

# **SISU** **dokument**

## **En metod för användbarhetsdriven prototyputveckling**

Peter Rosengren  
Ulf Wingstedt

**nr**  
**13**

# 1. Introduktion

Idag är de flesta överens om att en av de viktigaste egenskaperna hos ett datoriserat informationssystem är dess *användbarhet* (usability). Dåligt utformade system som inte är anpassade för sina användare och deras arbetsuppgifter leder till krånglig och ineffektiv användning. Ofta försvinner förväntade vinster av systemet till följd av att användarna gör fel, tvingas leta i manualer, fråga kollegor et c. Krångliga och svårförstådda system leder också till missnöjda användare med minskad arbetsglädje som följd.

Användbarhet tillmäts så stor betydelse att en speciell ISO-standard, ISO 9241, för programs användbarhet har definierats [ISO91, ISO92].

Att *vara medveten* om användbarhetens betydelse är en sak, att *veta hur* man uppnår hög användbarhet är en helt annan sak. Det senare är ett problem som de flesta utvecklingsorganisationer brottas med idag.

Inom forskningsområdet MDI (Människa-Dator-Interaktion) har forskarna länge studerat användbarhetsfrågor. För den som inte är bekant med forskningsområdet MDI, är det dock svårt att hitta rätt bland alla rön, begrepp och tankegångar. Mycket av forskningsresultaten har ännu ej heller stöpts i en sådan form att de direkt kan tillämpas vid praktisk systemutveckling. Dessutom har fokus ofta legat på att identifiera brister i användbarhet snarare än att erbjuda hjälp att öka den.

Syftet med detta dokument är att kortfattat beskriva en praktisk metod för användbarhetsdriven prototyputveckling. Metoden beskriver ett tillvägagångsätt för att utveckla prototyper av datorstöd. Den kännetecknas av täta kontakter mellan utvecklare och användare och är tämligen informell vad gäller krav på till exempel dokumentutformning. Metoden har influerats av internationell standardisering [ISO91, ISO92] samt metoder för användbarhetsutvärderingar [Monk93], men kanske allra mest av synsättet Usability Engineering (ung. användbarhetsdriven systemutveckling) som bland annat förts fram av Whiteside [Whiteside88] och Nielsen [Nielsen92, Nielsen93].

Den ledande tanken bakom metoden är att utvecklare inte ska försöka gissa sig fram till användarnas krav på datorstödet utan istället skaffa sig kunskap om användarnas situation, deras arbetsuppgifter, erfarenheter och arbetsmiljö och låta detta styra utvecklingsarbetet. Det innebär dock *inte* att utvecklaren okritiskt ska realisera alla användarens krav. Utvecklaren bör vara den som med hjälp av erfarenhet och den förvärvade kunskapen samt dessutom egen kreativitet utformar förslag till prototypens design. Denna design verifieras sedan genom utvärdering med användarna.

Metoden är utformad för att kunna genomföras på kort tid, idealt 2-3 veckor med 3-4 personer exklusive användarrepresentanter. Det innebär bland annat mängden funktionalitet i prototypen blir starkt begränsad. Detta är helt i sin ordning, huvudsaken är att prototypen är tillräckligt utvecklad för att tillåta utvärdering av dess utformning. Vår erfarenhet av moderna utvecklingsverktyg är också att förhållandevis mycket kan åstadkommas på en begränsad tid.

Metoden tillämpas med fördel i analysfasen av ett systemutvecklingsprojekt\* där den kan komplettera de abstrakta beskrivningar av verksamheten i form av t ex informations- och affärsprocessbeskrivningar som ofta tas fram där. Även om användare medverkat i framtagningen av de abstrakta modellerna kan vanligen en prototyp starkt bidra till ökad förståelse för de praktiska konsekvenserna av modellerna för användarnas arbete. Detta gäller speciellt om analysen av verksamheten gett att den ska förändras,

---

\* Ursprungligen har metoden utvecklats tillsammans med Telia Data för att infogas i analysmetoden IE-Start.

t ex att arbetet ska genomföras på ett nytt mer effektivt sätt. Det kan då vara svårt att föreställa sig konsekvensen för det dagliga arbetet. På så vis kan prototypen dels bidra till att stämma av specifikationerna och förankra dessa hos användarna, dels ge rekommendationer till det framtida utvecklingsprojektet.

