



Sammanfattning

Fyra Farliga Fenomen:

- Allt rätt från början
- Kontrollera aktiviteter
- Tillfällig resursorganisation
- Identifiera & eliminera risk

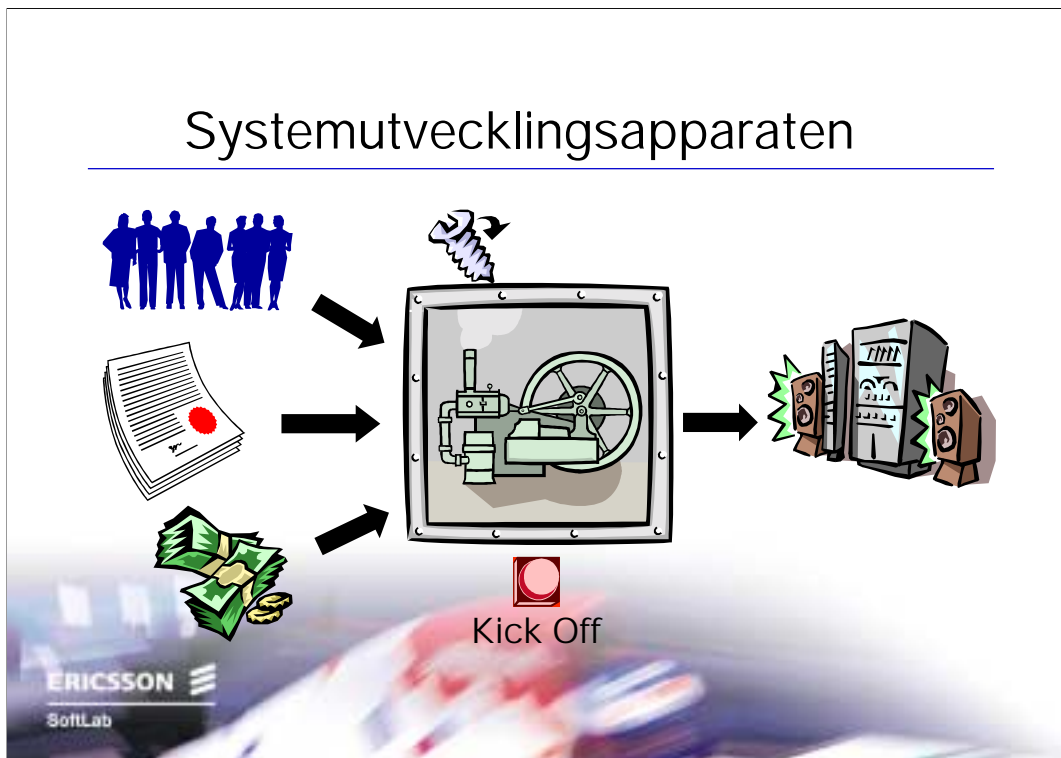


Fyra Framgångsrika Förslag:

- Gör successivt rätt
- Kontrollera levererbart resultat
- Långsiktig produktorganisation
- Acceptera & hantera osäkerhet



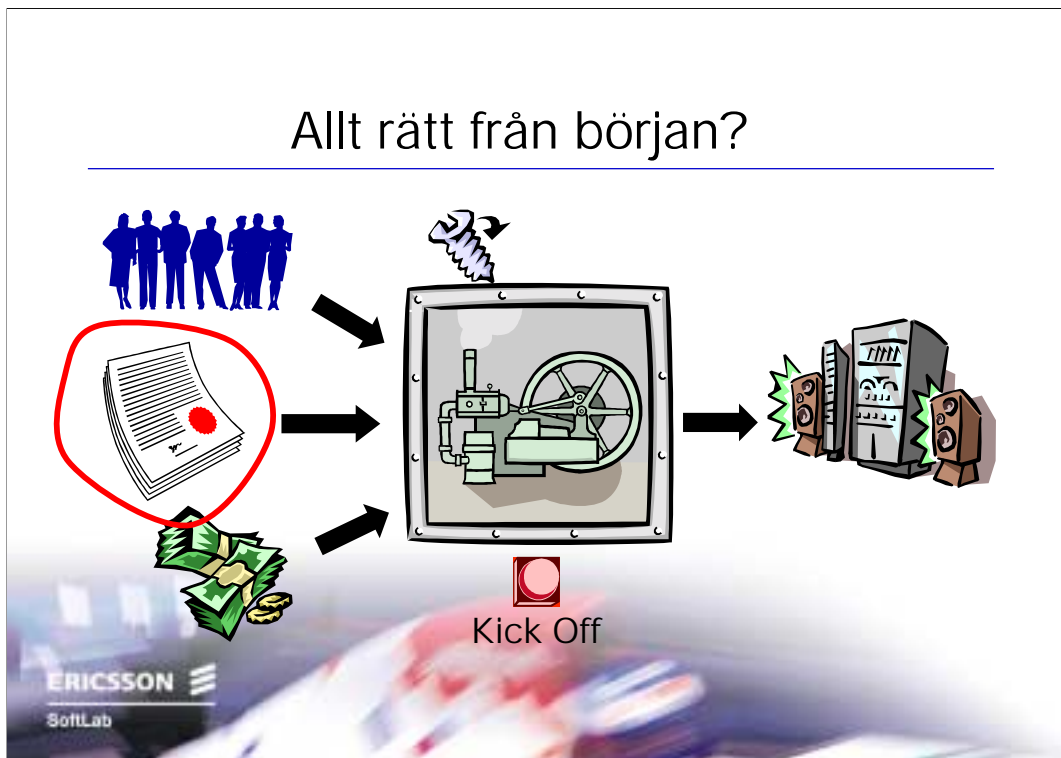
- För företag som ny- och vidareutvecklar egna system (till skillnad från rena resurskonsulter som jobbar på kontrakt) är projektet som arbetsform inte med säkerhet det bästa. Projektformen leder ofta till ett antal fenomen/strategier som inte nödvändigtvis är det man bör eftersträva.
- I presentationen möts "fyra farliga fenomen" med fyra alternativa förslag.



