



Handelshögskolan
VID GÖTEBORGS UNIVERSITET

**Institutionen för
informatik**

2003-06-04

Projektstyrning och hantering av runaway projects

Abstrakt

Runaway projects är IT-projekt som tycks stå utanför organisationens kontroll och fortsätter att leva trots att det rationella är att helt styra om eller avsluta dem. Huvudproblemet i den här studien har varit att undersöka hur olika projektstyrningsmodeller stöder en hantering av runaway projects. Dels har moderna projektstyrningsmodeller i systemvetenskaplig litteratur studerats och dels har en etnografisk studie skett på IT-avdelningen på Försäkringskassan i Västra Götaland. De projektstyrningsmodeller som har studerats fungerar dåligt som stöd för hantering av runaway projects. Detta beror mycket på dess utopiska karaktär samt avsaknad av hantering av de psykologiska påfrestningar som människan utsätts för i projektarbete och beslutfattande. På Försäkringskassan går varje pågående systemprojekt igenom varje vecka av utvecklingsgruppen vilket fungerar som ett mycket bra stöd. En brist är att det saknas metoder för projektuppföljning vilket gör det svårt att lära sig av egna erfarenheter.

Nyckelord: runaway, projects, project, escalation, commitment, projektstyrning

Författare: Erik Fredman
Handledare: Urban Nuldén
Examensarbete 1, 10 p.

Innehållsförteckning

1 Inledning	3
1.1 Syfte	3
1.2 Undersökningsfrågor	3
1.3 Disposition	4
1.4 Miljö- och situationsbeskrivning	4
Riksförsäkringsverket	4
IT-avdelningens roll i organisationen	4
Arbetsituation	5
Projektstyrning	6
2 Teoretiskt ramverk	7
2.1 Runaway system projects	7
2.2 Faktorer som påverkar project escalation	8
2.3 Förklarande teorier	9
Self-justification theory	9
Prospect theory	9
Agency theory	10
2.4 Angreppssätt mot project escalation	10
Utgångspunkt i påverkande faktorer	10
Utgångspunkt i self-justification-, prospect- och agency theory	11
2.5 Learning failure	12
2.6 Projektstyrningsmodeller i systemvetenskaplig litteratur	15
3 Material och Metod	20
3.1 Datainsamling	20
3.2 Intervjuer	20
3.3 Observerande deltagare	21
3.4 Metoddiskussion	21
4 Resultatredovisning	22
4.1 Projektstyrning i utvecklingsgruppen	22
4.2 Erfarenhet av Runaway projects	24
Fallbeskrivning 1	24
Fallbeskrivning 2	25
Övrig erfarenhet	25
4.3 Tillgänglighet av projektinformation	27
4.4 Personligt ansvar för utgången	28
4.5 Metod för återkoppling och lärande	29
4.6 Stöd i projektstyrningsmodellen	29
Datumstyrd utveckling	29
Projektspråk	29
Projektdokumentation	30
Aktivitetsplan	31
Utvärdering	31
5 Diskussion	33
6 Slutsats	37
Referenser	38
Bilaga 1 Intervjufrågor	39
Bilaga 2 Tre teorier om eskalering och dess underliggande antaganden	40

1 Inledning

Projekt, som har blivit en naturlig arbetsform inom IT, kan tyckas vara relativt lätt att styra med lättillgänglig kunskap och ”steg för steg” anvisningar. Dock kan många projekt i efterhand mer eller mindre anses som misslyckade p.g.a. missade mål, budgetöverskridning eller kraftiga förseningar etc. Det kan ibland konstateras att beslut borde ha tagits på avgörande tidpunkter för att omdirigera eller t.o.m. avbryta arbetet vilket inte har gjorts (Keil 1995c). IT-projekt som har tendenser att leva sina egna liv och tilldelas mer resurser än budgeterat och som aldrig tycks avslutas kallas ”runaway systems projects”. Keil har beskrivit olika faktorer som har avgörande betydelse för om det finns risk för att beslut som behövs tas i projekt inte kommer att tas. Individuella och organisatoriska riktlinjer definieras för att kunna granska den egna situationen och se om det föreligger risk för ett irrationellt handlande. Organisationer i allmänhet förväntas ha en form av standardiserad arbetsprocess vid arbete i projektform som skall verka som stöd för att projekt skall få en lyckad utgång. Denna arbetsprocess kan kombineras med riktlinjer för styrning och hur arbetet skall fortlöpa och tillsammans skapar det vad som i denna rapport kallas för projektstyrning. När projektstyrningsmetodik beskrivs i litteraturen innefattar den faktorer som kvalitetssäkring, riskbedömning, schemaläggning, resursallokering etc. Vad som är det intressanta i sammanhanget av den här studien är om det finns faktorer i projektstyrningsmodeller som stödjer en hantering av runaway projects.