Även om metoden som den beskrivs här är avsedd att användas i analysfasen så är principerna som beskrivs generella för design av gränssnitt och metoden kan med vissa tillägg även användas för prototyputformning av det slutliga användargränssnittet.

Metoden består av sju steg:

- Besök användarna på deras arbetsplats.
- Skapa uppgiftsbeskrivning
- Fastställ användbarhetsmål
- Prototypspecifikation
- Scenarioanalys
- Realisering av prototyp
- Utvärdering

Det är viktigt att förstå att denna metod är en iterativ process. De olika delaktiviteterna är beskrivna i en sekvens men i praktiken kommer man att behöva gå tillbaka och revidera olika steg beroende på att ny information kommer fram. Detta blir speciellt tydligt så fort prototypen börjar konkretiseras, antingen på papper eller i datoriserad form. För användarna blir då systemet mycket tydligare och de kan då ge bättre specifikationer av sina krav och önskemål på funktioner. Det kan även leda till att tidigare steg i systemutvecklingskedjan kan behöva revideras, till exempel affärsprocessbeskrivningar.

Varje steg, eller aktivitet, i metodkedjan beskrivs enligt följande mall:

- *Syfte*. Under denna rubrik beskriver vi varför aktiviteten utförs.
- *Problem*. Under denna rubrik beskriver vi våra egna erfarenheter av vilka praktiska problem som kan uppstå i samband med aktiviteten.
- *Arbetsresultat*. Här beskrivs kortfattat vilka resultat aktiviteten producerar.
- *Arbetsuppgifter*. Här beskrivs vilka arbetsuppgifter som ska genomföras och hur de ska genomföras.

## 2. Besök användarna på deras arbetsplats.

### Syfte

Lär känna användaren och skapa dig en bild av användarens nuvarande situation och förväntningar på det kommande systemet.

### Problem

Utvecklarorganisationen vill ej att användarna ska få personlig kontakt med utvecklarna. Risken finns att de då hellre kontaktar utvecklarna än supportorganisationen om fel skulle uppstå.

Säljorganisationen vill inte att utvecklare har direktkontakt med kunden.

Användarorganisationen vill ej avsätta tid för kontakter med utvecklare.

Svårt att göra uppgiftsbeskrivning om användarna inte är införstådda med eventuella beslutade förändringar i arbetssättet.

### Arbetsresultat

Intervjudokumentation

Användarbeskrivning

### Arbetsuppgifter

Delaktiviteten genomförs för att skapa förståelse hos projektgruppen för användarnas bakgrund, arbetssituation och tidigare erfarenheter.

#### Intervjuer

Genomför intervjuer med en eller helst flera användare för att samla in information om användarna, arbetsuppgifter, arbetsmiljö, organisation etc.

Be användaren beskriva i egna ord hur han arbetar idag. Be om förtydligande om det är begrepp som du själv inte förstår.

Be användaren visa de system som han arbetar med idag. Fråga vad i de gamla systemen han är nöjd respektive missnöjd med.

Det är i allmänhet nödvändigt att göra flera besök på användarens arbetsplats. Därför är det bättre att ha flera korta besök än ett långt.

#### Användarbeskrivning

Gör en beskrivning av den totala användargruppen. Speciellt bör följande punkter återfinnas i beskrivningen:

- Antal användare
- Åldersfördelning
- Utbildningsnivå
- Befattningar i nuvarande organisation
- Datorvana
- Verksamhetskunskap
- Arbetsplatsens utformning
- Övriga kännetecken

Om användargruppen är inbördes heterogen (vilket vanligen är fallet) är det viktigt att försöka identifiera de viktigaste olika typerna av användare.

## 3. Skapa uppgiftsbeskrivning.

### Syfte

Syftet med aktiviteten är att upprätta en av datorstöd oberoende beskrivning av användarnas arbetsuppgifter idag och eventuellt användarnas tänkta arbetsuppgifter om affärsprocesserna ska förändras. Uppgiftsbeskrivningen ska konkretisera och förtydliga de tidigare utförda beskrivningarna och bidra till att öka projektgruppens förståelse för problemområdet samt tjäna som en kontrollpunkt för specifikationerna av arbetsuppgifterna.

### Problem

Denna aktivitet är kritisk för det fortsatta arbete med prototyputförande och aktiviteten kan lätt bli den mest tidsödande. Det händer ofta att projektgruppen kommer in i diskussioner om hur användarna kan tänkas göra olika moment. Sådana diskussioner kan bli långa och tidsödande utan att man kommer fram till något resultat. Därför är det mycket bra om en användare deltar eller åtminstone finns tillgänglig per telefon för frågor.

