

- 1. Börja här: Tre förutsättningar**
2. Så skapas mänskligare datasystem
3. Stå emot! Åtta försvar mot det som stressar oss
4. Se upp! Återvändsgränder, villovägar och fallgropar
5. Ett mänskligare arbetsliv

Börja här: Tre förutsättningar

Om vi ska få både människovärdigare och effektivare datasystem är det mycket som måste ändras och tänkas om. Det finns inte någon snabb fix, någon enda enkel lösning. Bakom problemen finns en rad ganska olika faktorer.

En del handlar om detaljer i hur dataprogrammens funktioner och gränssnitt är utformade. En annan del rör själva metoderna för att ta fram och införa datasystem. I kapitlen *Så skapas mänskligare datasystem* och *Stå emot: Åtta försvar mot det som stressar oss* kan du läsa om ett antal saker som behöver ändras.

Jag tror också det handlar om hur vi ska se på människor och maskiner i arbetslivet som helhet. Om det handlar det sista kapitlet i boken, *Ett mänskligare arbetsliv*.

Men bakom allt detta finns tre grundförutsättningar som jag tror måste uppfyllas för att förändring över huvud taget ska vara möjlig.

Ett: Användarna måste bli medvetna!

Det första som krävs är att alla användare blir medvetna om att dåliga it-system är ett arbetsmiljöproblem.

Det är inte ett individuellt problem.

Jag tror att detta är viktigast. Utan den insikten kommer det inte att gå att ändra på allt det andra.

Jag har föreläst om it i arbetslivet för tusentals personer, i många branscher, från både privat och offentlig sektor. Det är påtagligt hur många som efteråt gör sig besvär med att komma fram och prata. Nästan alla säger samma sak: ”Tack. Det var så skönt! Nu förstår jag att det inte är mig det är fel på.”

Många som brottas med frustration och problem kan inte riktigt sätta fingret på vad felet är. Det är inte konstigt. Den som inte är tränad inom till exempel interaktionsdesign kan ha svårt att se hur – eller ens *att* – de system som de är tvungna att arbeta i kunde vara mer lätthanterliga.

Ett av mina viktigaste syften med den här boken är just att du som användare ska bli medveten om varför du känner dig pressad. I slutet finns också några råd om vad du konkret kan göra på din arbetsplats för att komma till rätta med problemen.

Två: Ensam är inte stark

Den andra grundförutsättningen är att användarna måste ha mod att säga ifrån.

Det är skämmigt att erkänna att man inte behärskar tekniken. Vi förutsätts ju kunna hantera vilket system som helst.

Och ovanpå skammen är många helt enkelt oroliga för att förlora jobbet.

När SVT granskade det skandalösa införandet av patientjournal-systemet Cosmic 2006 var personalen på vårdcentralerna frispråkig – så länge kamera och bandspelare inte var på. Så här beskriver reportrarna vad som hände:

”Vårt största problem har trots allt varit att få läkarsekreterare och sköterskor ute på vårdcentralerna i Uppsala att inför kameran vittna om den stress och olust som Cosmic skapat. Här lyckades vi inte. Deras förklaring lyder att de

inte vågar. De är rädda för att stöta sig med sin arbetsgivare i tider av nedskärningar och omorganisationer.

I stället är det manliga doktorer som vittnar både för sig själv och sina kvinnliga arbetskamrater. Heder åt dem.”⁸²

Kommer du ihåg problemen med det personaladministrativa systemet i Nacka kommun? (Kapitlet *Pålagor, inte arbete.*)

I Nacka stod faktiskt kommunen på sig mot leverantören. ”Det är självklart att vi vill betala mindre på grund av allt strul”, förklarade ekonomidirektören.

Skulle då de anställda, som faktiskt fått bära bördan av det krånglande systemet, få någon sorts kompensation? Eller åtminstone ett erkännande från sin arbetsgivare för sitt slit eller kanske någon form av ursäkt?

