

# Uppgiftsanalys och användbarhetskrav

## Del 1 Kravformulering

Av Stefan Blomkvist

**Vem är användare av systemet och varför använder personen systemet? Hur och var används systemet? Det är frågor som måste besvaras när ett datorsystem ska utvecklas.**

**Och de frågorna kan en utvecklare inte besvara själv. Att förstå användaren, är att förstå att du inte är användaren. Vi måste studera riktiga användare för att få kunskap.**

### Inledning

När datorsystem utvecklas behövs kunskap om vilka användarna är, deras mål och vilka uppgifter de vill utföra. Att ta fram denna kunskap, går under benämningen användaranalys respektive uppgiftsanalys. De är viktiga steg i att kunna formulera mål och krav för det datorstöd som ska tas fram.

Denna fas i systemutvecklingen går under lite olika benämningar, men vanligast är *kravanalys* eller *kravformulering* (requirements analysis / gathering).

### Kravanalys

Kravanalysen utförs i början av systemutvecklingsprocessen för att tidigt kunna klargöra användarnas behov och identifiera krav, även sådana som är otydliga, förbisedda och omöjliga. Kunskapen formuleras med krav på systemet i en kravspecifikation (requirements specification).

I traditionell systemutveckling ligger fokus på funktionella och tekniska krav. I en användarcentrerad utvecklingsprocess är användbarhetskrav viktigare. Om inte en viss funktion kan användas på ett bra sätt, är den ändå mer eller mindre värdelös, eller kan t.o.m. göra arbetet för användarna ineffektivt eller farligt. I bland delas kraven upp i funktionella och icke-funktionella krav. Krav på användbarhet återfinns då i den senare kategorin.

En viktig insikt är att vi inte kommer att kunna formulera alla relevanta krav första gången. Det beror på att problemen som ska lösas med systemet tillsammans med användare och organisation tenderar att bli komplexa. Och vi har inte all kunskap om hur denna komplexitet fungerar.

Vi börjar med en begränsad kunskap om problemen som ska lösas och de krav som finns. Efterhand lär vi oss mer om problemen och kan utforska hur de ska lösas - ett iterativt arbetssätt. Iterativ utveckling är att förfinas efter hand och upprepa samma moment flera gånger, med bättre kunskap och resultat varje gång.

### Kunskap om användare

Det är viktigt att begripa att systemutvecklare, programmerare, gränssnittsdesigners, verksamhetsanalytiker och andra som är involverade i utveckling av systemet, inte själva är typiska användare. De kan inte heller ersätta riktiga användare som källor till kunskap. Att förstå användaren, är att förstå att du inte är användaren.

Hur kan vi samla in kunskap om användarna? Det bästa sättet är att studera, intervjuva och mer eller mindre leva med användarna i deras arbetsmiljö. Det ger nödvändiga insikter i arbetsförhållanden, skillnader mellan användare och andra förutsättningar.

Vi måste ut till användarnas arbetsplats för att lyckas fånga kunskap som inte går att greppa på annat sätt än genom att vara där. Det är först då vi har möjlighet att se informella organisationsstrukturer och ta del av "tyst kunskap".

Om användarna inte är tillgängliga av något skäl, t.ex. att vi ännu inte vet vilka de är, får man försöka samla delar av den nödvändiga kunskapen genom t.ex. marknadsundersökningar eller enkäter.

### Steg i kravanalysen

Ett problem med att utveckla användbara system är att veta om det färdiga systemet verkligen är användbart. Är målen uppfyllda eller inte?

För att kunna utvärdera systemets användbarhet måste vi ställa upp kriterier eller krav för systemet. Kraven måste också vara **mätbara** och **specificerade i förväg** för att vi ska kunna veta om målen uppfylls.

Systemet eller prototyper som tas fram under utvecklingen utvärderas enligt uppsatta användbarhetskrav. Det räcker inte med att bara säga att systemet ska vara "användarvänligt", det måste gå att visa att så är fallet.