Om de uppgiftsbeskrivningar som tidigare gjorts i analysen ej stämmer överens med den uppgiftsbeskrivningen som görs här bör problemet återremitteras för vidare analys.

### Arbetsresultat

Uppgiftsbeskrivning

### Arbetsuppgifter

#### Uppgiftsbeskrivning

Uppgiftsbeskrivningen genomförs med hjälp av intervjuer med användare som fås att beskriva sina arbetsuppgifter idag samt, om så är fallet, hur de uppfattar att de nya affärsprocesserna kommer att påverka deras arbete.

Beskrivning av och därmed förståelse för användares arbetsuppgifter är centralt för det fortsatta prototyputförningsarbetet. Om prototyputförningen görs av personer ur projektgruppen som tidigare genomfört de detaljerade beskrivningarna av aktiviteter kan denna delaktivitet mer ses som en kontrollpunkt, som en avstämning av det tidigare resultatet inför prototyputförningen. Om däremot nya personer kommer in i det här skedet, är det av största vikt att uppgiftsbeskrivningen utförs grundligt.

De arbetsuppgifter som beskrivs i uppgiftsbeskrivningen ska ur användarens perspektiv vara rimliga och logiskt sammanhängande.

Uppgiftsbeskrivningen bör innehålla beskrivningar av:

- Arbetssätt inklusive metoder och algoritmer som används.
- "Verkligheten" runt om kring, t ex arbetsmiljö, kontakter med kollegor eller kunder.
- Tid och resurskrav.
- Arbetets omfattning.

Uppgiftsbeskrivningen ska fånga upp i vilken sekvens olika uppgifter görs, om flera uppgifter görs parallellt, om man hoppar mellan flera olika uppgifter et c.

Var speciellt uppmärksam på om det verkar som om olika användare tänker sig skilda arbetsätt. Detta kan vara en indikation på att affärsprocessanalysen inte är tillräckligt genomarbetad eller förankrad. Om du uppfattar att användarna trots upprepade diskussioner fortfarande tycks oense om arbetets utförande och att du inte ser hur de olika arbetsätten kan stödjas av en och samma prototyp ska du remittera affärsprocessbeskrivningen tillbaka för vidare analys tillsammans med dina iakttagelser.

## 4. Fastställ användbarhetsmål

### Syfte

Användbarhetsmålen utnyttjas som ett sätt att stämma av användarnas/beställarnas förväntningar om det planerade systemets användbarhet.

### Problem

Utan analys av kostnader för att uppnå ett visst mål kan en orealistisk målbild byggas upp.

### Arbetsresultat

Användbarhetspecifikation i tabell. Slutsatser och konsekvenser för utformningen av systemet.

Rekommendationer till prototyputformning.

### Arbetsuppgifter

#### Utforma användbarhetspecifikation

Görs i samarbete med användare. Definiera användbarhetsmål avseende:

- Lätlärdhet
- Effektivitet
- Lätt att komma ihåg
- Misstagsfrekvens (felfrekvens)
- Omdöme

Gradera målen i: Idealt, Acceptabelt, Helt oacceptabelt.

Definiera mätmetod för varje mål. Nedan finns ett exempel på en användbarhetstabell:

| Egenskap/<br>värden                                | Idealt        | Acceptabelt   | Helt<br>oacceptabelt | Mätmetod  |
|--|---------------|---------------|----------------------|---|
| Lätlärdhet   | 2 dagars kurs | 3 dagars kurs | > 5 dagars kurs      | Test efter kursen. 8 av 10 signifikanta uppgifter rätt räknas som att kunna systemet    |
| Effektivitet<br>(tid att klara registreringen)     | < 1 timme     | 1-2 timmar    | > 2 timmar           | Tidstudie efter 2 månaders användning   |
| Lätt att komma ihåg<br>(efter 5 månaders uppehåll) | 0 timmar      | 2 timmar      | gå kursen igen       | Test efter 5 månader. 8 av 10 signifikanta uppgifter rätt räknas som att kunna systemet |
| Felfrekvens  |               |               |                      |   |

