


Usability Matters!

NUMMER 2
VÅREN 1995

När det gäller användbarhetsarbete i professionell systemutveckling finns det inga enkla svar. Hur man ska arbeta beror i stället på förutsättningarna. Olika typer av projekt ger olika möjligheter. Kontraktprojekt ställer till exempel helt andra krav än produktutveckling (sid 3). Det är också nödvändigt att lära av andras erfarenheter. På sidan 9 tipsar vi om en färsk bok med fallstudier av professionellt användbarhetsarbete. 

Användbarhet i sitt sammanhang . . 3 Professionellt användbarhetsarbete . . 9

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

UM består av gruppledaren 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 är inte lätt att säga allmänna sanningar om användbarhetens roll i professionell systemutveckling. Sådant som är centralt för ett projekt kan vara totalt ovidkommande för ett annat. Vi måste alltså också göra klart vilket sammanhang vi talar om för att kunna diskutera användbarhetens roll.

Att precisera sammanhanget skulle kunna betyda att beskriva varje enskilt projekt, på en sådan detaljnivå att läsaren eller lyssnaren kunde sätta sig in i systemutvecklarens eller projektledarens roll och förstå varje beslut och handling. Det är egentligen en bra idé, eftersom varje projekt på sätt och vis är unikt. Dock skulle erfarenheterna bli svåra att tillämpa i andra projekt.

En annan variant är att försöka hitta kategorier som är generella nog att täcka in många systemutvecklings-sammanhang men samtidigt så skarpa att de säger oss något. En forskare vid namn Jonathan Grudin skrev för några år sedan en artikel där han föreslår tre sådana kategorier: *kontraktsupphandling*, *produktutveckling* och *internutveckling*.

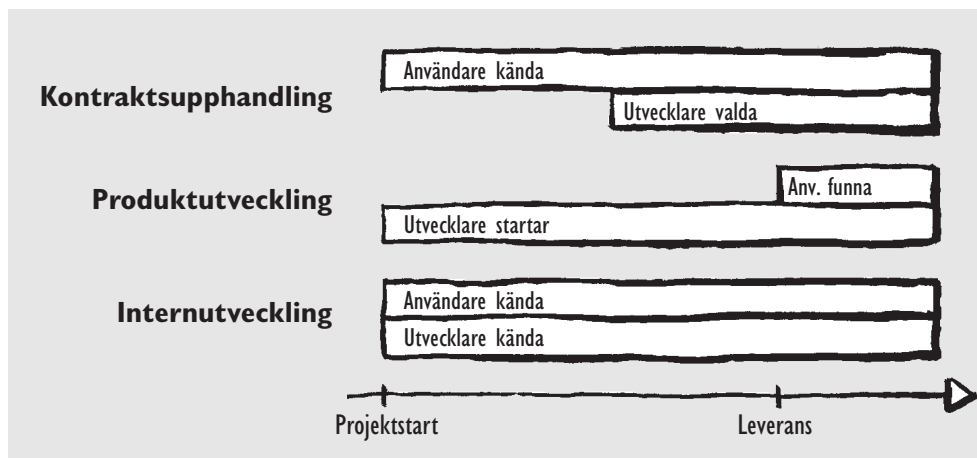
Enkelt uttryckt kan man säga att kontraktsupphandling är projekt där en kund beställer ett specialutvecklat system av en leverantör. Produktutveckling avser utvecklingen av system för en bredare marknad. Vid internutveckling handlar det typiskt om att ett företags dataavdelning utvecklar ett system åt en annan avdelning inom företaget. På nästa sida finns en mer precis definition av de olika kategorierna.

Naturligtvis innebär det alltid en överförenkling att börja tala om så här breda kategorier — interna dataavdelningar kan mycket väl tänkas lägga anbud på offertförfrågningar från andra avdelningar, produktutvecklare kan börja med att utveckla en skräddarsydd version åt en viss kund för att lära sig problemområdet. Poängen är ändå att synen på användbarhetsarbete blir helt olika i de tre kategorierna. För att illustrera det

Användbarhet i sitt sammanhang

Jonas Löwgren

Finns det ett bästa sätt att
uppnå användbarhet i alla
typer av projekt?
Självfallet inte. Det gäller
istället att anpassa
användbarhetsarbetet
till sammanhanget.



- Kontraktsupphandling börjar med användarna. Användarorganisationens upphandlare lägger ut en offertförfrågan som utvecklingsorganisationer besvarar med offerter. Upphandlarna väljer en leverantör.

- I produktutveckling är situationen den omvända. Utvecklarna satsar på en ny produkt och exakt vilka som ska använda den är inte känt. Man kan dock förmoda att marknadsavdelningen har en ganska god uppfattning om avsedd målgrupp.