- Tänk om man hade en systemutvecklingsapparat! Man stoppar in lite folk, pengar och en specifikation. Skruvar lite på apparatens för att optimera den till de rådande förutsättningarna, och sedan trycker man på "kick off". Ut kommer, efter något år eller så, systemet som alla vill ha och betala för.
- Det finns ingen sådan maskin, men projektet som arbetsform har dock vissa likheter med apparaten. Presentationen försöka visa likheterna mellan apparaten och projektet, det orimliga/olämpliga i det synsättet, och ett alternativt sätt att bedriva/organisera effektiv systemutveckling.



"Allt rätt från början" - strategin



- "Allt rätt från början"- strategin grundar sig på en uppfattning om att man i förväg kan definiera (entydigt och fullständigt) ett system som uppfyller alla de behov som föreligger när systemet väl levereras.
- Två problem:
 - Systemutveckling är inte momentan utan tar tid, omvärlden förändras och intressenters behov kommer med säkerhet se annorlunda ut när det väl är dags för leverans
 - Det tar tid att göra en fullständig och heltäckande specifikation av systemet, nämligen exakt så lång tid som det tar att utveckla det.

Hur blir allt rätt från början?

"I och med att ett projekt endast genomförs en gång får avvägningarna i de tidiga faserna ofta stora konsekvenser för arbetet i efterföljande faser."

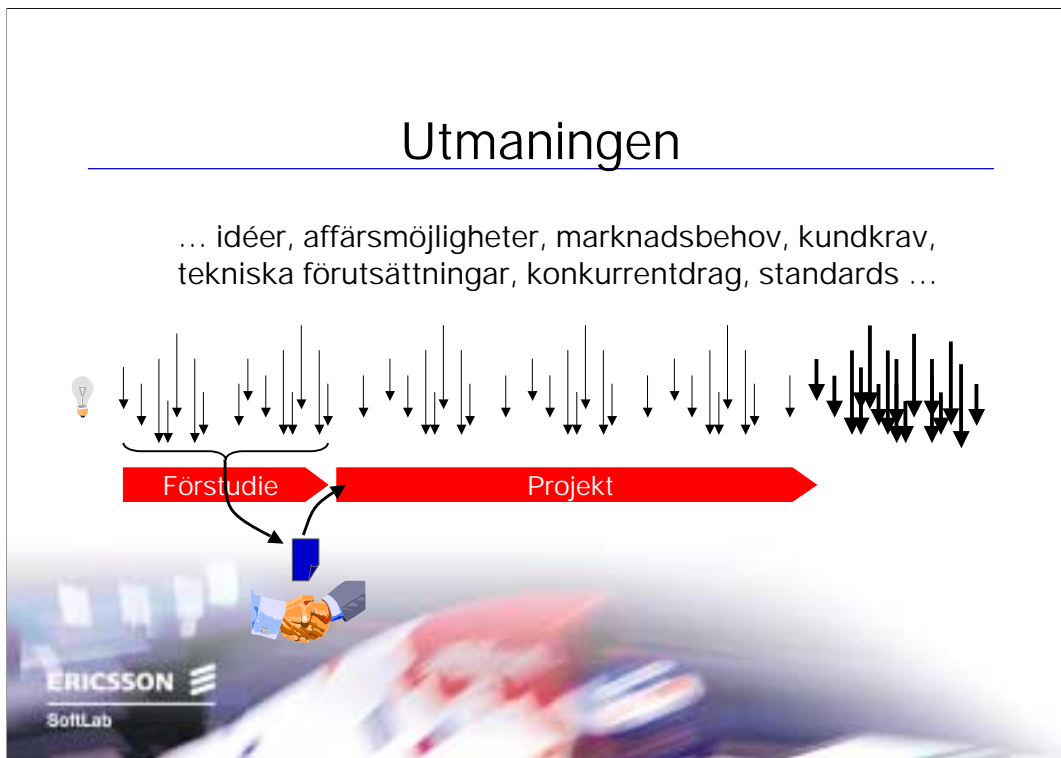
▲Nationalencyklopedin

"Allt rätt från början"-strategin manifesteras i:

- Mer resurser till förstudier
- Bättre metoder för insamling, analys och modellering av krav
- Strävan efter den heltäckande och frysta kravspecifikationen ("kontraktet")
- Tung procedur för kravändring (ändringar i nödfall)



- Hur har vi hamnat i "allt rätt från början"-strategin? Två orsaker:
 - Vi har lärt oss: "Det är billigare att rätta fel i början än i slutet" (men det beror ju till stor del på avståndet mellan "början" och "slutet")
 - Man har en kontraktssituation (beställare – utförare) där parterna anser det vara nödvändigt att i förväg specificera detaljerade villkor för resultatet och utförandet.
- Två kontraktsparter är sällan tydliga i företag som utvecklar egna system mot en marknad. Man försöker då ibland "spela rollerna internt". Det fungerar inte eftersom parterna är ömsesidigt beroende av varandra (symbios) och de faktiska behoven och pengarna också "är på låtsas".