|                 |   |   |  |                            |
|-----------------|---|---|--|----------------------------|
| - hjälpfunktion | <1 ggr/dag  | 1-2 ggr/dag<br>10 min per fel                             | > 5 ggr / dag  | loggning av alla användare |
| - Annan kollega | < 1 ggr/vecka   | 1 gång/vecka  | > 3 ggr / vecka  | enkät                      |
| - Help Desk     | < 1 ggr / varannan vecka  | 1 gång varannan vecka                                     | > 2 ggr per vecka  | enkät                      |
| Omdöme          | 75 procent upplever systemet som likvärdigt med Microsofts produkter, t ex Excel och Word | 75 procent upplever "äntligen har vi fått ett bra system" | en majoritet upplever det som att "det blev samma gamla vanliga visa igen" | enkät                      |

Det är viktigt att användbarhetsmålen stäms av med flera användare så att de speglar en grupp användares förväntningar snarare än en specifik användare.

#### **Kostnads/intäktsanalys**

Sätt pris på realisering av användbarhetsmålen

#### **Eventuell omformulering av användbarhetsspecifikationen**

Utvärdera om kostnaderna för att uppnå användbarhetsmålen är rimliga. Om ej, omformulera dem och gör ny kostnads/intäktsanalys.

#### **Slutsatsdragning**

På grundval av användar- och uppgiftsbeskrivningarna samt användbarhetsmålen dras slutsatser om de allmänna kraven på systemet samt krav specifika för användargränssnittets utformning. Hur användbarhetsmål ska översättas till konkreta slutsatser om gränssnittsutformning är svårt att ge några generella regler för.

#### **Rekommendationer till prototyputformning**

Konkretisera information från användar- och uppgiftsbeskrivningar, användbarhetsanalys och annan information från intervjuer med användarna till specifika rekommendationer om hur prototypen bör utformas.



# 5. Prototypspecifikation

## Syfte

Syftet med aktiviteten är att tidigt få återkoppling från användarna om prototypens design.

## Problem

Intervjuer med användarna har inte gett tillräcklig förståelse för hur deras arbetsuppgifter ska komma att utföras vilket kan leda till allt för långa diskussioner i projektgruppen om hur användarna egentligen gör.

## Arbetsresultat

Prototypspecifikation

## Arbetsuppgifter

I denna arbetsuppgift är det nödvändigt att man har tillgång till åtminstone någon användare som man kan ringa till eller besöka om förtydliganden angående arbetsuppgifter behövs. Nya besök hos användarna leder förmodligen till att resultaten från tidigare steg måste revideras.

### Avgör om prototypen ska vara vertikal eller horisontell

En horisontell prototyp är grund men bred, dvs den är en prototyp över hela systemet sett ur användarnas perspektiv. En horisontell prototyp väljer man om det viktigaste är att visa på de övergripande egenskaperna hos ett system.

En vertikal prototyp är smal men djup, dvs den innehåller många funktioner för ett begränsat område. En vertikal prototyp väljer man om det är något deluppgift i användarens arbete som är speciellt kritisk eller oklar.

### Metaforanalys

Med en metafor menas en liknelse med någon känd verklighet. Ett typiskt exempel är skrivbordsmetaforen för Macintosh och Windows där användaren upplever systemet som ett elektroniskt skrivbord. Bra utformade metaforer underlättar användarens inlärning och förståelse av systemet.

Ur uppgiftsbeskrivningen ska framgå vilka begrepp som är centrala i användarens uppgift. Ett bra knep är att basera metaforen på dessa. Om t ex användarna säger att de arbetar med kunder och order ska det vara samma begrepp, Kund och Order, som dyker upp som objekt i gränssnittet och inte abstrakta begrepp såsom "dokument", "fil", "poster" och "fält".

### Beskrivning av användningsmodellen

Med användningsmodell menas det övergripande arbetsättet användarna tillämpar i systemet. Användningsmodellen beskriver i vilken ordning olika uppgifter ska utföras. Användningsmodellen ska beskriva det bästa sättet att utföra en uppgift på. Notera dock att systemet ändå så långt det är möjligt ska tillåta alternativa arbetssätt. Vi människor är och tänker olika vilket ett datorsystem måste ta hänsyn till.

### Övergripande prototyputformning

När du har en första metafor och användningsmodell kan du göra en första skiss på hur användargränssnittet ska se. Gör det på papper eller white-board som du fotograferar av. Det är ingen idé att börja med en datorprototyp i det här läget.