1.1 Syfte

Runaway systems projects är en typ av fenomen som förekommer i många organisationer till vilket mycket resurser går åt. Resurser som kunde ha investerats i mer meningsfulla projekt som stärker konkurrenskraften istället för tvärtom. (Keil 1995b) Denna rapport är ett försök till att bidra med kunskap som kan användas i arbetet med att förbättra styrningen av IT-projekt. Projektstyrningsmetoder skall användas i syfte att skapa mer värde för de pengar och resurser som satsas. Samtidigt bör man ha stöd under projekttiden för att identifiera och avbryta eller styra om projekt som sannolikt kommer att misslyckas. Ett syfte är således att bidra med kunskap inom området där forskning pågår om hur projektmetodik kan utvecklas för att höja en organisations konkurrenskraft genom att höja produktkvalitet i förhållande till satsade resurser.

1.2 Undersökningsfrågor

Den övergripande frågeställningen för studien är:

- Vilket stöd ger projektstyrningsmodeller för hantering av runaway system projects?

En fallstudie kommer att företas inom en systemutvecklingsgrupp på Försäkringskassans IT-avdelning i Västra Götalandsregionen. Detta görs i syfte att utreda den övergripande frågeställningen samt att besvara följande fallstudiespecifika frågor:

- Har utvecklingsgruppen upplevt runaway projects?
- Vilket stöd finns i nuvarande projektstyrningsmodell för hantering av runaway projects?

Parallellt sker en studie av moderna projektstyrningsmodeller som presenteras och rekommenderas i systemvetenskaplig litteratur.

1.3 Disposition

Rapporten börjar med en beskrivning av Försäkringskassan där studien har genomförts. Beskrivningen syftar till att ge läsaren en möjlighet att sätta in det material som presenteras i ett sammanhang vilket ökar förståelsen. I teoridelen tas i huvudsak två linjer upp som beskriver olika förklarande faktorer bakom fenomenet runaway projects. Därefter sker en kort genomgång av vanliga projektstyrningsmodeller som beskrivs och rekommenderas i modern systemvetenskaplig litteratur. I resultatdelen behandlas det som har framkommit vid observationer och intervjuer på Försäkringskassan. Empirin följs av en diskussions- och analysdel där författaren utreder frågeställningen i kontexten av teori och empiri. I delen eftersöks vetenskapliga samband som har avgörande betydelse för studiens utfall. Diskussionen skall vägleda läsaren in i de slutsatser som dras i slutet av rapporten.

1.4 Miljö- och situationsbeskrivning

Riksförsäkringsverket

Riksförsäkringsverket (RFV) tillsammans med försäkringskassorna svarar för huvuddelen av landets sociala skyddsnät genom administrering och utbetalning av bl. a. sjuk- och föräldraförsäkring, ålderspension och barnbidrag. RFV agerar på central nivå genom tillsyn och utvärdering av verksamheten i syfte att säkerställa kvalitet och likformighet vid handläggning av ärenden. Ärenden handläggs på lokal nivå av enskilda försäkringskassor runt om i landet. RFV Data ansvarar för utveckling, förvaltning och produktion av IT-stöd för försäkringskassorna och RFV. Cirka 650 personer jobbar i dagsläget på RFV Data med uppgifter som spänner över hela IT-området.