- Den stora utmaningen i "förstudien" blir att samla in alla existerande och framtida behov som den tänkta produkten skall uppfylla.
- Förändringar i omvärlden under projektets gång kan man eventuellt hantera med CR-rutiner. I projekt betraktas förändring som undantag, något man vill undvika (och till och med försvåra). Projektformen med all sin långsiktiga planering blir "jobbig" om man ständigt måste planera om.
- Men, det är först vid leverans de verkliga behoven dyker upp. När intressenter får se och köra systemet får de insikter om vad de egentligen behöver.

*The road to wisdom is plain and simple to express,
to err, and err, and err again,
but,
less, and less, and less.*

Piet Hein



ERICSSON
SoftLab

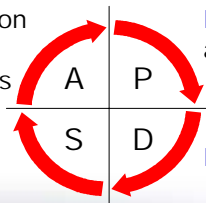
- Piet Hein, dansk poet, författare och levnadskonstnär. Pyslade tydligen en del med systemutveckling också ...
- Det är svårt att göra allt rätt från början. Det behövs mycket klokskap för att bygga rätt system. För att lära oss behöver vi "prova på" och tillåtas att misslyckas. Lämpligen då i det lilla formatet.

Demings "förbättringscykel"

The PDSA Cycle for Learning and Improvement

ACT. Adopt the change, or abandon it, or run through the cycle again, possibly under different conditions

STUDY. Study the results



PLAN. Plan a change aimed at improvement

DO. Carry out the change

W. Edwards Deming

ERICSSON
SoftLab

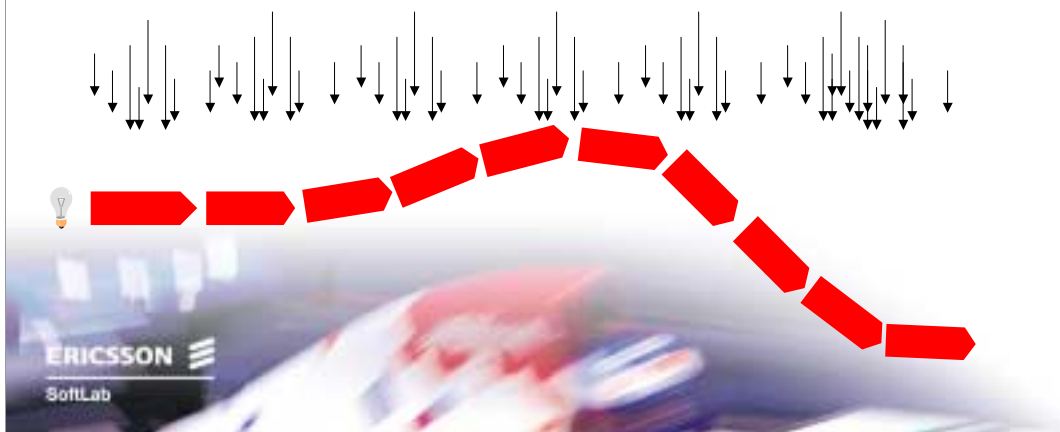
- Demings PDSA-cykel togs fram på 70-talet i arbetet med "statistisk processkontroll". Användningen var processförbättring, dvs förbättring/anpassning av "systemutvecklingsapparaten".
- Inom programvaruindustrin på 80-90-talet lades mycket tid på förbättringar av "produktionsprocessen" som då ansågs vara av samma art som produktionsprocesserna inom andra ingenjörscienser. Inom andra discipliner (bygg, mekanik, el) motsvarar produktionsprocessen ca 80% av utvecklingskostnaderna och stod därför i fokus för effektiviseringarna. Idag menar många att det inte finns någon motsvarighet till produktionsprocessen inom programvaruutvecklingen. Det skulle i så fall vara "kompilera och länka". Allt är design! Och designprocessen är svår att formalisera, även inom andra discipliner.
- Ramkvist, dåvarande VD på Ericsson, delade ut PDSA-cykel till alla anställda under 90-talet. Tanken var att cykeln skulle ligga till grund för "den lärande organisationen" och för allt förbättringsarbete.
- PDSA handlar om att förbättra genom att iterera över ett antal återkommande aktiviteter, att successivt lära sig göra rätt saker genom att få relevant feedback på små stegvisa förbättringsansatser.



- Den viktigaste insikten vad gäller utvecklingsmodeller under de senaste 10 åren är: Systemutveckling är precis det som Demings kallar förbättringsarbete och innebär ett ständigt lärande genom konkret och relevant feedback. PDSA är en utmärkt systemutvecklingsmodell!
- RUP, XP, DSDM etc, alla har de PDSA som grundfilosofi. Det man tyvärr ofta glömmer i tillämpningarna är feedback-steget, att demonstrera för intressenter vad som uppnåtts, vad det körbara systemet nu kan erbjuda, och därmed få den återmatning som ligger till grund för beslut/val-steget.
- Konsekvensen av en "allt rätt från början"-strategi blir att PDSA-cykeln bara körs ett varv, och PDSA motsvarar då faserna i det vi kallar ett traditionellt vattenfallsprojekt. Beslutssteget blir i bästa fall bara ett konstaterande.
- Att hävda att man jobbar inkrementellt (enligt PDSA) därför att man "bygger systemet varje dag" är ett exempel på missuppfattning kring feedback-steget. Vad utvärderas vid ett systembygge? Jo, att man gör saker på rätt sätt, inte att man gör rätt saker. Feedback handlar om återmatning med avseende på "har vi utvecklat rätt funktion/egenskap". Och det är huvudintressenterna som skall göra de bedömningarna.