Visa och gå igenom din prototypskiss med olika användare. Fråga efter alternativa lösningar på olika moment. Var lyhörd för deras synpunkter och åsikter även om de

verkar konstiga. Tänk på att det är användarnas framtida arbetsmiljö du håller på att utforma.

Revidera din prototypskiss efter dina samtal med användare. Var uppmärksam på om det verkar som om någon del av din skiss var oklar för användarna och tänk efter vad du kan göra för att förtydliga den i sådana fall.

Notera att det i detta skede inte är meningsfullt att diskutera det exakta utseende på skärmen, dvs vilka knappar som ska finnas, hur menyer ska struktureras, färgval et c. Här handlar om att fånga de övergripande designkraven.

# 6. Scenarioanalys

## Syfte

Ett scenario är en konkretisering av en specifik användningsituation med den tidigare specificerade prototypen. Scenarioanalysen utgör underlag för prototyputveckling och test.

## Problem

Prototypspecifikationen är orealistisk eller stämmer inte överens med användarnas arbete.

## Arbetsresultat

Scenariobeskrivningar

## Arbetsuppgifter

### Utforma tre scenarier

Utforma tre olika scenarier som innehåller beskrivningar av:

- en användare,
- de funktioner han använder,
- användarens mål med systemanvändningen,
- de omständigheter han arbetar under,
- det tidsintervall som scenariot utspelas under,
- samt själva händelseförloppet.

Av de tre scenarierna ska ett spegla en normal användningssituation, ett spegla en sällsynt men viktig uppgift, samt ett spegla återhämtning från en felsituation eller problem.

Utformningen gör du tillsammans med en användare. Var detaljerad i scenariot. Var realistisk när scenariot beskrivs. Det är t ex inte normalt att man kan läsa igenom tio dokument och utföra fem Excel-kalkyler samt fatta ett viktigt beslut på tio minuter. Det är inte heller troligt att man kan arbeta en hel dag utan avbrott, t ex att telefonen ringer.

# 7. Realisering av prototyp

## Syfte

Realisera prototyp enligt specifikation i syfte att prototypen ska kunna utvärderas.

## Problem

Det verktyg man valt för prototyputveckling är för tungrott och man når inte fram till avsett resultat inom given tid.

## Arbetsresultat

Prototyp.

Prototypbeskrivning

## Arbetsuppgifter

### Utveckla prototyp

Prototypen utvecklas lämpligen med något av de prototypverktyg som finns på marknaden. Det är normalt sett inte lämpligt att välja en fullfjädrad programmeringsmiljö som C för denna fas.

Prototypen utvecklas iterativt med kortare avstämningar med användare. Dessa ska dock inte delta i arbetet utan konsulteras vid behov då designgruppen behöver mer input.

För att klara tidskraven blir man i allmänhet tvungen att göra avkall på prototypens generella funktionalitet och istället "hårdkoda" vissa delar vilket är fullt acceptabelt.

### Beskriv prototyp

Prototypen beskrivs kortfattat med hjälp av skärmbilder och beskrivning av hur en tänkt användare arbetar med systemet.

## 8. Utvärdering

### Syfte

Syftet med utvärderingen är att avgöra om den utvecklade prototypen är realistisk och överensstämmer med användarnas förväntningar.

Syftet är också att identifiera eventuella användbarhetsproblem för att kunna ge en bättre kravspecifikation. Det är värt att notera att en utvärdering av den här typen inte syftar till att göra en uttömmande granskning av samtliga användningsproblem, utan snarare är ett snabbt och effektivt sätt att identifiera några viktiga problem.

### Problem

Prototypen är ej tillräckligt komplett (ej heltäckande, innehåller "buggar" osv) för att användarna ska kunna prova den själva under realistiska förhållanden (t ex att scenarierna ej kan användas).

För få försökspersoner för att kunna dra säkra slutsatser.

### Arbetsresultat

Utvärderingsrapport.

### Arbetsuppgifter

Prototypen utvärderas av användare och designgruppen tillsammans [Monk93]. Användarna som agerar försökspersoner får genomföra representativa uppgifter som baseras på de tidigare framtagna scenarierna under observation av försöksledare som också ofta är utvecklare i designgruppen.