IT-avdelningens roll i organisationen

Försäkringskassan i Västra Götaland har en mindre grupp (4 personer) som sköter utveckling av visst IT-stöd på de olika kontoren över hela Västra Götalandsregionen. Gruppen fungerar som internkonsulter och tar emot uppdragsförfrågningar främst från grupp- eller avdelningschefer. Uppdragen granskas och utvärderas utifrån ett antal kriterier för att säkerställa att satsningar görs på projekt som ger den största kundnyttan. Ansvarsfördelningen mellan RFV Data och utvecklingsgruppen är inte helt fastställd och beskrivs lite som en gråzon. Dock fungerar det i praktiken tillfredsställande att avgöra vilken typ av IT-stöd som

skall utvecklas på de olika nivåerna. Det har funnits förespråkare som har ställt sig negativa till lokal systemutveckling. Dock är den frågan i stort sett avgjord och man har kommit fram till att det bör finnas. Den studerande utvecklingsgruppen utvecklar s.k. ”verksamhetsnära” eller ”användarnära” applikationer i syfte att tekniskt stödja processer i specifika arbetsuppgifter på kontoren inom regionen. Levererade applikationer har, till skillnad från RFV Datas applikationer, relativt få användare och används oftast lokalt men man strävar efter att hitta skalbara lösningar som även går att använda vid utveckling av IT-stöd för andra arbetsgrupper. I flera fall har utvecklade applikationer börjat användas på försäkringskassor även utanför den egna regionen.

Ett krav som finns vid exportering av en applikation till en annan region är att applikationen har genomgått en certifieringsprocess och godkänts av RFV Data innan användning börjar. Syftet med denna certifieringsprocess har främst varit ur driftsynpunkt, dvs. att minimera risken för driftstörningar orsakade av applikationer med bristande kvalitet. I dagsläget pågår ett stort projekt med att byta plattform på både klient- och serversidan i hela socialförsäkringsorganisationen. Bytet innebär också en övergång till en annan utvecklingsmiljö och delvis nya utvecklingsverktyg för den studerade gruppen. I projektet ingår att se över hela applikationsutbudet för att dels rationalisera bort överflödiga applikationer och dels migrera över applikationer till den nya plattformen. I och med övergången skall strängare regler implementeras kring hur applikationsutveckling skall gå till samt att all utveckling av applikationer skall certifieras av RFV Data innan de tas i drift. RFV Data kan alltså sägas agera som en övermyndighet över IT-avdelningen i Västra Götaland.

Arbetsituation

IT-avdelningen har under de senaste åren genomgått stora organisationsförändringar. År 1999 slogs hela regionen ihop till en Försäkringskassa då en för regionen central IT-avdelning bildades. Senare startades ett projekt för utveckling och implementering av en helpdesk för hela regionen. Helpdesk har varit i drift sedan oktober 2001 och har blivit mycket uppskattad. Tidigare arbetade IT-personal främst med användarstöd och själva driften av systemen. Den applikationsutveckling som företogs skedde utan ett fastställt arbetsförfarande. Personalen tog många ”korridoruppdrag” dvs. vem som helst kunde göra en applikationsbeställning och sedan togs beslut om utveckling enskilt av beställningsmottagaren. En utveckling och uppdelning har sedan skett av olika arbetssysslor som systemutveckling, teknik, strategifrågor, kompetensutveckling och användarstöd. En teknisk arbetsgrupp som bl. a. till stor del sköter regionens plattformbyte är placerad i Lidköping.

Utvecklingsgruppen som informellt har verkat i några år men som formellt har verkat i drygt ett år arbetar i dagsläget betydligt mer strukturerat än tidigare. Man arbetar till största delen med utveckling av IT-stöd främst mot specifika handläggargrupper. I vissa fall köper man in kommersiella applikationer och då arbetar gruppen med implementering och utbildning. Vid sidan om detta sker arbete med utformning av IT-strategi för regionen samt inventering och utveckling av kompetens samt drift av server. Man har nyligen gått från en tillämpning av Klient/Server (DLL:r s.k. ”Windows-applikationer”) till rena webblösningar vilket är en del av IT-strategin och ger lättare installation och underhåll. Sammanlagt har gruppen ca 35 års erfarenhet av systemutveckling (4 personer).