Gör successivt rätt!

Tidiga och frekventa leveranser av konkreta delresultat (tjänster och egenskaper) är en förutsättning för relevant feedback, som i sin tur säkerställer att vi gör rätt saker, efter hand.



- Slutmålet kan vi oftast inte veta från början. Vi måste ha en strategi för att successivt hitta fram till det.

"Developing software from requirements is like walking on water; it's easier if it's frozen"

Barry Boehm

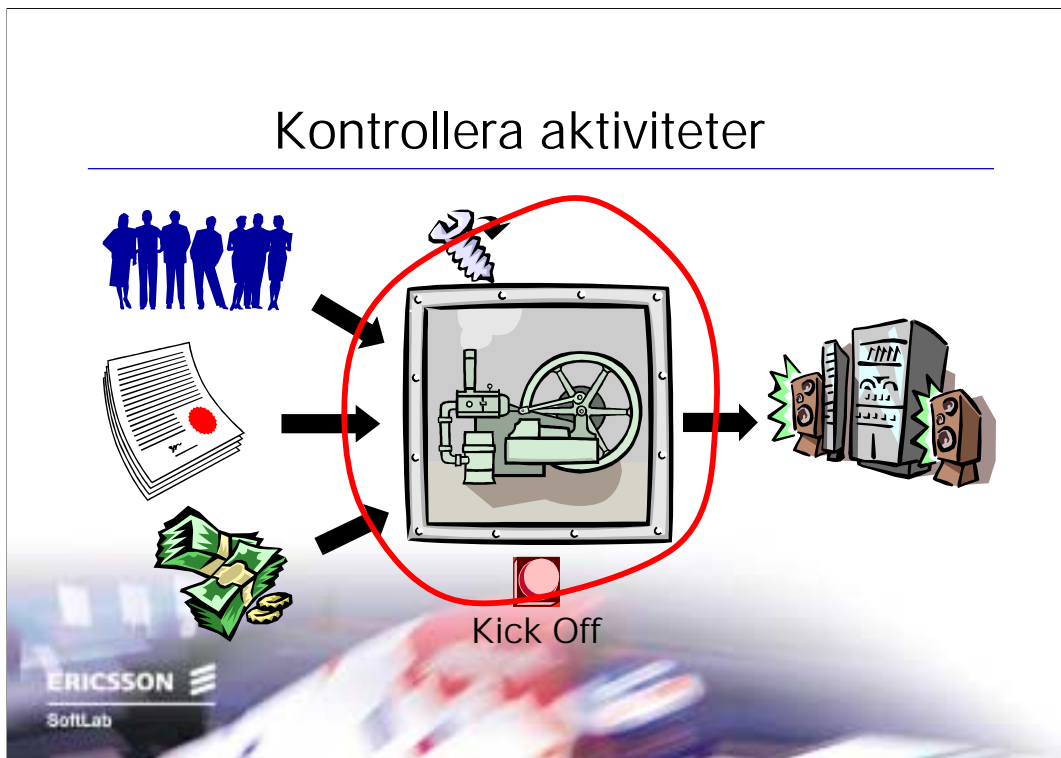


"Okej, men det är faktiskt lättare att frysa en pöl än en hel ocean"

Ulrik Pettersson







- Varför är det ett sådant fokus på genomförandet av aktiviteter i projekt? Planera, följa upp, mäta, effektivisera, etc.
- En förklaring kan vara tron på att ett perfekt genomförande av aktiviteter garanterar system av hög kvalitet. Det stämmer till viss del i tillverkningsindustrin. Där är en "kvalitetsprodukt" en produkt som överensstämmer med det fullständiga och entydiga konstruktionsunderlaget. Och då är det hur man genomför aktiviteterna/tillverkningsstegen som styr kvaliteten.
- I programvaruindustrin har man ett annat kvalitetsbegrepp. En produkt med hög kvalitet är en som kunder/marknader är beredda att betala pengar för, när den väl finns tillgänglig. Det spelar alltså inge roll om "operationen var lyckad" om det visar sig att patienten är död.

I projekt är faser och aktiviteter i fokus

*"Ett strukturerat projektarbete anses genomgå olika faser i en sekventiell ordningsföljd. Faserna följer i princip följande indelning:
1) målformulering/förstudie 2) planering 3) genomförande 4) avslutning."*

▲ Nationalencyklopedin

Det är genomförandet av aktiviteter som "driver". Det är dom man ska:

- estimerar, budgeterar, planerar, resurssätta ...
- genomföra, följa upp, utvärdera
- mäta, förbättra
- avsluta



Vad är problemet med aktiviteter?