Vikten av en dialog mellan försöksledare och försökspersonerna poängteras. Det handlar alltså inte om att försöksledaren bara passivt observerar försökspersonernas användning av prototypen, utan också om att skapa en diskussion med dessa om hur de uppfattar och tolkar prototypens signaler och återkoppling på deras handlingar när de försöker lösa försöksuppgifterna. Diskussionen ger även mycket information om hur försökspersonerna uppfattar prototypens anpassning till deras arbetsuppgift.

Resultatet av utvärderingen är kvalitativt, dvs säger något om den provade prototypen i kvalitativa termer som "Detta är inte bra" eller "Orderfördelningen är svår att överblicka". Detta som en motsats till andra metoder med kvantitativt resultat där man får mer konkreta svar på frågor som "Hur många?" eller "Hur ofta?". Den här beskrivna utvärderingsmetoden är också en sk formande metod, dvs syftet är att nå ett resultat som kan användas för att förbättra (forma) systemet.

Orsaken till att en formande, kvalitativt inriktad, metod används är att vi här utvärderar prototyper i ett tidigt stadiet av utveckling. Summerande, kvantitativt inriktade metoder lämpar sig bättre för kvalitetsuppföljning och efterkontroll av det kompletta systemet.

#### Val av försökspersoner

Vid rekrytering av försökspersoner utgår man från den användarbeskrivning som gjorts i steg 1. För att få en så god bild som möjligt av användningen bör försökspersonerna spegla hela användargruppen vad gäller tidigare datorerfarenhet, verksamhetskunskap, utbildningsnivå osv. Antalet försökspersoner bestäms ofta av praktiska begränsningar som tillgänglig tid och antal tillgängliga användare. Sex till åtta försökspersoner anses i praktiken vara ett lagom antal, därefter börjar mängden nya observationer avta.

#### Konstruktion av försöksuppgifter

Utvärderingen bygger på att ett antal uppgifter genomförs av försökspersonerna. Dessa uppgifter kan formuleras utifrån de tidigare gjorda uppgifts- och scenariobeskrivning-

arna och bör uttryckas i konkreta ordalag som t ex "Registrera följande beställning". Formuleringen "Gör ditt dagliga arbete" blir för vag.

Vid utformningen av uppgifter måste man ta hänsyn till hur långt prototypen som ska utvärderas har utvecklats. Tidiga prototyper tillåter ofta inte stora sammansatta uppgifter utan de blir med nödvändighet enklare och mer begränsade. Vidare måste uppgifterna vara möjliga att genomföra under den tid som finns till förfogande för varje försöksperson. Om det är ett stort system med mycket att lära för varje försöksperson är det bättre att använda två användare i tio timmar (uppdelat på flera tillfällen) än fyra användare i fem timmar.

Innan försöket startar är det viktigt att kontrollera att allt fungerar som planerat. Exempelvis bör man arbeta igenom försöksuppgifterna för att försöka uppskatta om de är möjliga att genomföra på given tid. Detta måste också stämmas av mot antalet försökspersoner.

### **Genomförande**

Försöket genomförs på så vis att försökspersonen arbetar igenom försöksuppgifterna i dialog med försöksledaren som också observerar eventuellt tillsammans med en ytterligare observatör. Det kan vara lämpligt att vara två observatörer eftersom försöksledaren ofta blir inbegripen i en diskussion med försökspersonen vilket hindrar försöksledaren från att göra anteckningar. Försöksledaren bör dessutom prioritera diskussionerna med försökspersonen före anteckningarna.

Som komplement till observatörernas anteckningar är det lämpligt att ha någon form av automatisk loggning, t ex en bandspelare (med räkneverk) eller ännu hellre en videokamera för ljud och bildupptagning. Kameran bör då vara riktad mot skärmen, det är sällan möjligt att få med både användare, tangentbord, och skärm i samma bild.

För anteckningar och ljudband eller videoband behövs en gemensam klocka. T ex kan man lägga datorns klocka på bildskärmen eller också lägga in tiden i videobilden om den funktionen finns tillgänglig.

När så försökspersonerna till slut anländer är det mycket viktigt att få dem att känna sig trygga och tillfreds med situationen. Det är mycket viktigt att försökspersonerna inte får uppfattningen att det är de och deras sätt att handskas med prototypen som utvärderas, utan att det är prototypen som är föremålet för utvärderingen.

Beroende på försökspersonernas tidigare vana av liknande program eller tidigare datorerfarenhet fortsätter man sedan med en kort introduktion till metoden samt en mer eller mindre detaljerad introduktion till programmet.

