kostar extra att utveckla användbara system kommer det inte att ske så länge det inte efterfrågas.
- Vid professionell utveckling av programvara vill man tidigt formulera och fastställa kraven på det blivande systemet. Självfallet är det bra att få en måttstock för att kunna avgöra om det färdiga systemet håller måttet, men en kravspecifikation kan faktiskt försämra användbarheten. Om den inte innehåller tydliga och observerbara användbarhetskrav kan man leverera ett oanvändbart system som ändå uppfyller specifikationen. Om den är detaljerad men dåligt underbyggd är risken istället att man för tidigt styrt in på en dålig lösning.
- De utvecklingsverktyg som finns idag för att stödja konstruktion av användargränssnitt är mycket begränsande. Att välja verktyg för tidigt kan innebära att man också väljer bort en stor mängd designmöjligheter som kunde varit värdefulla.
- Åtskilliga forskare har studerat problemet att utveckla användbara system. Det finns stora mängder vetenskaplig kunskap i form av metoder, verktyg och tekniker. Tyvärr är den ofta svårtillgänglig, där den står på forskningsbibliotekets hyllor. Den är inte heller alltid så relevant; mycket forskning ägnas åt att lösa problem som bara är intressanta för andra forskare.

Våra mål

Vi som kallar oss Usability Matters eller UM är en forskargrupp inom institutionen för datavetenskap vid Linköpings universitet. Organisatoriskt hör vi till laboratoriet för applikationssystem, som leds av professor Sture Hägglund.

Vårt övergripande mål är att samhället ska få mer användbara datasystem. För att uppnå det övergripande målet har vi ställt upp delmål både för professionella utvecklare och för forskarsamhället. Vi vill öka motivationen och höja kompetensen för användbarhetsarbete hos praktiker. För forskarsamhället vill vi ge exempel på industriellt relevant forskning, som samtidigt har vetenskapligt värde. Slutligen har vi också personliga mål, som bl a omfattar akademiska avhandlingar.


För att nå våra mål arbetar vi i projekt, där vi strävar efter samarbete med olika utvecklingsföretag. Varje projekt har inslag av både professionella och vetenskapliga mål. Vi kan alltså inte åta oss konsultuppdrag utan vetenskapligt värde, men vi gör inte heller grundforskning utan näringslivsrelevans. Här vill vi berätta kort om våra olika projekt.

Torbjörn Näslund forskar kring hur utvärderingar av en prototyps användbarhet påverkar utvecklingsarbetet. Torbjörn har systemvetenskaplig bakgrund och är en skicklig utvärderare. Han har medverkat i flera industriella utvecklingsprojekt, utvärderat prototyper i olika stadier och studerat vilka faktorer som påverkar utvecklarnas sätt att använda synpunkterna.

Pär Carlshamre studerar samarbete mellan systemutvecklare och teknikinformatörer. I ett nyligen avslutat projekt utvecklade vi, i samarbete med ett konsultföretag, en systemutvecklingsmetod som bl a innehöll sådant samarbete. Vi utvärderade metoden genom att använda den i ett av företagets kundprojekt. Pär var

med som utvecklare i kundprojektet och fick på så sätt en mycket god bild av hur metoden och samarbetet fungerade.

Mikael Ericsson är intresserad av datorstöd för design av användargränssnitt. Det finns många verktyg för att realisera en design. Vi har tidigare visat att det går att komplettera sådana system med en modul som kommenterar det framväxande användargränssnittet. Mikael kommer med hjälp av experiment att studera hur ett sådant system ska bete sig för att verkligen stödja designarbetet. Ska systemet avbryta designern så fort det finns något att kommentera, eller ska designern själv begära kommentarer? Ska kommentarerna presenteras i text eller grafiskt?

Jonas Löwgren deltar i ett Esprit-samarbete med flera universitet och företag i Europa, där vi har uppgiften att studera verktyg för design. Vi sysslar främst med att utveckla stöd för resonemangen kring en designlösning. Vårt mål är att åstadkomma en utvecklingsmiljö som stödjer samarbete och resonemang om olika designlösningar och deras för- och nackdelar. På sikt räknar vi med att kunna pröva våra idéer bland professionella utvecklare. 

Debatt

Det är inte bara systemutvecklare som ser till att data-system blir lätta att använda. I många böcker inom MDI-området framhålles t ex kognitionspsykologer och ibland till och med antropologer som så kallade användbarhetsresurser. Användbarhetsresurser är resurser, i form av t ex verktyg eller kunskap, som är bra att ha till hands vid utveckling av användbara system. Få företag har dock tillgång till kognitionspsykologer, och förmodligen än färre till antropologer. Vad man dock oftare har tillgång till är teknikinformatörer, dvs de som utvecklar olika typer av hjälpsystem i form av t ex användarhandledningar och online-hjälp. Dessa teknikinformatörer har visat sig utgöra en mycket god användbarhetsresurs, eftersom de dels är vana att ha kontakt med användare och dels har kännedom om vad användare normalt tenderar att ha problem med — det är ju deras jobb!