Projektstyrning

Gruppens arbete sker hela tiden i projektform med minst två personer som projektdeltagare. Säreget för gruppen är att medlemmarna ständigt ingår i ett antal projekt och något egentligt linjearbete existerar inte. Förfrågningar och beställningar av uppdrag har kanaliserats och skall numera gå genom gruppens chef. Beställningar föregås av en förundersökning enligt en bestämd process och noga överväganden sker över vilka uppdrag som skall accepteras. Projekttiden är relativt kort (4-6 månader), men det förkommer en del längre projekt. Man har tidigare delvis arbetat efter projektstyrningsmetoder utvecklade på central nivå hos RFV och RFV Data. De metoderna är dock skapade för större projekt och har således skapat mycket onödigt merarbete i form av s.k. ”overhead”. Relativt nyligen har gruppen implementerat en ny metodik för projektstyrning som anses mer anpassad till de aktuella projekten. Efter en kort tids användning är nu gruppen intresserad av en utvärdering av hur den fungerar och hur man kan utveckla metodiken. Studien omfattar endast utvecklingsgruppen och dess syn på runaway projects och projektstyrningsmodeller. Studien behandlar ämnet projektstyrning med utgångspunkt i fenomenet runaway projects, inte riskhantering eller projektstyrning i allmänhet.

2 Teoretiskt ramverk

2.1 Runaway system projects

Redan 1974 uppmärksammades i forskningen det intressanta fenomenet med IT-projekt som fortgår betydligt längre än vad som är rationellt eller något vidare sunt. Runaway system projects kan beskrivas som projekt som tycks få och leva ett eget liv utan att någonsin nå fram till de mål som en gång var uppsatta (Keil 1995c). Stephen Keider ger sin beskrivning på runaway projects (Keil 1995c):

Rather, he said, they become like Moses, condemned to wander till the end of their days without seeing the promised land.

Det finns betydande ekonomiska risker med sådana projekt vilket vi också kan se tydliga exempel på. Ett exempel är projektet Taurus som skulle utveckla ett handelssystem på Londonbörsen. Projektet avslutades efter 6 år utan att en enda modul hade installerats och hade då kostat sammanlagt \$700 miljoner. En mängd projekt kan kopplas samman med fenomenet och det faktum att mycket pengar har använts utan att de uppsatta målen har uppnåtts. Man kan ofta i efterhand konstatera att dessa projekt fick pågå betydligt längre, överhuvudtaget eller i samma riktning, utan att beslut om ändringar togs (Keil, Mixon, Saarinen, Tuunainen 1995, Keil 1995c, Lyytinen & Robey 1999). Istället fortsätter de involverade att hålla liv i projekten genom att chefer hela tiden satsar mer pengar och resurser, eller som Keil (1995a) uttrycker det: "throwing good money after bad". Det har talats mycket om misslyckade IT-projekt och anledningen därtill. Trots att flertalet försök har gjorts att förklara och komma till rätta med systemutvecklingsproblem är systemutveckling förknippat med hög risk (Lyytinen, Mathiassen, Ropponen 1998). Förklaringar kring havererade IT-projekt sker ofta i allmänna termer utan konkreta inslag. Dålig projektstyrning tycks vara den mest vanliga förklaringen och innefattar ett brett spektrum av olika alternativ (Keil 1995a). Ett flertal försök har gjorts i syfte att förklara fenomenet med runaway system projects.