När ekonomidirektören kom in på de anställda bytte hon plötsligt både melodi och tonart. När det gällde personalen ville hon ”tona ner problemen”:

”Hon pekar bland annat på att skillnader i datamognad hos personalen komplicerat det hela.

– Det är klart att det tar tid innan folk lär sig den nya arbetsmetodiken. Särskilt för äldre personal som kanske har mindre datamognad. Men inom en femårsperiod ska vi ha sparat in kostnaderna genom minskning av personal.”⁸³

Vilka tror du det blir som får gå?

Och hur villig tror du personalen är att säga vad de egentligen tycker om systemet efter det beskedet?

Ensam är inte stark. Den enskilde ska inte själv behöva stå upp för arbetsmiljön. Det krävs starka arbetsmiljöombud och skyddskommittéer som kan driva frågan på arbetsplatsen. Det krävs samarbete och nätverk, som gärna kan ta sig många former: Facebookgrupper likväl som fackklubbar.

Tre: Prata klarspråk!

Vi måste hitta bättre begrepp för vad som händer i arbetsmiljön än ”lättare psykiska diagnoser”, ”psykosociala faktorer” eller ”bildskärmsarbete”.

Vi måste börja tala om den samlade digitala arbetsmiljön, och om kognitiv ergonomi.

Den som är frustrerad på grund av dålig it ska inte behöva känna att hon lider av en ”lättare psykisk diagnos”. Psykosociala faktorer kan i praktiken betyda allt och inget. Och det viktiga är inte bildskärmarna utan det som finns innanför dem.

Begreppet ”bildskärmsarbete” riktar uppmärksamheten mot frågor om strålning, arbetsställning och påverkan på armbågar eller nacke. Det är också viktiga områden. Men trots lågtstrålande skärmar och ergonomiska datormöss har problemen inte minskat, utan ökat. Därför behöver vi ett nytt begrepp.

Så här skrev en personal- och miljöchef till mig efter en av mina föreläsningar:

”Begreppet *digital arbetsmiljö* var en ögonöppnare för mig.

Som arbetsmiljöingenjör och miljöansvarig har jag tyckt mig ha en bred ansats i att förstå de arbetsvillkor som råder, men som många andra har jag underskattat den kraftfulla aspekt som alla de datasystem som vi dagligen använder utgör tillsammans.

Det blev många aha-upplevelser när du berättade, hur många gånger har man inte själv befunnit sig intrasslad i obegripliga webbsidor där man till slut inte kommer ihåg vad det man ursprungligen försökte få gjort...

Sammanfattningsvis tycker jag att det var en mycket intressant genomgång av en aspekt på arbetsmiljön som varit mycket försummad men som i varierande utsträckning angår oss alla.”

* * *

Har vi dessa tre förutsättningar på plats kommer vi att ha mycket lättare att driva på förändringen kring hur system ska utvecklas och hur de ska införas. Men också när det kommer till makten i arbetslivet.

- 1. Börja här: Tre förutsättningar**
- 2. Så skapas mänskligare datasystem**
3. Stå emot! Åtta försvar
mot det som stressar oss
4. Se upp! Återvändsgränder,
villovägar och fallgropar
5. Ett mänskligare arbetsliv

Så skapas mänskligare datasystem

Egentligen vet vi ganska väl hur man skapar system som är väl anpassade till människor. Redan i mitten av 90-talet försökte det amerikanska Standish Group ta reda på varför så många it-satsningar misslyckades. (Av de drygt 8 000 projekt som var grunden för studien, kunde bara strax över 15 % betraktas som lyckade.)

Man fann att den viktigaste faktorn bakom lyckade projekt var ”user involvement” – att användarna fått vara med i projektet.