Ett problem i sammanhanget är dock den vanliga synen att gränssnitt och användardokumentation är två separata enheter. I Deltaprojektet samarbetade just systemutvecklare och teknikinformatörer i ett användbarhetsorienterat utvecklingsprojekt. Vår intention var att betrakta användargränssnittet och användardokumentation som en enhet. Detta visade sig emellertid problematiskt, i och med att separationen manifesterades på så många olika nivåer.

- På den lägsta nivån visade det sig att utvecklingsverktyg är anpassade efter den allmänna synen att det finns en skiljelinje mellan gränssnitt och dokumentation. De stödjer inte ett samarbete av den typ som vi ville uppnå, utan snarare tvärtom. Det finns dokumentationsverktyg och gränssnittsverktyg, men ingen som helst koppling däremellan.

Omvärdera gränssnittsbegreppet!

Pär Carlshamre

I Delta-projektet studerade vi samarbete mellan systemutvecklare och teknikinformatörer. Vi fann att skiljelinjen mellan användargränssnitt och användardokumentation är problematisk och hämmar användbarhetsarbetet.

Delta-projektet

. . . var ett samarbete mellan Ericsson Infocom Consultants AB och forskargruppen UM. Syftet med projektet var att utveckla en användbarhetsorienterad systemutvecklingsmetod, som bättre tillvaratog den erfarenhet som teknikinformatörer har av att arbeta nära användarna.

Projektet, som delvis finansierades av Nutek, startade under våren 1992, och har för Infocom del resulterat i två metodhandböcker för olika målgrupper. Dessa är nu tillgängliga för alla intresserade (se kontaktnamn nedan). För UM-gruppens del resulterade projektet i en licentiatavhandling samt ett antal publikationer.

För mer information om Deltametoden, kontakta

Martin Rantzer
Ericsson Infocom Consultants AB
Datalinjen 3, 583 30 Linköping

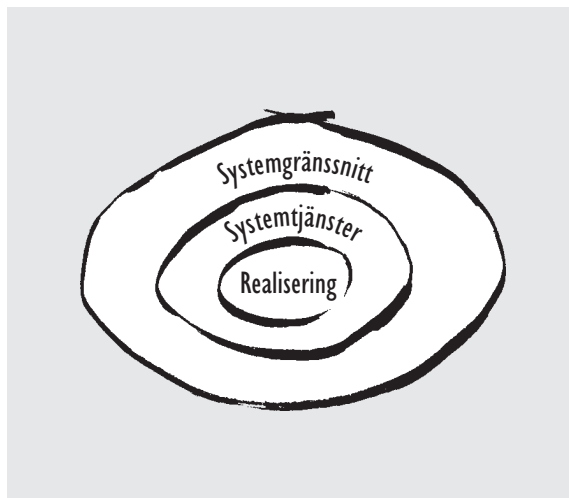
Se också "Mer att läsa" på sid. 11.

- På organisatorisk nivå såg vi nästa problem, som till stor del beror av olika traditioner inom respektive område. Exempel på sådana är olika syn på arbetsmetoder, resursallokering och projektplanering.
- På den allra översta nivån finns det en stor (och i litteraturen tämligen väl beskriven) kulturskillnad mellan teknikinformatörer och ingenjörer. Denna bottnar ofta i ömsesidig bristande förståelse och respekt för varandras kompetenser och färdigheter. Många utvecklare betraktar t ex teknikinformatörens arbetsuppgifter som inte bara tråkiga, utan också mindre krävande: vem som helst kan väl *skriva!*?

Finns det då någon naturlig skiljelinje mellan gränssnitt och dokumentation, eller är dessa problem resultatet av ett alltför snävt tankesätt? Om skiljelinjen finns, var går den? Om en knapp hör till gränssnittet, vart hör då texten på knappen? Vart hör texten i menyerna? Eller de små ledtexterna i statusfältet längst ned i ett fönster? I Deltaprojektet visade det sig att gränsen bestämdes av bland annat vad det använda verktyget klarade av att hantera. De delar som inte stöddes av verktyget och därmed måste implementeras i källkoden fick således systemutvecklarna ta hand om. De kom därför implicit att utgöra delar av gränssnittet, snarare än dokumentationen. Det finns således inga inneboende egenskaper hos de olika objekten som avgör huruvida de skall betraktas som hörande till den ena eller andra sidan. Om det finns en skiljelinje, så är den väldigt vag och svårdefinierad. Trots detta har den alltså fått långtgående konsekvenser.