Att klarlägga **användbarhetskrav** för systemet (usability requirements ), eller användbarhetsmål, innebär tre huvudtyper av kunskapsinsamling och analys. Det generella syftet med analyserna är att definiera vilka tjänster eller funktioner som systemet måste tillhandahålla. Målet med uppgiftsanalys är att organisera och ordna dessa tjänster för att stödja användarens arbete. Dessutom är vi intresserade av att förstå och beskriva själva användarna och den miljö de är verksamma i.

Följande steg ingår:

1. Användaranalysen ska svara på frågor om vilka användarna är och vilka egenskaper hos användarna som viktiga för det aktuella systemet.
2. Kontextuell analys fokuserar på den omgivning som användaren verkar eller kommer att verka i. Det inkluderar aspekter på den fysiska miljön likväl som arbetsorganisation.
3. Uppgiftsanalysen ska svara på frågor om vilka uppgifter användarna utför och hur dessa genomförs.

## Del 2 Användaranalys

**Användaranalysen skall svara på frågor om vilka användarna är och vilka egenskaper hos användarna som viktiga för det aktuella systemet. Det är också viktigt att undersöka den omgivning de verkar i, eftersom det kan ställa särskilda krav på systemet.**

### Vilka är användarna?

*En modern telefonväxel hos ett mindre företag är utrustad med en mängd tekniska finesser. Via sina telefoner kan personalen på företaget knappa in att de är på lunch eller på resa. De kan lyssna av telefonsvararen som finns i växeln och de kan förstås ringa på telefonerna.*

*I företagets entré finns receptionisten som sitter framför en skärm med programmet som styr växeln. Receptionisten kan se när alla i personalen har meddelat att de är på lunch och när de*

kommer tillbaka. Via programmet kan de se att någon är upptagen i telefon, och då ta meddelande personligen eller låta svarsfunktionen sköta det. Ringer man till någon på företaget som råkar vara ute, kan man välja att lämna ett meddelande eller trycka en knapp för att bli kopplad till receptionisten.

I bland är det uppenbart vilka användarna är, men ofta finns det grupper av användare som är svårare att identifiera. I exemplet med telefonväxeln, finns det två självklara grupper av användare:

- Receptionisterna
- Övrig personal som har en telefon

Dessa två kategorier behöver man naturligtvis studera och ta hänsyn till när systemets interaktion utformas. En något mindre uppenbar användarkategori är alla som ringer till företaget och möts av telefonsvararmeddelanden och andra funktioner i växel. Även dessa användare måste beaktas vid utvecklingen. Det går att tänka sig fler grupper av användare, som inte är beskrivna i exemplet. Det finns alldeles säkert personer som kommer att underhålla systemet i olika omfattning. Dels någon supporttekniker på företaget som installerar växelprogrammet i en dator, dels supporttekniker från växeltilverkaren som behöver arbeta med systemet när större fel inträffar.

I exemplet förstår vi att receptionisten har helt andra behov än övriga telefonanvändare. Men vilka behov är det och vilka egenskaper utmärker just receptionisterna?

## Användarkategorier

Användarna kan kategoriseras efter olika egenskaper. Vilka egenskaper som är viktiga att kartlägga beror på vilken typ av datorstöd som ska utvecklas. Några typiska egenskaper:

- Mål och uppgifter – användarnas övergripande mål med att använda systemet, personliga, praktiska, organisatoriska, vilka huvudsakliga arbetsuppgifter har användarna.
- Sammansättning – ålder, kön, antal
- Kognitiva och fysiska förutsättningar och begränsningar
- Utbildning och erfarenhet – bakgrund, kunskap om arbetsområdet, språk, datorvana och andra färdigheter
- Hjälpmedel och verktyg idag – support, dokumentation
- Användning av systemet – hur ofta, hur länge, obligatoriskt, valfritt
- Organisation – hur organiseras och fördelas arbetet inom organisationen
- Omgivning – i vilken miljö verkar användarna: utomhus/inomhus, på kontor, kontrollrum etc. Störande och speciella saker i omgivningen.

## Studera användare

Kunskap om användarna kan komma genom olika metoder, exempelvis:

- Enkäter
- Intervjuer
- Observationer på plats hos användare och i testmiljöer

Om möjligt är det alltid bäst att kunna möta och studera riktiga, levande användare i deras normala miljö. Men det går inte alltid, som t.ex. när helt nya konsumentprodukter utvecklas. Då är de kommande användarna fortfarande okända. I sådana fall används t.ex. marknadsundersökningar för att försöka kartlägga potentiella användare. Alternativt går det att hitta användare för liknande eller äldre produkter. Testmiljöer där verkligheten återskapas på bästa möjliga sätt får ersätta riktiga miljöer.