När man bedömde vilka faktorer som fick projekt att misslyckas var bilden likadan: ”lack of user input” var den vanligaste orsaken.⁸⁴

Fem år senare gjorde Alan MacCormack och Marco Iansiti vid Harvard Business School en djupare och mer kvalitativt inriktad studie. Även i den visade det sig att användarnas synpunkter var avgörande för lyckade it-projekt:

”The most remarkable finding was that getting a low-functionality version of the product into customer’s hands at the earliest opportunity improves quality dramatically”, konstaterade forskarna.⁸⁵

En ”low-functionality version” kan vara en enkel skiss på papper av hur det ska se ut på skärmen. Runt skissen för man samtal med

användarna, eller ger dem uppgifter som ”Om du vill få reda på saldot på ditt konto, var skulle du klicka då? Peka!”

Resultatet från sådana tester ger underlag för nya prototyper, stegvis allt mer avancerade, som testas på samma sätt, till dess man har en helt färdig produkt att leverera.

Detta sätt att arbeta kallas *användarcentrerat*.

Att använda enkla prototyper är bara ett av många verktyg inom användarcentrerad utveckling och design. På fackspråk har de ofta engelska namn som personor, scenarios, wireframes, contextual inquiry och många andra.

Verktygen finns alltså. Men de används inte.

Frågan är då – varför?

Svaret har med it-branschens roller och metoder att göra.

Användaren har ingen egen representant i it-utvecklingen

Idag hävdar många it-företag deras produkter är ”användarvänliga” eller ”användbara”.

Visst händer det att de anlitar specialister inom användbarhet, till exempel som en del av en förstudie. Ibland gör de användningstester när produkten är nästan klar.⁸⁶

Men det rör sig nästan alltid om begränsade insatser. Så som it-utveckling sker idag, finns det så gott som aldrig någon person med uppgift att bevaka användarnas intressen under hela processen – och som har makt att påverka den.

Låt mig göra en jämförelse med hur man bygger hus (något människor sysslat med, tämligen framgångsrikt, i ett par tusen år).

Byggnadsingenjörer kan konstruera hus som står pall för många årtionden av väder och vind utan att vittra sönder, och som till och med kan klara jordbävningar utan att kollapsa. Deras kollegor VVS-ingenjörerna kan försörja huset med vatten och värme, och elektroingenjörer kan ge huset säker och trygg el.

Trots det är det sällsynt att man låter ingenjörer konstruera hus på egen hand. Sedan mycket, mycket länge engagerar man också arkitekter när hus ska byggas.⁸⁷

Arkitekten ansvarar inte bara för det estetiska, för färg och form. Arkitekten har framför allt det övergripande ansvaret för att *byggnaderna blir anpassade till människor*. Att de fungerar för de aktiviteter de är avsedda för – som kontor eller bostadshus, sporthall eller garage. Det är arkitekten som drar upp ritningarna; som ser till att utrymmena är tillräckligt stora för det de ska användas till. Att man kan ta sig från det ena utrymmet till det andra.

Men det finns ingen motsvarighet till arkitektens roll i it-branschen.

Det finns i och för sig som regel en person som kallas ”arkitekt” i it-projekt, nämligen *system*-arkitekten. Men hennes övergripande ansvar är just tekniken. Den verkliga arkitektrollen, med ansvaret för att det man skapar är anpassat till *människor*, är *tom*.

I många it-projekt plockar man in en eller annan aktivitet från den användarcentrerade utvecklingens metoder. Men utan en person med det samlade ansvaret att se till användarna, och föra deras talan, är risken stor att användarnas unika krav kompromissas bort.

De andra deltagarna i ett projekt driver sina krav. Projektledaren ser till att budget och tidsramar hålls och att arbetet i projektet löper smidigt. Systemarkitekten kan driva krav på tekniken. När projekten måste göra vägval, måste kompromissa eller improvisera, finns det ingen som står upp för användarna. Därför kommer deras krav som regel att hamna underst.

– Vi har följt ett antal införandeprojekt och vår forskning visar att det alltid är andra saker än användbarheten som till slut får styra. Det framgår väldigt tydligt i nästan alla projekt vi har studerat, säger Bengt Sandblad, professor i människa-datorinteraktion i Uppsala.⁸⁸

– Även om ambitionen är god från början är det alltid tekniken som tar över och styr beslut i pressade situationer. Man tycker att det är viktigare att hålla en tidsplan än att systemet som implementeras är användbart.