Att styra och kontrollera utvecklingen genom att planera och följa upp dess aktiviteter inger falsk trygghet eftersom:

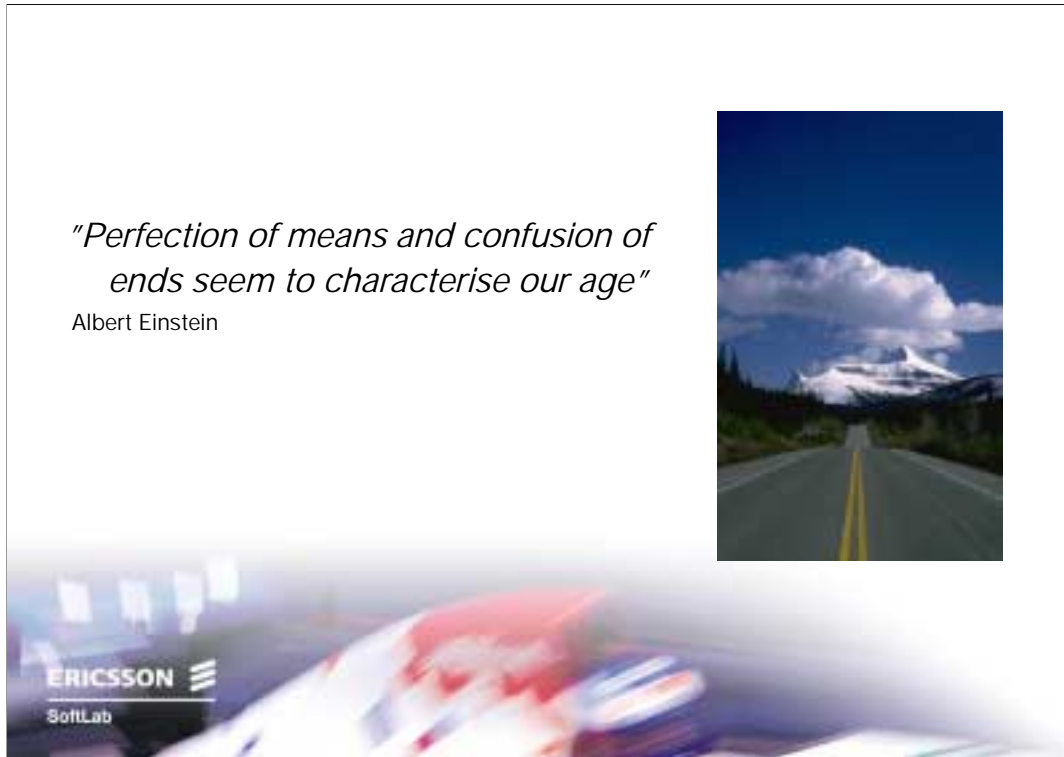
- dom är svåra att särskilja (dokumenttyper)
- dom tenderar att aldrig riktigt vilja ta slut (milstolpar)
- koppling mellan avslut och verklig progress är otydlig (tid)

Att kontrollera aktiviteter är att:

- kontrollera medel, inte mål
- säkerställa att vi gör saker på rätt sätt, inte att vi gör rätt saker



- Med hjälp av dokumenttyper kan vi definiera start/slut på aktiviteter: aktiviteten är klar när dokumentet är godkänt. Men det är fortfarande en otydlig indikation på hur långt vi kommit (verklig progress) och huruvida vi gör rätt saker.
- Milstolpar kan typiskt passeras när vissa specifika aktiviteter är färdiga, dvs när någon typ av information kan betraktas som fullständig eller tillräcklig. I de flesta fall förändras emellertid det tänkta systemets egenskaper och funktion fram till den dag då det levereras. Dvs alla milstolpar passeras i praktiken samtidigt - dagen för leverans.
- Allt vi vet om aktiviteter är att de tar tid. Det innebär ofta att progressuppföljning blir en uppföljning av upparbetad tid. Utan tydlig koppling till verkligt resultat (levererbart värde).



- Ett symptom på den aktivitetsfokusering som råder inom vår bransch: vi har många projektledare ("aktivitetsledare") och projektledarutbildningar, men få produktledare ("resultatstyrare") och produktledarutbildningar.

Kontrollera behov/resultat!

Målet för systemutveckling är att uppfylla intressenters behov. Det görs genom att leverera konkreta tjänster och systemegenskaper som möter behoven.

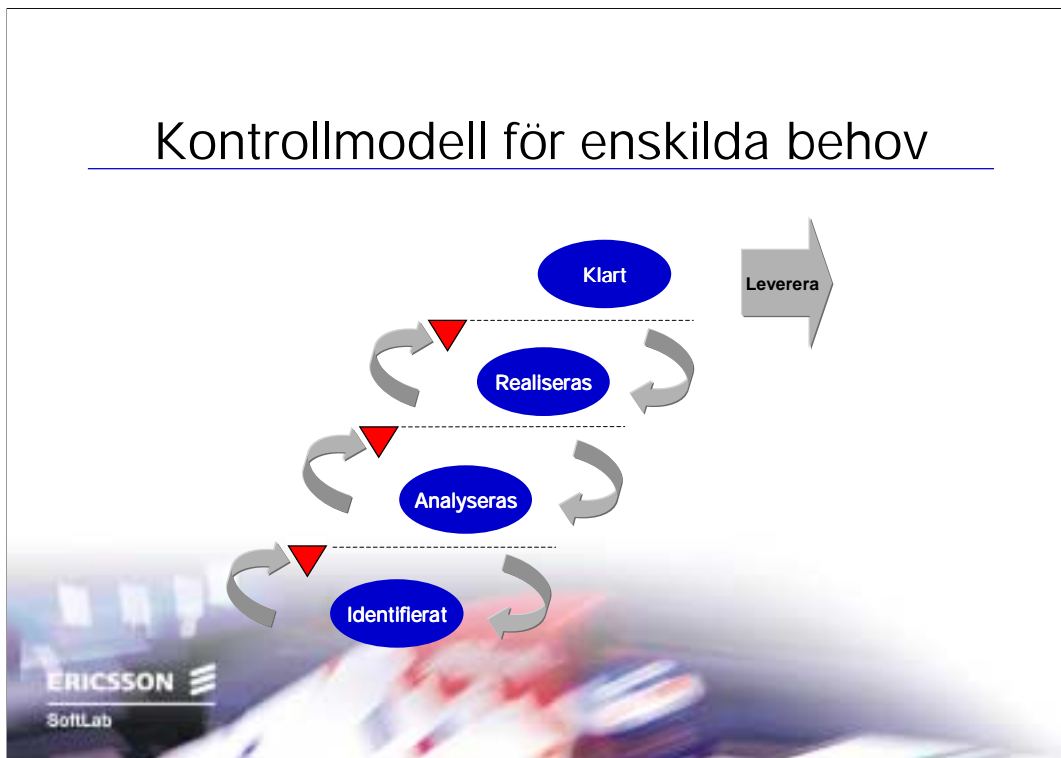
Det är dessa tjänster och egenskaper som primärt ska:

- Hanteras som enskilda objekt
- Värderas, prioriteras, planeras, följas upp och beslutas om
- Mätas, förbättras
- Levereras

Vi behöver en modell för att styra och följa upp tjänster och egenskaper i ett system, inte aktiviteterna i ett projekt.

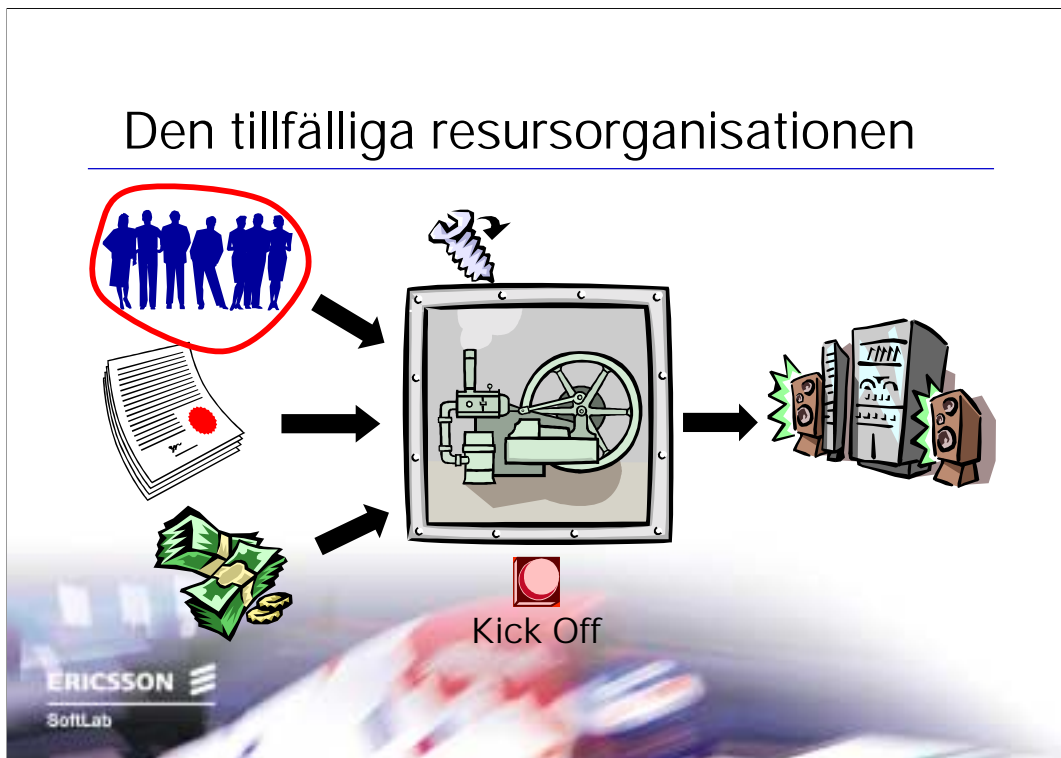


- Vi behöver ett ökat fokus på behov och levererbart/säljbart resultat. Istället för att försöka klumpa ihop alla behov och säga "nu är alla identifierade", "nu är alla analyserade", "nu är alla designade" etc, så måste vi börja betrakta dem mer som enskilda enheter. Detta är en nödvändighet för att kunna styra och kontrollera utvecklingen av ett system som skall uppfylla rätt behov i en ständigt föränderlig värld.



- Ett ärende (behov/krav/mål) kan vara i olika tillstånd (som anger mognad/färdiggrad). Det är tillståndsövergångarna som skall beslutas av produktledningen. Det är egentligen det enda de behöver göra. Många beslut blir det. Och det är ett ständigt pågående arbete. Det räcker alltså inte med att fatta beslut vid någon "tollgate" en gång var tredje månad.
- Det är inte nödvändigtvis en sekventiell ordning på ett ärendes tillstånd. Vid varje beslutstillfälle kan man i princip besluta att tilldela ärendet vilket annat tillstånd som helst.





- Att välja den tillfälliga resursinsatsen som strategi för sin systemutveckling antyder följande förhoppning: om man har en tillräckligt bra systemutvecklingsapparat så kan man stoppa in ett godtyckligt gäng med duktiga utvecklare, och få ett gott resultat.