Ett begrepp som är nära knutet till, och till stor del beskriver, runaway system projects är "escalation of commitment". Escalation of commitment definieras av Brockner (Keil 1995a) som:

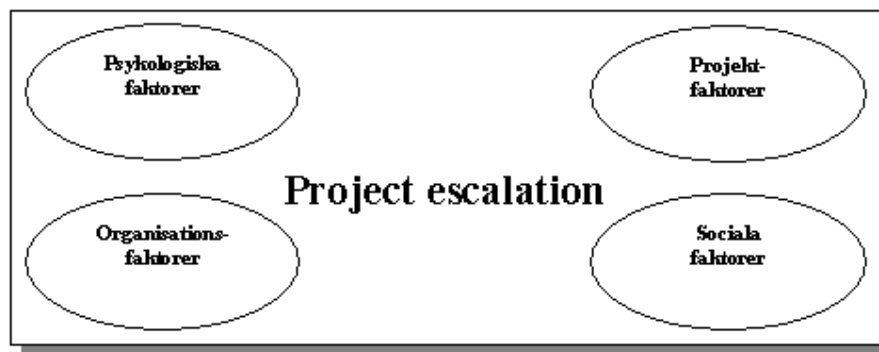
Ett kontinuerligt engagemang trots vetskapen om tidigare felsatsningar som äventyrar målluppfyllelsen.

Trots uppenbara brister eller vetskapen om en ouppnåelig målbild fortsätter alltså ansvariga att hålla liv och engagera sig i projektet. Project escalation uppstår i projekt där ovan nämnda händer, dvs. projektet fortsätter med fortsatt engagemang trots uppenbar negativ information. Negativ information är information som beskriver projektets bristande kvalitet, t. ex. en ouppnåelig målbild, budgetöverskridning eller kraftig försening (Keil 1995a). Projektet Taurus är ett exempel på project escalation. Projektledningen ignorerade med automatik alla

varningssignaler om risker och intresser fortsatte att påverka för en fortsatt systemutveckling trots att det fanns oklarheter kring syftet med och designen av systemet. Mot slutet fanns det många av projektets förespråkare som krampaktigt, nästan vidskepligt, höll fast sin tro på projektet och avlog alla förändringsförslag (Lyytinen & Robey 1999).

2.2 Faktorer som påverkar project escalation (Keil 1995b)

Project escalation är ett komplext fenomen och kan orsakas av ett flertal faktorer. Dessa faktorer kan enligt Keil delas upp i fyra kategorier: projektfaktorer, psykologiska faktorer, sociala faktorer och organisatoriska faktorer.



Figur 1. Faktorer som påverkar project escalation (Omarbetat efter Keil 1995b, s. 2)

- Projektfaktorer är projektets mål och resultat samt hur dessa uppfattas och styrs i projektet. Kostnad och vinst är projektfaktorer som påverkar engagemanget. Desto högre insatsen och avkastningen är, desto viktigare och engagerande blir det. Projekttiden kan också påverka project escalation. Ett långsiktigt projekt kan resultera i project escalation speciellt om det krävs ett långsiktigt arbete innan man börjar göra vinst på investerat kapital. Projekt som förknippas med en högre svårhetsgrad påverkar också vid sidan om vilken syn man har på motgångar i projekt. Negligeras motgångar och endast ses som överkomliga problem ökar risken för project escalation.
- Psykologiska faktorer hör speciellt samman med den/de som leder det aktuella projektet. Faktorer som orsakar ett behov av att övertyga sig själv att projektet ser bättre ut än vad det gör kan sägas höra till denna kategori. Så länge man fortsätter att arbeta på som vanligt så löser sig problemen och resultatet blir lyckat. Sådant resonemang kan föras av projektledare som tidigare har arbetat i liknande projekt eller känner ett starkt personligt engagemang för projektresultatet. Risken för project escalation ökar om det finns en historia av lyckade projekt där projektledningen/ledaren har varit delaktig samt om det finns en hög grad av personligt resultatansvar. Ytterligare faktorer har att göra med hur information om projektet tas emot och bearbetas. Beslutstagare producerar en mängd snedvridningar och manipuleringar av information i beslutssituationer vilket ökar risken för project escalation. Denna snedvridning sker oftast undermedvetet och kan t. ex. lätt

leda till en satsning av ytterligare resurser och projektinsatser för att styra rätt ett projekt som är på glid.