Det är viktigt att försökspersonen tänker högt och berättar om vad han/hon ska göra, hur han reagerar osv, när han använder prototypen. Om försökspersonen tystnar och försjunkar i arbetet måste försöksledaren hålla dialogen igång genom att ställa frågor som t ex "Vad vill du göra nu?" "Hur uppfattar du programmets meddelande här", "Vad har programmet gjort nu?". Det är viktigt att inte försöksledaren låter kritisk, det är kanske inte så bra att fråga "Varför gjorde du så?!" utan bättre med "Vad trodde du programmet skulle göra när du gjorde det där?".

En variant är att under ett försök ha två försökspersoner som samtidigt provar prototypen. Detta ger både för- och nackdelar: Det blir enkelt att få reda på försökspersonernas uppfattning om prototypen eftersom de hela tiden diskuterar försöksuppgifternas lösning och prototypens egenskaper sinsemellan. Detta är en fördel eftersom det annars kan vara svårt att få försökspersonen att spontant tänka högt. En nackdel är att två försökspersoner vanligen löser försöksuppgifterna snabbare än vad en person gör vilket gör det svårt att bedöma uppgifternas svårighetsgrad i ett verkligt läge. Om två samtidiga försökspersoner används bör man också se till att de kommer ur liknande användarkategorier.

Observatörerna ska ge akt på alla kommentarer från försökspersonerna såväl som på allt oväntat beteende, dvs beteenden som inte hade förutsetts av designgruppen. Dessutom bör observatörerna noga iakttä försökspersonerna för se när dessa verkar osäkra, förvånade osv. Allt detta kan avslöja eventuella användningsproblem som sedan kan mejslas fram och konkretiseras under diskussion med försökspersonen.

Försökspersonen måste tillåtas att begå misstag, hitta problem och ställa frågor. Det är precis denna typ av information som vi är intresserade av vilket innebär att försöksledaren inte ska förekomma försökspersonerna och hjälpa dem på traven, även om det tydligt syns att de håller på att begå ett misstag. Försöksledaren kan mycket väl svara på frågor från användaren men bör först fundera över om han först vill ha reda på försökspersonens uppfattning innan han ger det korrekta svaret. Försöksledaren får också göra bedömningen om frågans relevans för utvärderingsresultatet.

### **Analys**

Utvärderingsresultatet sammanfattas sedan i en rapport där följande bör ingå:

- En korta bakgrund till utvärderingen inklusive beskrivning av testuppgifterna, försökspersonerna samt försöketsförutsättningar.
- Prototypens anpassning till arbetsuppgiften
- Övriga observerade användningsproblem och saknade funktioner
- Prototypens överensstämmelse med användarnas förväntningar
- Slutsatser och rekommendationer

Kännedom om de förutsättningar under vilket användbarhetstestet genomfördes är viktigt eftersom en korrekt tolkning av användbarhetstestet resultat annars inte kan göras. Det är därför viktigt att dessa förutsättningar beskrivs i rapporten för att personer utanför deltagargruppen ska kunna tolka resultatet av utvärderingen.

Ett problem med större försök som kanske dessutom videofilmats, är att mängden material att analysera snabbt blir mycket stor. Olika uppskattningar av förhållandet mellan försökstid och utvärderingstid menar att utvärderingstiden är mellan tre till fem gånger längre än försökstiden. Det är ofta bäst att använda observatörernas anteckningar som huvudkälla i analysen medan en videofilm kan användas som referens om man vill kontrollera något i efterhand.

## 9. Referenser

- [Nielsen92] J. Nielsen, "The Usability Engineering Life Cycle", IEEE Computer, Mars 1992.
- [Nielsen93] J. Nielsen, "Usability Engineering", Academic Press Ltd, 1993.
- [ISO91] ISO9241, First Committee Draft, del 10, "Dialogue Principles", 1991.
- [ISO92] ISO9241, Second Committee Draft, del 11, "Usability Principles", 1992.
- [Whiteside88] J. Whiteside, J. Bennet, K. Holtzblatt, "Usability Engineering: Our Experience and Evolution", ur Handbook of Human-Computer Interaction, sid. 791ff, Elsevier Science Publishers B.V. (North-Holland), 1988.



# SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING

SISU

Electrum 212, 164 40 Kista  
Isafjordsgatan 26  
Telefon 08-752 16 00 Telefax 08-752 68 00