Vi vet nu att teknikinformatörer är värdefulla i användbarhetsorienterad systemutveckling, att vårt nuvarande gränssnittsbegrepp hindrar ett sådant samarbete, och att detta gränssnittsbegrepp saknar egentlig grund. En naturlig följd borde vara att gränssnitts-


begreppet omvärderas. Man borde helt enkelt utöka det till att omfatta även vad som idag allmänt kallas användardokumentation. Figuren här intill visar en modell som tar hänsyn till problemen, och som erbjuder en naturlig gemensam bas för teknikinformatörer och systemutvecklare. Den är lättast att beskriva med utgångspunkt från det mittersta skiktet, nämligen systemtjänsterna.



En modell för det utökade gränssnittet.

Systemtjänster är de tjänster som ett system erbjuder användaren, t ex att rita ett diagram över ett antal kolumner i ett kalkylark. Systemtjänsterna är således konceptuella, d v s frikopplade från källkodsfunktioner. De källkodsfunktioner som kan behövas för att realisera dessa tjänster, t ex grafikprimitiver, matematiska rutiner, etc, återfinns i kärnan i figuren. Systemgränssnittet blir nu allting som hjälper användaren att utnyttja en systemtjänst, såsom menyer, ledtexter, online-information, delar av kursmaterial, knappar, o s v.

Detta utökade gränssnitt inte bara tillåter utan *uppmuntrar* en helhetssyn på gränssnittsdesign, där vad vi idag kallar dokumentation integreras på ett naturligt sätt. Det uppmuntrar således vad som kallas för konceptuell design, dvs att man fokuserar på systemets *tjänster* snarare än vilka källkodsfunktioner som skall implementeras, på så sätt att systemtjänsterna är frikopplade från realiseringen. Vidare ges utrymme för de olika specialistkompetenser som teknikinformatörer och systemutvecklare har, så att var och en helt enkelt gör det man är bäst på. Således, när det är dags för realisering så programmerar ingenjörerna och skriver teknikinformatörerna.

Det utökade gränssnittet ställer emellertid nya krav på de verktyg som används i utvecklingsarbetet. Dagens gränssnittsverktyg är alltför begränsande, och klarar till exempel inte av att hantera hjälp i någon annan form än text kopplad till gränssnittsobjekt. Dessutom är de så gott som uteslutande uppbyggda kring ett antal grafiska gränssnittsobjekt, som knappar, menyer och hjälptexter. Ett utvecklingsverktyg som stödjer det utökade gränssnittsbegreppet måste kunna hantera *systemtjänster och uppgifter i första hand*. Själva gränssnittsobjekten utgör därmed bara olika möjliga medel att underlätta för användaren att utnyttja dessa tjänster. Detta får dock bli ett ämne för en senare artikel. 

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 teknikinformatorer i professionell systemutveckling. Pärns studie visar på goda möjligheter till användbarhetsarbete i praktiken, diskuterar några kritiska problem och visar hur de kan undvikas.

Carlshamre, P. (1994). Technical communicators and system developers collaborating in usability-oriented systems development: A case study.

En mera kortfattad beskrivning av resultaten från Delta-projektet. Speciellt fokuseras samarbetet mellan systemutvecklare och teknikinformatorer.

Löwgren, J. (1994). Empirical foundations for design rationale as user-interface design support.

Det finns många som hävdar att design rationale, alltså notationer för att dokumentera argumentationen kring en design, vore ett värdefullt hjälpmedel för designarbete. Ett krav är dock att detta designarbete bedrivs i tät växling mellan skapande och reflektion. Den här studien visar att yrkesverksamma designers växlar mycket ofta, och dessutom begår vissa typer av misstag som skulle kunna avhjälpas med en argumentativ designnotation.

Löwgren, J. och Laurén, U. (1993). Supporting the use of style guides and guidelines in professional user-interface design.

Tekniskt sett är det fullt möjligt att lagra regler för design av användargränssnitt i ett kunskapssystem som kan analysera och kommentera ett designförslag. Vi

(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 Lena Larsson, tel: 013 - 282692, email: lenla@ida.liu.se

redovisar här en undersökning av den nytta yrkesverksamma utvecklare skulle ha av ett sådant stöd, och vilka krav de skulle ställa på det.

Näslund, T. (under tryckning). Supporting design communication with explicit representation of evaluation feedback.

Formativa utvärderingar av användbarhet är en del av designargumentationen kring ett framväxande datasystem. Artikeln illustrerar hur en notation för designrationale kan användas för att lyfta fram designargumentationen som finns underförstådd i utvärderingsrapporter från formativa utvärderingar.

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. Hur är det då med vanliga läroböcker i systemutveckling och programvaruproduktion (software engineering)? Artikeln visar att båda typerna av böcker tenderar att bortse från användbarhet. 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 i början av 1995.

Det kommer att handla om användbarhetens roll i professionell systemutveckling.