Personen som borde finnas på arkitektens plats i processen för att ta fram it-system kan kallas informationsarkitekt. Hon kan kallas användbarhets-specialist, ”usability champion”⁸⁹, eller interaktionsdesigner. Den exakta titeln spelar kanske inte så stor roll. Det avgörande är att det måste finnas en person som har ett samlat ansvar och verklig makt.

Det måste också vara en person med rätt kompetens. Det förekommer att man i organisationer tar in någon av dem som ska jobba praktiskt med ett kommande system – en ”vanlig” handläggare, polis, lärare eller sjuksköterska – i själva projektgruppen. (Projektmetoden DSDM föreskriver till exempel att en användare ska sitta med utvecklarna på heltid.) Många företag som utvecklar it-produkter framhåller att de har regelbundna möten med användare, exempelvis i fokusgrupper.

Det är gott tänkt, men i de flesta fall inte en tillräcklig (eller ens bra) lösning. Användarna kan naturligtvis sin verksamhet, men de är inte specialister på hur gränssnitt ska utformas eller hur man ska säkra användbarheten. De har oftast svårt att förstå teknikernas språk och kultur och alla förutsättningar bakom utvecklingen av ett it-system. I praktiken blir de snabbt en sorts gisslan i projektet.

Ingen skulle nog komma på idén att synpunkter från kommande hyresgäster gör en verklig arkitekt onödig. Men så sker regelbundet i it-branschen.

IT-projekt fokuserar på teknik

Kanske borde den här boken inte heta ”Jävla skitsystem!” Kanske borde den heta ”Jävla skitprojekt!”. I de flesta fall är det ju inte tekniken i sig som är problemet. Felen ligger snarare i

- vad man väljer att använda tekniken till,
- hur man utvecklar system,
- hur man planerar för dem,
- hur de införs,
- vilka förväntningar man ställer upp,
- hur man följer upp och utvärderar projekten.

Eller – för att vara tydligare – vilka *dåliga* val man gör. Hur *liten* hänsyn man tar till användarna i utvecklingen. Hur *illa* planeringen fungerar.

Under allt det här ligger dock i sin tur något annat. Utvecklarna, projektledarna och it-cheferna är inte onda. Inte ens politikerna. Men de är alla präglade av en fixering vid tekniken, som är så bred, så djup och omfattande att varken de själva eller vi andra – vi som har att hantera systemen – är riktigt medvetna om att den finns.

Ett litet exempel. Jag har medvetet varit lite slarvig när jag pratat om ”system” i den här boken. Oftast har jag använt ordet om någon typ av program och den eller de maskiner de körs på.

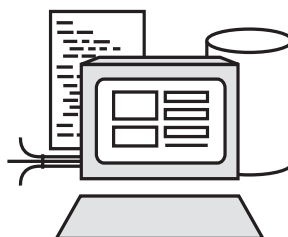
Men ett system är egentligen aldrig bara den där datorn och den där programvaran. Det ”systemet” är bara död materia: magnetiserade partiklar på en hårddisk, i en burk med järn- och kiselskrot. Så förblir det – ända till dess det används av människor.

I praktiken är människor alltid en del av systemen. Det finns människor som matar in data i ena änden av systemet. I någon annan ände finns människor som får del av resultatet av processen: en webbsida att läsa, en summa pengar på sitt lönekonto, en bok med posten, eller ett besked om byggnadslov.⁹⁰

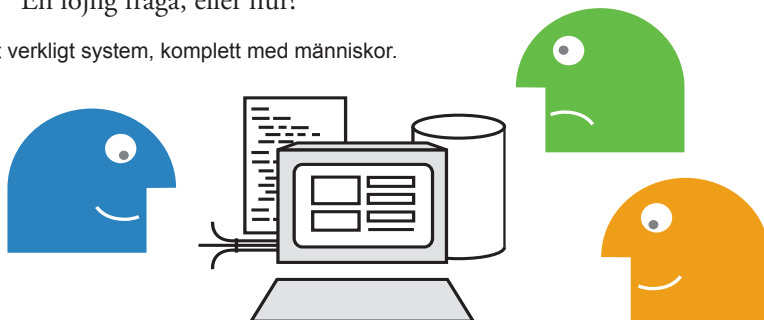
Fundera nu på vilken av de båda bilderna ovan som beskriver ett mer komplicerat system – den första eller den andra?