Oavsett metod syftar användaranalysen till att urskilja typer eller kategorier av användare med liknande bakgrund, förutsättningar, arbetsuppgifter och krav på användargränssnitt. Analysen kan resultera i:

- användarprofiler (användarkategorier/klasser)
- designrekommendationer, samt
- en del av underlag för en kravspekifikation.

---

## Exempel användarprofiler - Bokning av tvättider

Totalt finns ca 500 presumtiva användare. Bostadsrättsföreningen innehåller både unga, ensamstående, barnfamiljer och äldre. Användarna kan delas in i följande kategorier:

	<b>Yngre</b>	<b>Barnfamilj</b>	<b>Äldre</b>
<b>Ålder</b>	18-30	20-50	50-80
<b>Hushållstyp</b>	1-2 pers	2-5 pers	1-2 pers
<b>Kön</b>	Kvinnor/män	Kvinnor/män	Flest kvinnor
<b>Datorvana</b>	Stor	Mellan - stor	Liten
<b>Tvätterfarenhet</b>	Rel liten	Mellan - stor	Stor, fasta vanor
<b>Tvättvanor</b>	Små mängder oregelbundna tider sen kväll spontantvätt	Stora mängder ofta och regelbundet eftermiddag, tidig kväll, helg paniktvätt	Små mängder regelbundet dagtid fast tvättidag

## Språk

Svenska talas och förstås hjälpligt av i stort sett hela användargruppen. Det finns några personer med små kunskaper i svenska. För deras skull måste språket i gränssnittet vara mycket enkelt och kompletteras med symboler.

## Funktionshinder

Följande funktionshinder finns representerade; synskador/nedsett syn, hörselskador/nedsett hörsel, färgblindhet.

## Omsättning

Genomströmningen i husen är relativt låg och sker framförallt i gruppen unga - ensamstående/par.

## Hjälpmedel

Idag bokar man tvätttider på en tavla i tvättstugan. Tanken är att man ska kunna göra det över intranätet istället. Majoriteten av hushållen har tillgång till egen dator, Majoriteten av användarna har PC eller Mac - av varierande ålder, med varierande operativsystem och de vanligen förekommande webbläsarna. Några personer har inte tillgång till dator - för dem som inte har det måste bokningen göras tillgänglig på annat sätt. Vid bokning behöver man tillgång till en kalender.

## Yttre faktorer – regler

Bokning av tvätttider styrs idag av hur tvättavlan ser ut. Det finns 4 tvättmaskiner, 2 torktumlare och ett torkskåp. Dessa bokas i 2-timmarspass. Man har tillgång till torikutrustningen en timme efter bokad tid. Tvättstugan är öppen mellan 07.00 och 22.00. Eventuella problem hanteras av fastighetsskötaren.

## Användning

Bokar tvätttider gör man relativt sällan. Som mest någon eller några gånger i veckan. Eftersom man idag har tvättavla (varje hushåll har en tvättnyckel) bokar man oftast i samband med att man tvättar. Varje användningstillfälle varar högst någon minut. När det nya systemet införs tas den gamla tvättavlan bort.

## Utvecklare är inte användare

Det är viktigt att se skillnaden mellan utvecklare, verksamhetsexperten och slutanvändare. Utvecklarna kan aldrig anses vara användare. De är alltid präglade av sin roll som just utvecklare, även om de kan verksamheten i övrigt. Verksamhetsexperten kan anses vara användare i en roll som experter inom det område för vilket man utvecklar systemet. De är mycket viktiga för att få fram kunskap om tillämpningsområdet, men måste kompletteras med slutanvändare – de som i sitt dagliga arbete skall jobba med systemet. Det är egentligen endast dessa slutanvändare som kan tala om för oss, genom studier och utvärderingar om vi lyckats göra ett användbart system.

Mer läsning och exempel på användarkategorier: *Getting to know your users* (<http://www.it.bton.ac.uk/staff/lp22/CP303/userslecture.html>).