- Internutveckling är den enda typ där både utvecklare och användare kan pekats ut när projektet börjar. Det kan t ex röra sig om att dataavdelningen ska utveckla ett skräddarsytt system för en annan avdelning inom företaget.

kommer jag nu att beskriva ett bra och ett dåligt exempel för varje kategori. Exemplet är påhittade och möjligen överdrivna, men de kan tjäna som hjälpmedel för tankar kring användbarhetens roll i systemutveckling.

Kontraktsupphandling: det dåliga exemplet

Den upphandlande organisationen tänker skaffa ett nytt system och i de inledande diskussionerna talas det om hur viktigt det är att det nya systemet blir användbart. Upphandlarna gör en utredning som fokuseras på tekniska behov. En kravspecifikation upprättas med främst tekniska krav. Den innehåller också kravet att systemet ska godkännas av användarrepresentanter vid leverans.

Olika utvecklingsorganisationer lämnar anbud på att bygga ett system enligt specifikationen. Upphandlarna väljer den billigaste, som inte innehåller några resurskrävande användbarhetsorienterade aktiviteter. Utvecklarnas arbete inriktas helt på att uppfylla specifikationen. När tiden börjar bli knapp väljer de till och med lösningar som uppenbart kommer att bli besvärliga för användarna. Vid leveranskontrollen hamnar användarrepresentanterna i en besvärlig sits: de skulle vilja anmärka på en del saker, men upphandlarna mer eller mindre tvingar dem att godkänna systemet eftersom budgeten är förbrukad och projektet får inte kosta mer.

Kontraktsupphandling: det goda exemplet

Den upphandlande organisationen tänker skaffa ett nytt system och i de inledande diskussionerna talas det mycket om hur viktigt det är att det nya systemet blir användbart. Första steget är att anlita en skicklig konsult som gör en verksamhets- och förändringsanalys och en användarstudie på de avdelningar som är berörda. Resultatet av två manmånaders förarbete är en uppsättning mätbara *verksamhets-* och *användbarhetsmål*. Upphandlarna identifierar därtill ett antal krav på *processer* som utvecklingsorganisationen måste följa för att man ska kunna säkerställa att målen uppnås.

Efter utvärdering av anbuden skapas arbetsgrupper på olika nivåer mellan upphandlingsorganisationen och den valda utvecklingsorganisationen. Utvecklare får möjlighet att samarbeta med blivande användare för att precisera målen vid behov. Upphandlarna diskuterar former för acceptans och målrevision med utvecklingsorganisationens ledning. Utvecklingen tar formen av en iterativ process där inte bara systemets utformning diskuteras, utan också den förståelse för verksamhets- och användbarhetsmålen som gradvis växer fram. Det slutar med att de prioriterade målen uppfylls medan vissa av de andra målen avfärdats som mindre viktiga.

Användbarhet i kontraktsupphandling

I det dåliga exemplet kan vi se hur utvecklingen styrs av specifikationen, som ses som fastslagen. Eftersom upphandlarna skrivit specifikationen själva reflekterar den heller inte användarnas behov särskilt väl. Projektkontraktet stipulerar att användarna ska godkänna systemet vid leverans, men där finns inget som reglerar hur eventuella förändringar ska göras.

I det goda exemplet kan vi speciellt märka att de som verkligen ska använda systemet är aktiva och man utnyttjar deras speciella kompetens inom arbetet som ska stödjas. Specifikationen innehåller mål för det nya, datorstödda arbetet och för datorstöddets användbarhet. Det finns också fastställda former för prioriteringen av de specificerade målen. Projektet är väl förankrat i köpar- och leverantörsföretagens ledning.

Produktutveckling: det dåliga exemplet

På hög nivå i produktutvecklingsföretaget fattas ett strategiskt beslut om att försöka slå sig in i ett nytt och viktigt marknadssegment genom en ny produkt. Detta beslut anses som alltför känsligt att sprida i hela företaget — utvecklarna får därför bara veta i allmänna ordalag vad den nya produkten ska kunna göra. De studerar jämförbara produkter och konstaterar att det främsta konkurrensmedlet kommer att bli användbarhet. Prototyper utvecklas och användbarhetstestas med hjälp av frivilliga testpersoner som har ungefär samma datorvana som de avsedda användarna. Betaversioner sprids också till facktidningarnas produktredaktörer och till ett antal inköpare hos etablerade kunder, och den färdiga produkten anpassas till reaktionerna.