Projektet är en tillfällig resursinsats

"De medarbetare som engageras för projektarbetet organiseras under projektledaren i en tillfällig projektorganisation, och när projektet är slutfört upplöses denna och dess medlemmar övergår till andra arbetsuppgifter"

▲Nationalencyklopedin

I produktutvecklande företag leder den tillfälliga organisationen till:

- oklarheter vad gäller ansvar och beslut
- onödiga överlämnanden (-> mer dokumentation)
- förvaltningsproblemet ("någon annan får ta hand om det")
- brist på tillämpningskunskap
- ständigt nya processanpassningar
- resursallokeringsproblemet



- Med en linje- och en projektorganisation som samtidigt arbetar inom ett och samma produktområde fås ofta överlappande roller och ansvarsområden som i sin tur innebär otydliga beslutsvägar.
- Åtminstone två formella överlämnanden är onödiga: linje/produktledning till utförare/projekt och utförare/projekt till linje/förvaltare. Vidare består ofta den tillfälliga resursorganisationen av utvecklare med olika specialiteter inom software engineering (testare, arkitekter, designers, dokumentatörer, etc). Dessa tenderar att forma "aktivitetsgrupper" mellan vilka ett antal formella överlämningar skapas.
- Alla vill jobba med nyutveckling (fränt), ingen med underhåll (unket). Trots att det är svårt att exakt säga vad som är vad, och trots att det krävs samma kunskap, engagemang och färdigheter. I projektet finns dock en risk för kortsiktighet. Den tillfälliga organisationen har ingen egen drivkraft för att skapa system som är väl dokumenterade och lätta att underhålla/vidareutveckla – någon annan får ta hand om det.
- Den tillfälliga resursorganisationens kompetensfokus skall vara generell software engineering medan det för ett produktutvecklande bolag ofta är betydligt viktigare med kompetens inom produkternas tillämpningsområde. Den skapas primärt genom att jobba med samma produkter under lång tid.
- För varje nytt projekt skall en generell arbetsmodell anpassas till de unika förhållanden som råder i projektet. Arbetsmodeller bör istället anpassas för produktområden som bör vara mer långlivade än projekt. Därmed krävs inte lika många anpassningsinsatser. Visserligen görs sällan "projektanpassningar" av de generella modellerna, men det innebär bara att man arbetar på ett sätt som med största sannolikhet inte är optimalt.
- "Vi kan inte köra igång detta projekt för det andra projektet är försenat", "Vi låter den där killen jobba 20% i projekt X, 30% i projekt Y och 50% i projekt Z".

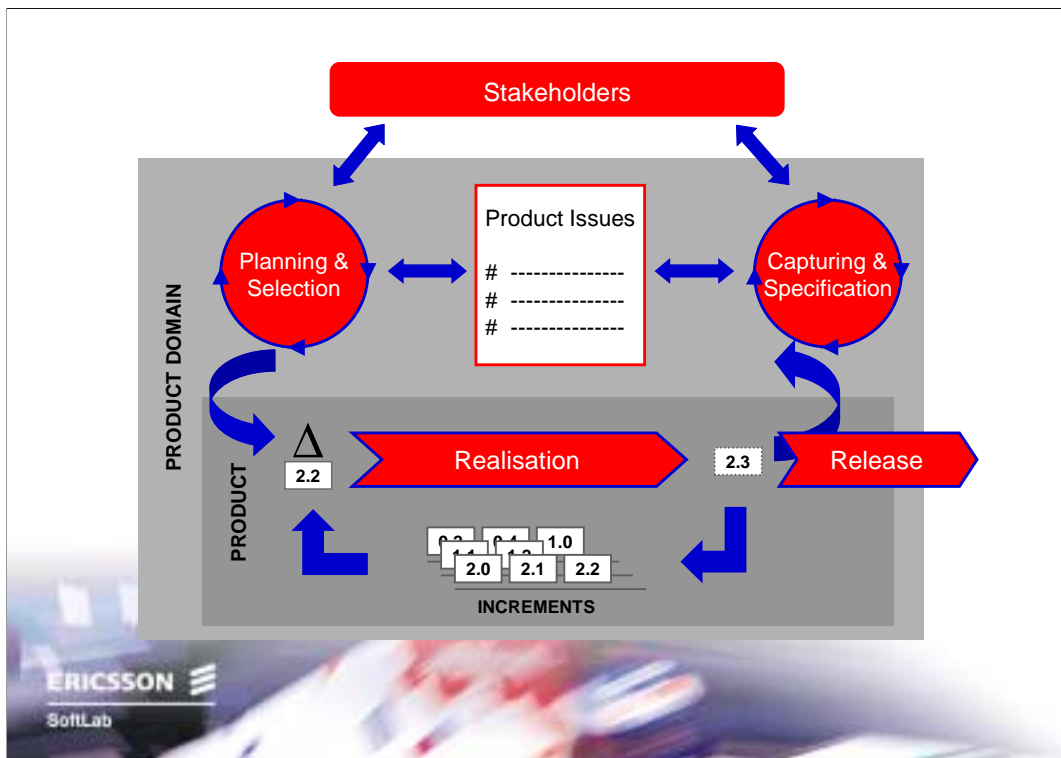
Organisera i produktområden!