En löjlig fråga, eller hur?

Ett verkligt system, komplett med människor.



Ett system ur teknisk synvinkel: lite kod, en databas, ett nätverk, några datorer...



Inte nog med att det andra systemet har fler delar. Människor är knepigare att förstå sig på än maskiner. Människor kan känna entusiasm, dåligt samvete, brådska. Få ont i magen. Ha motstridiga känslor. Tjura. Ljuga. Behöva hämta barnen på dagis. Dessutom interagerar de med varandra – inte bara med maskinerna. Louise kan säga ”Det här jävla skitsystemet fungerar inte!” till Mehmet.

Ändå läggs ofantligt mycket mer tid, resurser, pengar, ansträngningar på den mer begränsade delen – det tekniska – när nya system ska skapas.

Den tid som i it-projekt sätts av för att studera användarna eller utbilda dem är normalt bara en bråkdel av den tid som läggs på tekniken.

Det går tusentals platsannonser för systemerare, utvecklare eller projektledare på några enstaka rekryteringar av personer som arbetar med den mänskliga sidan av IT.

Den erfarne konsulten David Craig och journalisten Richard Brooks har i boken *Plundering the public sector* granskat ett stort antal misslyckade it-projekt i Storbritannien. Dagens it-utveckling har genomgående stora brister, skriver de:

”We used to have a simple acronym: UIA – Understand, Improve, Automate. If you wanted to implement an it-system, you first had to understand precisely how the users worked today – what we call the ‘As-Is’. You then should improve current work processes – to create the ”To-Be. Finally, you should design your system to support the future improved way of working. All very simple really.”⁹¹

UIA-modellen har försvunnit, menar Craig. U:et – ”understand” – prioriteras regelmässigt bort. Sedan vänder man på ordningen. Man börjar med att automatisera, och försöker sedan införa ett ”förbättrat” arbetssätt – baserat på tekniken, inte på verksamheten. När resultatet inte blir det förväntade kommer naturligtvis, menar Craig, ”[the] predictable criticism of users for not using the marvellous new system that has been developed especially for them.”⁹²

Craig dokumenterar fall efter fall där man utvecklar system utan att prata med användarna. Ett av dem är det så kallade Libra-systemet – ursprungligen budgeterat till 146 miljoner pund, så småningom löpande på över 400 miljoner pund, utan att någonting ännu fungerar.

”The Department decided not to talk to the people who would use the system and went off and started building a new IT system that did not reflect how the users either worked now or would work in the future. (...)

Presumably, the Department hoped that once the system was installed, some mysterious and magical process would take place whereby all the users were suddenly transformed into happy, efficient, contented employees all working in exactly the same way and all grateful to the Department for providing such a wondrous system to make their working lives more productive and fulfilling.”⁹³

Det finns gott om liknande exempel i Sverige.

Våren 2006 startade ett stort spjutspetsprojekt inom så kallad digital förvaltning – försöken att modernisera och effektivisera administrationen i den offentliga sektorn. Projekt skulle leverera de första tjänsterna som länsstyrelserna skulle ta i bruk för att erbjuda bättre service till medborgarna.

Projektet innehöll ett stort antal tekniska system som skulle integreras med varandra och engagerade ett stort antal konsulter från en mängd stora teknikleverantörer. Den totala budgeten låg på flera tusen timmar.

I projektet fanns avsatt exakt noll timmar för att trygga användbarheten hos systemets insida, de skärmbilder som tjänstemännen på länsstyrelserna skulle använda för att övervaka flödet av ärenden från medborgarna. Noll timmar för att studera hur tjänstemännen arbetade idag. Noll timmar för att testa prototyper.