## Personas

Ett alternativt eller komplementär sätt till användarkategorier är att beskriva användare som en *persona*. Begreppet har tagits fram av Alan Cooper (1999) och är en del i en designfilosofi som han kallar för målstyrd design (goal-directed design).

En persona är en representativ modell av användare som fokuserar på individens mål och vad hon vill åstadkomma. När användare inte finns tillgängliga eller ännu inte finns i verkligheten, får istället personen representera användarna. Genom att designa för personen, ska produkten kunna bli användbar för de verkliga användarna, menar Cooper.

Persona-modellen påminner om vanliga användarprofiler, men med några viktiga skillnader. Det är en arketypisk representation av verkliga eller potentiella användare. Det är inte en beskrivning

av en individuell eller en *genomsnittlig* användare. En persona representerar mönster av användares beteenden, mål och motiv, sammanställd i en fiktiv beskrivning av en individuell användare. Den innehåller också uppiktade personliga detaljer för att göra personen mer levande och verklig för utvecklarna av systemet.

Denna typ av användarbeskrivning passar bäst när användarna är svåra att få tag på eller är delvis okända (t.ex. när en helt ny produkt ska tas fram). Coopers idéer är också inriktade på tekniska produkter, inte på datorstöd som utvecklas inom en organisation.

## Del 3 Uppgiftsanalys

**Uppgiftsanalysen ska svara på frågor om vilka uppgifter användarna utför och hur dessa genomförs.**

### Vad gör användarna?

När det gäller att utveckla datorstöd för arbetslivet handlar det om arbetsuppgifter. I andra sammanhang utför också användarna olika uppgifter, t.ex. att skicka sms med mobilen.

Frågor som uppgiftsanalysen ska svara på är av typen:

1. Varför utför användaren en viss uppgift? Mål.
2. Hur ofta utförs uppgiften? Frekvens.
3. Hur lång tid tar det att utföra uppgiften? Tid.
4. Var utförs uppgiften, i vilken omgivning används systemet? Omgivning/miljö.
5. Vilka steg eller handgrepp behövs för att utföra uppgiften? Deluppgifter.
6. Samarbetar användaren med någon annan när uppgiften utförs? Organisation.
7. Vilka hjälpmedel; datorsystem, produkter, blanketter m.m., behöver användaren för att utföra arbetsuppgiften?
8. Finns det kritiska uppgifter och ”flaskhalsar” som gör uppgiften svår?
9. Hur kan situationen och informationsstödet förbättras?

För att ta fram ett användbart system är det viktigt att göra en noggrann uppgiftsanalys. Med den som underlag går det att få en bild av vilka arbetsuppgifter eller funktioner användarna ska kunna utföra med hjälp av systemet. Om inte användarna kan utföra sina uppgifter med systemet, så kommer det ändå inte att användas. Ett vanligt problem är att systemen innehåller allt för många funktioner – med följd att det är svårt att hitta de som är relevanta. I de fallen har inte utvecklarna förstått vilka uppgifter som är viktiga och hur användarna bäst utför dem.

### Studera användare

Uppgiftsanalysen kan genomföras med olika metoder. De vanligaste är strukturerade intervjuer och observationsintervjuer. Bra att tänka på vid intervjuer kan vara:

- Fråga om specifika och konkreta situationer och produkter av användarnas arbete.
- Undvik abstrakta frågeställningar om hur de tror deras arbetssituation kan förbättras och ändras.

Komplettera med observationer av användare i arbete.

Exempel på frågeställningar vid intervjuer:

- "Varför gör du det?" – relatera till mål.
- "Hur gör du det?" – dela in i deluppgifter.
- "Varför gör du inte på ett annat sätt?"
- "Blir det ibland fel när du gör det här?"
- "Vad händer om det blir fel?"

## Metoder för uppgiftsanlys

Resultatet från intervjuerna och observationerna, sammanställs och struktureras. Här finns det olika metoder att använda sig av – från makrometoder, där hela systemet i termer av organisatoriska, sociala och tekniska aspekter beaktas, till mikrometoder, där enskilda uppgifter delas upp i mindre kognitiva enheter, t.ex. en knapptryckning.