Produktutveckling: det goda exemplet

Produktutvecklingsföretaget fattar ett beslut om att slå sig in i ett nytt och viktigt marknadssegment. Marknadsavdelningen och utvecklarna samarbetar i en större studie av användarnas behov och av de produkter som redan finns på marknaden. Målet för den nya produkten formuleras som ett *affärsmål*. Ett samarbetsavtal tecknas med en *användarorganisation* och en första version av den nya produkten utvecklas i samarbete med

Användbarhet i produktutveckling

Det dåliga exemplet visar hur facktidningar, inköpare och konkurrerande produkter styr arbetet medan användarna helt försvinner. Eftersom utvecklarna inte får del av det strategiska målgruppsbeslutet tvingas de testa sina prototyper på människor som inte vet särskilt mycket om det arbete som produkten är tänkt att stödja. Användbarhetsarbetet förvandlas till kosmetiska åtgärder, där man testat om gränssnittet är lätt att använda men inte om man bygger rätt system.

I det goda exemplet deltar utvecklarna från början i studien av marknaden och den tilltänkta målgruppens behov och förutsättningar. Genom att knyta användare till sig kan utvecklarna försäkra sig om att de bygger rätt system. Att uttrycka ett affärsmål för den nya produkten gör att man koncentrerar sig på "time-to-profit" i stället för "time-to-market", och därför arbetar igenom produkten på alla nivåer.

dem. Den testas sedan på två nivåer. När det gäller att testa hur väl produktidén stödjer de avsedda användarnas arbete tar man den kostnad det innebär att göra testerna hos tänkta kunder. Tester av hur begriplig och hanterlig produkten är på mera allmän nivå görs i användbarhetslabbet med en bred grupp av testpersoner. Produkten släpps på marknaden när utvecklarna, användbarhetsexperterna och marknadsavdelningen kan enas om att den är mogen för att uppfylla affärsmålet.

Internutveckling: det dåliga exemplet

Dataavdelningen på ett större tjänsteföretag får i uppgift att utveckla ett nytt system för intern kostnadsuppföljning. Alla inblandade är eniga om vikten av att det nya systemet blir användbart; därför utses en grupp användarrepresentanter från olika avdelningar att medverka i projektgruppen. Arbetet börjar med långa diskussioner om vad systemet ska kunna göra. Utvecklarna bygger en prototyp av det nya systemet och demonstrerar den på ett projektmöte. Användarrepresentanterna kommer med en mängd synpunkter och utvecklarna förändrar prototypen. Vid nästa möte finns en mängd nya synpunkter, och prototypen förändras igen.

Nu börjar användarrepresentanterna bli illa till mods — samtidigt som de fortfarande inte är nöjda med prototypen, känner de att de inte kan klaga mer på sina nya bekanskskaper från dataavdelningen. Efter ett par förändringsvarv till beslutar man att prototypen får gälla. Utvecklarna implementerar ett driftssystem, installerar det, lämnar dokumentationen till en systemförvaltare och avslutar projektet. Ett par månader efter installation är förvaltaren dränkt i klagomål och förändringsförslag, och en tid senare går man tillbaka till det gamla systemet.

Internutveckling: det goda exemplet

Dataavdelningen på ett större tjänsteföretag får i upp-
gift att utveckla ett nytt system för intern kostnads-
uppföljning. Diskussioner med ekonomiavdelningen
och en lång serie av studiebesök och intervjuer på olika
avdelningar visar att det gamla systemet är mycket be-
svärligt. Det visar sig vara viktigt att det nya blir mer
användbart. Företagsledningen sätter av en timme i
veckan för alla berörda anställda, som utvecklarna ut-
nyttjar för att starta *diskussionsmöten* på varje avdelning
om kostnadsuppföljning. Vid varje möte medverkar
ekonomer och utvecklare som dokumenterar de an-
ställdas behov av tekniskt stöd, dokumentation, utbild-
ning och underhåll. Mellan veckomötena diskuterar
utvecklarna sina erfarenheter och skapar en *gemensam
bild* av behoven.

När utvecklarna har fått en hygglig bild av vad som
behövs bygger de en prototyp som introduceras på de
olika avdelningarna vid veckomötet. Prototypen an-
vänds parallellt med det gamla systemet av frivilliga
anställda under en tid. Erfarenheter och problem tas
upp på veckomötena och ”testpiloterna” kommer med
svaren minst lika ofta som utvecklarna. När prototypen
stabiliserats bildas en *förvaltningsgrupp* med några eko-
nomer, utvecklare och testpiloter, som arbetar vidare


Användbarhet i internutveckling

Det dåliga exemplet visar främst på risken med att en liten grupp utsedda användarrepresentanter gradvis hamnar i en gisslansituation. Deras lojalitet mot sina kollegor bryts mot den nya lojalitet de efter hand får mot projektgruppen. Vi ser också hur utvecklingen och förvaltningen av systemet skiljs åt, och hur problemen förvärras av att projektet inte är särskilt väl förankrat bland de övriga anställda.