Först efter hårt arbete från några av mina kollegor lyckades man få igenom ett mycket litet antal timmar för att granska några enstaka skärmbilder ur användarnas synvinkel.

Styr projekt mot effekter – inte mot funktioner

Typiskt för hur it-projekt drivs idag är att man riktar in sig på att skapa en produkt med ett antal *funktioner*. I en kravspecifikation beskriver man att systemet ska kunna göra detta, och detta, och detta.

Redan från början har alltså *användaren* satts i bakgrunden. Man har helt enkelt tappat bort den grundläggande insikt som jag var inne på lite tidigare i det här kapitlet: ett system utan användare är bara död materia. Ett annat sätt att formulera samma sak är detta: *Nytan av systemet uppstår bara om människor använder det.*

Det är en sanning som är så enkel att den ofta glöms bort. Men i dagens it-projekt förutsätter man slentrianmässigt att *om bara funktionerna finns, så kommer de automatiskt att användas.*

I den här boken har jag försökt visa på varför det inte alltid blir så. Dåligt utformade gränssnitt, otillräcklig utbildning, eller en pressad arbetssituation är några av skälen till att användarna finner det rationellt och mer effektivt att *inte* använda ett system.

Men: Det finns ett annat sätt att styra it-projekt.

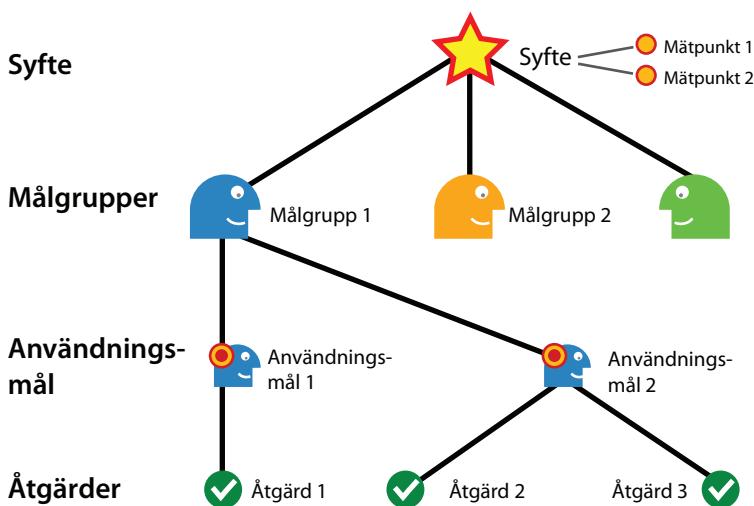
Istället för att beskriva vilka funktioner ett system ska ha, försöker man så konkret som möjligt beskriva de *effekter* man vill uppnå med det nya systemet.

Metoden kallas *effektstyrning* och har tagits fram av de mycket erfarna svenska konsulterna Ingrid Ottersten och Mijo Balic. De beskriver den utförligt i boken *Effektstyrning av IT*.⁹⁴ Här ger jag bara en snabb och översiktlig bild av metoden.

När man arbetar effektstyrt ser man först till att formulera det konkreta *syftet* med projektet. Det ska beskriva den varaktiga förändring av verksamheten som man vill uppnå. Låt oss säga att ett syfte är ”att effektivisera orderhanteringen” i ett affärssystem.

Nästa steg är att ta reda på *vilka* det är som kan bidra till att uppnå syftet. Dessa kallas i metoden för *målgrupper*. (Ett annat möjligt namn vore *effekttagenter*.) I vårt exempel är ordermottagarna säkert en sådan. Troligen försäljarna också – men det kanske finns fler.

Därefter gäller det att kartlägga vad målgrupperna *behöver och vill*, för att kunna bidra till det övergripande syftet. För att få reda på det



I ett effektstyrningsprojekt sammanfattar man arbetet i en *effekt-karta*. Med den säker man att det långsiktiga, övergripande målet – syftet med projektet – inte tappas bort bland detaljerna.

måste vi fråga verkliga representanter ur målgrupperna, eftersom det är de som är de verkliga experterna på verksamheten.