Allmänt innebär en analys av uppgifter följande:

- Formulera alltid ett övergripande mål med hela interaktionen.
- Formulera användarnas mål och delmål.
- Formulera sätt att uppnå målen.
- Formulera vilken information som behövs för att uppnå målen.
- Gör uppdelningen "bredden först" – identifiera gemensamma delmetoder.
- Stanna uppdelningen då "löven" i form av arbetsmoment nåts.
- Arbeta gärna med papper och penna t.ex. post-it notisar.

Resultatet blir ett material som tillsammans med användaranalysen bildar en bra utgångspunkt och underlag för designprocessen.

I kursboken (Preece, 1995) beskrivs två olika angreppssätt för att utföra en uppgiftsanlys: hierarkisk uppgiftsanlys (HTA) och kognitiv uppgiftsanlys. Här kommer vi att beskriva HTA översiktligt och dessutom några andra alternativa synsätt på uppgiftsanlys.

## Hierarkisk uppgiftsanlys – HTA

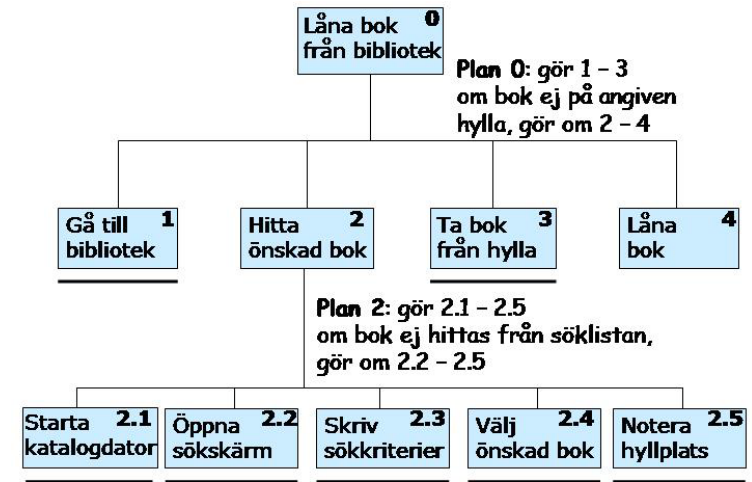
Hierarkier är viktiga strukturer ur ett kognitivt perspektiv. Vår problemlösning är ofta hierarkiskt ordnad med uppgifter som kan delas upp i mindre uppgifter, för att slutligen nå en mer eller mindre odelbar handling, som att trycka på en knapp. Målet med hierarkisk uppgiftsanlys (*hierarchical task analysis*, HTA) är att beskriva användarnas uppgifter i en hierarki av mål, uppgifter, operationer och planer.

- Mål (goals) – önskvärt tillståndet i systemet.
- Uppgifter (tasks) – hur målet kan uppnås
- Operationer eller handlingar (actions) – vad som görs för att utföra uppgiften, den minsta beskrivningsnivå av användarens handlingar
- Hierarkier av uppgifter och deluppgifter
  - Plan – beskriver under vilka villkor som en...
  - ...deluppgift (subtask) ska utföras
  - Stoppkriterier – hur långt en uppgift ska delas upp i deluppgifter.

Ett exempel på HTA (från Preece, Rogers & Sharp, 2002, *Interaction Design*):

Tänk dig in i att du vill låna en bestämd bok på biblioteket, det är ditt mål. För att uppnå målet behöver du utföra ett antal uppgifter, som att gå till biblioteket, hitta önskad bok, låna o.s.v. Hitta önskad bok kan i sin tur kan delas upp i aktivera datorn med bibliotekskatalogen, sök bok etc.

Figuren visar hur uppgiften kan struktureras hierarkiskt med HTA. Planerna anger i vilken ordning uppgifterna görs och under vilka villkor. Uppgifter som inte delas upp ytterligare, markeras med en streck under boxen. Deluppgift 4, låna bok kan däremot delas upp ytterligare, som att dra lånekort i kortläsare, ta lånekvitto etc.