- Sociala faktorer som till exempel gruppfiendlighet eller hård tävlingsanda kan påverka ett rationellt handlande och beslutsfattande. Behov av existensberättigande är en annan sak som påverkar, t. ex. en grups behov av att externt bli berättigad sin position och uppgift i organisationen. Project escalation händer oftare då det finns en rivalitet mellan grupper som har, respektive inte har, makt att ta vissa beslut. Vidare ökar risken för project escalation om projektintressenter har intalats att projektet går enligt plan samt att det finns en policy som förespråkar att man skall ”hålla kursen”, dvs. arbeta på i samma spår trots problem.
- Organisatoriska faktorer innefattar organisations- och maktstrukturen som omfattar projektet. Det politiska stödet kan för vissa projekt vara större än för andra, det vill säga ett stöd från högre chefer i organisationen vilket ökar risken för project escalation. Detta hänger ihop med det faktum att vissa projekt har en förmåga att bli institutionaliserade. Om projektets mål och värderingar stämmer överens och vävs ihop med organisationens mål och värdering ökar risken för institutionalisering och därmed project escalation.

2.3 Förklarande teorier (Keil 1995a)

Keil beskriver och jämför tre teorier i syfte att förklara hur och varför project escalation uppstår. En sammanfattning av teorierna, antaganden och begränsningar, finns i tabellform i bilaga 2.

Self-justification theory

Teorin påvisar att individer har en tendens att trappa upp deras engagemang i en viss handlingsplan (course of action) i syfte att försvara en tidigare gjord handling. Detta trots att man utsätter sig för risken för ytterligare negativa resultat. Trots vetskapen om negativ information i ett projekt satsar den med ett personligt ansvar ytterligare resurser i syfte att åtgärda tidigare begångna fel. En person utan ett personligt ansvar har större förmåga att handla rationellt i ett sådant läge. Den här uppdelningen går också att se mellan den som tidigare har satsat resurser på det aktuella projektet och den som inte har gjort det (Keil et al. 1995).

Prospect theory

Teorin behandlar och utgör ett stöd för förståelse kring den kognitiva snedvridning som påverkar människans beslutsfattande i förhållanden som anses vara osäkra och riskfyllda. På grund av hur ett problem formuleras agerar människan på olika sätt, risktagande eller riskundvikande. Individer uppträder på ett risktagande sätt när ett problem formuleras så att de ställs inför två negativa alternativ. Detta råder speciellt då det finns en initial finansiell förlust (sunk cost, historiskt investerade pengar) och en kombinerad möjlighet att antingen förlora ytterligare eller att få valuta för investerade pengar och hamna på ”noll” igen. Detta kan uttryckas som ”en säker förlust och en möjlighet att vinna tillbaka satsade pengar” eller

”en säker förlust och en möjlighet att förlora ytterligare pengar”. Ett resonemang som det senare alternativet resulterar alltså enligt teorin i ett mer riskbenäget beteende.

Agency theory

Agency theory tar i stort upp problematiken kring relationen mellan principalen (den som delegerar arbete, uppdragsgivare) och agenten (den som utför arbetet, uppdragstagare). Problem uppstår vid målkonflikter mellan de två aktörerna och om det samtidigt finns svårigheter med övervakning av uppdragstagaren. Principalen vill övervaka för att säkerställa att agenten sköter sina åtaganden på rätt sätt och att målen uppfylls. Informationen i en organisation kan vara publik på det sättet att principalen vet lika mycket som agenten. I detta fall måste agenten omdirigera eller avbryta ett projekt som håller på att misslyckas eftersom även principalen vet om projektstatusen. I ett annat läge kan agenten ha mer information än principalen och det är då en målkonflikt kan uppstå. Ett bristfälligt projekt (bristfälligt utan principalens vetskap) som egentligen borde avbrytas kan fortgå p.g.a. att det ligger i agentens egenintresse att projektet i slutändan kommer att betraktas som lyckat.