Ordermottagarna säger att för att jobba effektivare behöver de få färre ofullständiga ordrar. Detta är ett av deras *användningsmål*.

Från användningsmålen är det lätt att gå vidare till *åtgärder*: orderformuläret måste förenklas på en rad punkter. Men dessutom behöver kanske säljarna utbildas, så att de i sin tur kan lära kunderna att fylla i formuläret på rätt sätt.

Listan med åtgärder blir en direkt ritning för projektet: Vad ska utvecklas och hur ska det utformas?

Det kanske inte verkar vara så stor skillnad? Slutresultatet, listan med åtgärder, är ju i alla fall en kravlista, ungefär som en kravspecifikation i ett traditionellt projekt.

Men projekt där man listar ”funktioner” får starka inslag av teknisk önskelista: ”Det vore bra om systemet kunde göra *x*. Och *y*. Och helst *z* också.”

Det blir en fundamental skillnad när man i stället utgår från vad användarna behöver för verktyg för att nå de mål de själva varit med om att sätta upp.

Lättare att hålla rätt kurs under projektets gång

Det fattas en viktig roll i dagens it-projekt, konstaterade vi i förra avsnittet: den som ska se till *användningen* av det färdiga systemet. Men också själva *sättet* att driva projekten har alltså brister.

Problemen med att arbeta med tekniken i fokus blir påtagliga en bit in i projekten. Nämligen när man tvingas fatta alla detaljbeslut, som man inte kunde förutse när man drog upp kravspecifikationen.

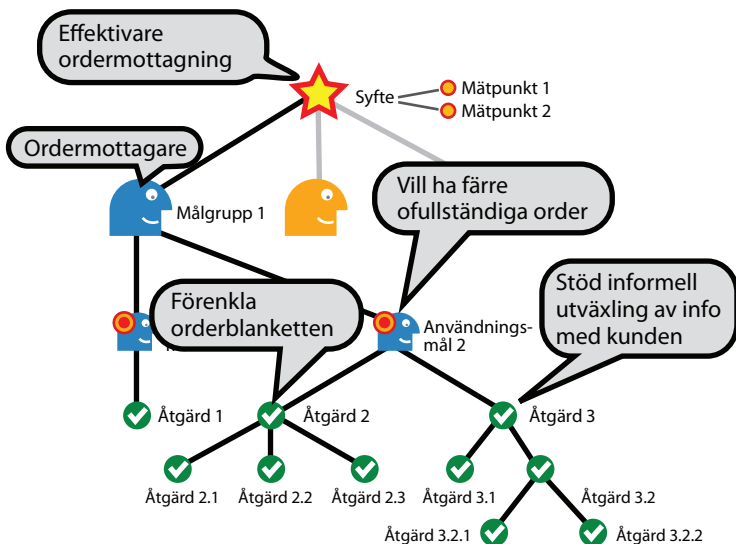
Någon funktion blev mer komplicerad än vi hade trott. Ska vi utveckla den, måste vi offra något annat. I så fall vad? Två andra funktioner kommer oväntat i konflikt med varandra. Vilken ska vi överge? Projektet har dragit över tiden, pengarna är snart slut, vi har fem funktioner kvar i kravlistan men hinner inte göra alla. Vilka ska vi välja?

Det har visat sig, att när man styr projektet mot funktioner blir det svårt att göra kloka prioriteringar i sådana lägen. Valen görs ofta bara på tid: vi är två veckor försenade, alltså stryker vi den funktion där vi beräknat att utvecklingstiden är två veckor.

Men tänk om denna funktion är den viktigaste för användarna, för att de ska kunna utföra sin uppgift snabbare än i det gamla systemet? För att det nya systemet faktiskt ska skapa nytta?