Som du märker, är det i princip möjligt att dela upp en uppgift i nästan hur små bitar som helst. Men till slut når man en gräns när det inte längre är meningsfullt att beskriva deluppgifter. Var den gränsen går är olika från fall till fall. I biblioteksexemplet räcker det med nivån 2.1 – 2.5. Det ger inte någon större mervärde att ange hur varje tangent ska tryckas in. I andra fall, när det finns hårda krav på att användaren ska göra absolut rätt och på kort tid, t.ex. en processoperator, kan det vara viktigt att analysera knapptryckningar i detalj.

För ett annat exempel, se *Koka te*, av Lars Oestreicher (pdf)

## Alternativ till HTA

Verkliga uppgifter är ofta komplexa. Ett problem med HTA och uppgiftsanlys överlag, är att verkliga problem blir för stora och komplexa att beskriva. Ett annat problem är att metoderna är

begränsade i vilka sorts uppgifter som kan modelleras. Uppgifter som överlappar varandra eller sker parallellt är svåra att beskriva, liksom avbrott eller ovanliga specialfall.

### Collins metod

Ett alternativt och praktiskt sätt att beskriva uppgifter har rekommenderats av Collins (1995). Metoden beskriver förhållandet mellan användare och uppgifter. Förhållandet ställs upp i en tabell, där raderna är uppgifter och kolumnerna är olika användargrupper. Det går också att lägga till en tredje dimension i tabellen för att beskriva uppgiftens/ användarens aktuella omgivning. I tabellens celler (skärningen uppgift/ användare/ omgivning) listas egenskaper för uppgiften: hur ofta, användarens mål, användbarhetskriterier m.m.

Ett exempel från hämtat från Collins (1995). Uppgiftsanalys för ett ordbehandlingsprogram.

Uppgift	Sekreterare	Teknisk skribent	Säljare
Skriva brev	ofta	ibland	ibland
Skriva dokument med flera kapitel	aldrig	ofta	aldrig
Skriva rapport	ibland	ofta	ofta
Lägga in grafik i dokument	ibland	ofta	ofta
O.s.v.	...	...	...

Varje uppgift bör också beskrivas i text med ytterligare detaljer (Collins, 1995):

1. Villkor som måste vara uppfyllda innan uppgiften påbörjas
2. Målet med uppgiften
3. Kritiska situationer där händelseförloppet i uppgiften är problematisk eller att något mål inte kan uppnås beroende på särskilda omständigheter. Identifierade kritiska situationer är möjligheter till förbättringar i systemet.
4. Villkor som måste vara uppfyllda efter uppgiften
5. Roller. Användaren i en särskild roll. Samma användare kan ha fler än en roll.
6. Objekt som är involverade i uppgiften. Objekt kan vara fysiska (t.ex. en penna) eller objekt på skärmen.
7. Utförande av uppgiften i olika steg som bör beskrivas i termer av handlingar med objekten.
8. Viktiga egenskaper hos uppgiften.

Collins metod kan också användas som komplement till andra, t.ex. HTA. Varje uppgift i tabellen analyseras med hjälp av HTA.

### Cooper – målstyrd design

Alan Cooper anser att det viktigaste i design av interaktiva system inte är uppgiften, utan användarnas mål. Cooper menar att det finns en viktig skillnad mellan mål och uppgift. Mål är ett sluttillstånd. Medan en uppgift är en mellanliggande process, nödvändig för att uppnå målet. Fokus på uppgifter kan leda till att man designar systemet efter uppgifter som egentligen inte är centrala för användarna. Rätt analys av användarnas mål, kan leda till de egentliga uppgifterna, vilket ger ett bättre system, enligt Cooper.

Metoden för att åstadkomma detta kallar han för goal-directed design. En viktig ingrediens är ett sätt att beskriva användare på, kallat personas (som togs upp under lektionens andra del, användaranalys). Mer information om goal-directed design och personas finns på följande länkar.

*Goal-directed design* av Alan Cooper  
([http://www.cooper.com/articles/art\\_goal\\_directed\\_design.htm](http://www.cooper.com/articles/art_goal_directed_design.htm) )

### Referenser

Collins, D. (1995) *Designing object-oriented user interfaces*. Redwood City, CA: Benjamin/Cummings.

Cooper, A. (1999). *The inmates are running the asylum*. Indianapolis, IA: SAMS/Macmillan.