I det goda exemplet blir systemutvecklingsprojektet till en angelägenhet för hela företaget. De blivande användarna får påverka systemet som gradvis växer fram och känner sig delaktiga i förändringen. Vissa tar ett större ansvar för det nya systemet. Dem utnyttjar man vid införande och förvaltning. Även den tekniska och ekonomiska kompetensen säkras i systemets förvaltning genom att utvecklare och ekonomer som var med vid utveckling finns kvar i förvaltningsgruppen.

med underhåll och utbildning. Veckomötena förvandlas så småningom till resurstillfällen där problem och utvecklingsmöjligheter diskuteras.

Avslutning

Vi har sett att användbarhet och användbarhetsarbete kan bestämmas av helt olika faktorer i olika utvecklings-sammanhang. Du kanske tycker att de tre olika kategorierna ger en alltför enkel bild av verkligheten. Det är möjligt att du har rätt, men kategorierna ger oss möjlighet att renodla och förhålla oss till tanken om användbarhet i olika sammanhang. Kände du igen dig och dina erfarenheter i något av exemplen? Finns det något du kan lära dig av de goda exemplen? 

Källor

Den artikel som definierar de olika systemutvecklingssammanhangen är J. Grudin: Interactive systems: Bridging the gap between developers and users. IEEE Computer, sid 59–69, April 1991. Den är förstås mycket läsvärd om man tycker att tankarna är intressanta och vill fördjupa sig lite.

I den nyutkomna boken "Usability in practice: How companies develop user-friendly products" får man lära sig mycket om hur frågor kring användbarhet hanteras av sjutton framstående företag baserade i USA; de flesta av dem dock med multinationell verksamhet. I vart och ett av bokens kapitel beskriver experter från respektive företags avdelning för användbarhet hur man arbetar. Boken ger därigenom en unik inblick i hur användbarhetsfrågor hanteras i praktiken.

Företagen som får komma till tals representerar flera olika branscher. Vissa av företagen framställer konsumentprodukter som kameror, videobandspelare och datorer. Andra företag framställer programvara, för massmarknad eller för bruk i den egna verksamheten. Gemensamt för företagen är dock att de har skaffat sig en större eller mindre avdelning som arbetar med användbarhetsfrågor i anslutning till produktutvecklingen.

Professionellt användbarhetsarbete

Torbjörn Näslund

Exempel på användbarhetsarbete i praktiken kan man hitta i en nyutkommen bok.


Fullständig referens

Michael E. Wiklund (redaktör):
Usability in practice: How companies develop user-friendly products. AP Professional, Boston, 1994. ISBN 0-12-751250-0.
609 sidor.

För mig som systemutvecklare är det frapperande att få läsa hur insikten om användbarhet har vuxit fram på företagen. En mycket vanlig början verkar ha varit att beteendevetare har fått lov att göra studier av hur användbara företagens produkter verkligen är, och påvisat användbarhetsbrister. Dessa påvisade brister har sedan fått produktutvecklare, designers och chefer på företagen att få upp ögonen för användbarhetsfrågor. Typfallet tycks dessvärre ha varit att produktutvecklarna inte själva har varit bekymrade för den bristande användbarheten hos produkterna förrän de närmast har fått det skrivet på näsan.

Många av företagen har en betoning på beteendevetenskapliga utvärderingar när en produkt är nästan helt färdig, medan man fortfarande arbetar med frågan om hur man ska få in användbarhetsfrågor tidigt i designarbetet.

Många av författarna är förvånansvärt öppna när det gäller att beskriva hur man arbetar på företagen. Det märks att detta är ett område där de professionella engagerar sig i sitt arbete, och verkligen vill sprida kunskap om hur man åstadkommer användbara produkter. Samtidigt är det givetvis så att många av författarna bara betonar de positiva erfarenheterna — det ska mycket till för att man öppet ska erkänna alla misstag som man har gjort.

Kapitlen är av varierande kvalitet med avseende på nyhetsvärde. Generellt för hela boken gäller dock att man som läsare får en god insikt i hur frågor kring användbarhet hanteras i praktiken. Boken utgör därför ett enastående komplement till forskningsrapporter och annan forskningsorienterad litteratur om användbarhet. Jag kan varmt rekommendera den till såväl praktiker som praktikintresserade forskare. 

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. Pärs 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 Birgitta Franzén,
tel: 013 - 282692,
email: birfr@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). 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). 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 argumentationen.

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 efter sommaren 1995.

Det kommer att handla om användbarhetsorienterade utvecklingsverktyg.