I praktiken är det svårt att hålla visionerna, det övergripande syftet med systemet, levande i det funktionsstyrda projektet. Det har säkert formulerats någon gång i workshops i projektets början. Men det tapas sedan bort bland detaljerna.

Med effektstyrning ställer man frågan: *Vilken åtgärd bidrar mest till att hjälpa användaren att nå sitt användningsmål?* Det är *den* vi ska välja när vi blir tvungna att prioritera.



En effektkarta med exempel på syfte, målgrupp, användningsmål och åtgärder. Både till det övergripande syftet och till användningsmålen finns bestämda mätpunkter, som ska följas upp.

Effektstyrning ger bättre stöd

Finnesserna med effektstyrning är flera. En är att den effektkartläggning som inleder projektet uppmuntrar till att man tänker bortom den rena tekniken.

I vårt exempel kom man fram till att säljarna behövde utbildning. Tack vare effektkartläggningen blir det lättare att se, att utan utbildning får projektet svårt att skapa den utlovade nyttan. Den får alltså inte prioriteras bort. (När man i traditionella projekt hamnar i valet mellan att hinna med att utveckla en teknisk funktion till, eller att hinna med att utbilda användarna, så är det alltid utbildningstiden som får stryka på foten.)

En annan styrka är att man sätter konkreta mätpunkter på syftet – den positiva effekt man vill uppnå. Som vi såg i kapitlet *Det allt högre*

tekniktempot är det sällsynt att man mäter effekter av nya it-system över huvud taget. Med effektstyrning mäter man också de användningsmål som målgrupperna formulerat.

Även för den konkreta designen av själva systemet ger effektstyrningen ett bättre underlag.

I exemplet med ordermottagningen var åtgärden ”förenkla formuläret”. Det kan konkretiseras till åtgärder, varav en kan vara ”gör texten på knapparna tydligare”. Designen blir mer påtagligt kopplad till det syfte vi vill uppnå, när den ligger i slutet av kedjan *syfte > målgrupp > användningsmål*. Den blir mindre ett resultat av tycke och smak hos en enskild designer.

Även i ett effektstyrt projekt använder man förstås den användarcentrerade designens olika verktyg. Användaren får testa en enkel prototyp eller skiss. Förstår hon orderformuläret bättre nu? Gör hon färre fel?

Effekt är ett ord som ledningen förstår

Ytterligare en poäng med effektstyrning är att metoden försöker komma runt den konfrontation som tycks uppstå mellan användarnas krav och organisationens krav på nya funktioner, nya affärsmål.

Ibland verkar de vara rakt motsatta. Användarna vill ha ett snabbt och enkelt system, vill kunna koncentrera sig på att göra sitt jobb. Företaget vill utveckla verksamheten, få nya verktyg och redskap. I praktiken får ”användbarhet” alltid ge vika för ”affärsmässighet” – ofta med en axelryckning. (Som när jag påpekade för några chefer i ett större svenskt företag hur svåränvänt deras affärssystem var för medarbetarna. ”Folk får väl lära sig”, var svaret.)

Genom att sätta fokus på effekterna kan båda sidor hitta ett gemensamt mål. God effekt av sina investeringar är något som även företagsledningen vill ha. Och eftersom goda effekter – verklig nytta – bara uppstår om systemet faktiskt används, får även användarnas krav en naturlig plats i projektet.

IT-projekt behöver en central roll som tar tillvara användarnas intressen, skrev jag tidigare i det här kapitlet.

I effektstyrningsmodellen talar man om en effektledare som den tredje rollen, jämte projektledaren och systemarkitekten. Effektledaren är den som genom hela projektet har ansvar för att systemet faktiskt ger den effekt man tänkt sig.

Namnet på rollen är inte så viktigt, skrev jag förut. Men kanske är det lättare att få gehör för en ”effektledare” än en ”usability champion”.

* * *

Detta är mer än en detaljfråga kring projektstyrning. Ytterst handlar det om att återföra makt till användarna. Om att ta hänsyn till användarnas motivation: att vilja göra ett bra jobb.