Organisationen bör återspegla produktutvecklingens behov av långsiktighet, kontinuitet, domänkunskap och ständigt pågående arbete:

- Definiera produktområden (vision)
- Bemanna med kärntrupp (flexibilitet)
- Ge dem helhetsansvar för både ny- och vidareutveckling (underhåll) av produkter inom området.
- Låt dem forma sin arbetsmodell knuten till produktområdet (och individerna)



- Låt "arbetsorganisationen" vara produktområden, inte projekt. Naturligtvis med flexibilitet vad gäller bemanning, men med långsiktighet och helhetsåtagande i fokus.

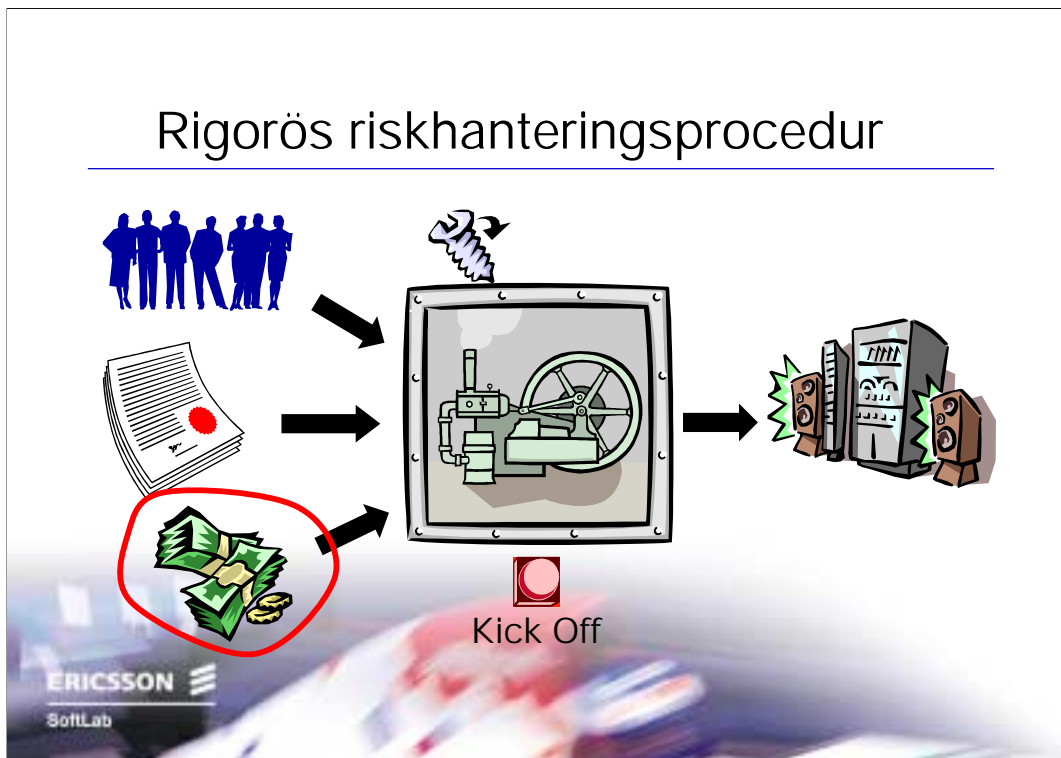


- Ett exempel på en arbetsmodell som etableras för produktområden, ej projekt. En och samma modell för allt utvecklingsarbete inom produktområdet (ledning, nyutveckling och underhåll/förvaltning).



Identifiera och eliminera risk

ERICSSON 
SoftLab



- Den som stoppar in pengar i ett projekt vill naturligtvis försäkra sig om att allt kommer att gå bra, att de investerade pengarna kommer tillbaka. Med ränta! Ju längre bort i tiden systemet förväntas vara klart, desto mer pengar handlar det om. Och desto mer energi läggs det på att identifiera, analysera och eliminera alla risker och osäkerheter.

Osäkerhet – Risk eller förutsättning?

"Riskreducerande åtgärder inleds med riskidentifiering och riskurval, som följs av en riskanalys, där riskkällor studeras från två aspekter: exponering och effekt."

▲Nationalencyklopedin

I traditionella projekt betraktar man gärna osäkerhet som risker. Dessa skall identifieras, analyseras och undvikas.

Men, osäkerhet är oundviklig - en naturlig förutsättning i systemutveckling. Vi kan dock mildra de negativa effekterna. Av alla önskade händelser. Genom att utveckla i små steg.



- Det man kan läsa i riskidentifieringsrapporter är ofta en beskrivning av vår verklighet. De innehåller de omständigheter som gäller för dagens systemutveckling ("otydliga krav", "osäker marknad", "aktiva konkurrenter", "viktiga nyckelpersoner", "omogen teknik", "halvfärdig standard", "aldrig gjort det förr" etc)



- Risken är helt enkelt den summa pengar du är beredd att stoppa in i ett arbete innan du kan se och utvärdera resultatet. Ju mindre "fisk" desto mindre risk!

Sammanfattning

Fyra Farliga Fenomen:

- Allt rätt från början
- Kontrollera aktiviteter
- Tillfällig resursorganisation
- Identifiera & eliminera risk



Fyra Framgångsrika Förslag:

- Gör successivt rätt
- Kontrollera levererbart resultat
- Långsiktig produktorganisation
- Acceptera & hantera osäkerhet





"Little drops of water, little grains of sand
Make the mighty ocean, make the pleasant land"
Julia Carney

ERICSSON 
SoftLab