2.4 Angreppssätt mot project escalation

Keil (1995a, 1995b) tar upp flertalet angreppssätt som utgår dels ifrån uppdelning av faktorer som orsakar escalation och dels ifrån de tre teorierna self-justification-, prospect- och agency theory. Projektstyrning av IT-projekt har ärvt mycket från den traditionella (s.k. rationella) projektstyrningsmodellen och dess verktyg och metoder. Man har förbisett systemutvecklingens speciella läggning och i synnerhet en sådan allvarlig sak som project escalation (Keil 1995a).

Utgångspunkt i påverkande faktorer (Keil 1995b)

Projektstyrningsmetodik för IT-projekt kräver en bredare syn på styrning och bör enligt Keil influeras av dels ett psykologiskt- och dels ett beteendevetenskapligt perspektiv. Dels måste åtgärder ske på det personliga planet hos projektledare och dels organisatoriskt dvs. en implementering av ett handlingssätt och policys i syfte att reducera risken för project escalation. Som ledare i ett projekt bör man alltid rannsaka sig själv för att hitta sådana tendenser. Det är en hårfin skillnad mellan en ”optimistisk/kan göra attityd” och en ”överengagerad attityd”. Det finns ett antal frågor man som ledare kan ställa sig för att få en indikation på om man är i riskzonen eller ej .

- Jag kan inte avgöra vad som skulle betyda ett misslyckande för projektet.
- Min syn på vad som orsakar att projektet lyckas eller ej har ändrats under projektets gång.
- Jag har svårt att lyssna på andras tankar om projektet.
- Jag är mer oroad för projektet än för organisationen som helhet.

Om man instämmer i något av de här uttrycken är det en indikation på att man är alltför engagerad i projektet. Man bör därför återigen utvärdera projektmålen och i allmänhet se över projektet igen innan man satsar mer resurser på det.

För att undvika att man ens hamnar i en situation där man är överengagerad i projektet bör man kontinuerligt göra en utvärdering utifrån ett objektiva perspektiv. Man kan kontinuerligt ställa sig frågan:

□ ***Om jag tog över det här projektet idag, skulle jag fortsätta med det eller avsluta det?***

Man bör hela tiden ha i åtanke alternativa sätt som projektet skulle kunna utveckla sig som skiljer sig från dagens inslagna väg och överväga dessa. Mycket av det här resonemanget förutsätter dock att den organisatoriska uppbyggnaden stöder ett sådant självkritiskt beteende. Utan rätt struktur och stimulans i organisationen skapas ingen sådan miljö. Det finns dock ett antal tekniker eller angreppssätt som tvingar in ledare i ett självkritiskt beteende.

En viktig faktor är att under hela projekttiden veta och låta andra veta i vilken fas av projektet man befinner sig i. Detta bör organisationen skapa system för. Det finns tydliga skillnader i hur man bör styra projekt beroende på i vilken fas man är i. Till exempel kan ett projekt som innefattar ny informationsteknik, som man måste lära sig innan man kan fortsätta, befinna sig i en forskningsfas under en tid. När man sedan går ur forskningsfasen och in i fasen då utveckling sker ändras förutsättningarna och en annan typ av styrning behövs.

Riskhantering skapar ytterligare förutsättningar för att motverka överengagemang och project escalation. Projektrisker bör utvärderas ständigt och så tidigt som möjligt skall man börja ställa sig frågan om det finns några ”redflags” dvs. faktorer som allvarligt hotar projektresultatet. En riskanalys bör hantera följande problem:

- Sannolikheten att man lyckas tekniskt
- Sannolikheten att kunden accepterar leveransen

Två aspekter kan granskas vid en bedömning av den tekniska utgången, hur mogen är den teknik som används och vilka kunskaper har projektgruppen att hantera den? Det är mycket viktigt att utvärdera användarunderlaget för det som skall utvecklas. En aspekt är att undersöka hur stort behovet är vilket skall göras noggrant innan för mycket resurser har lagts ned. Är behovet tillräckligt stort ökar chansen att kunden accepterar det som levereras. Om det finns ett organisatoriskt incitament att använda systemet är också en viktig faktor i riskanalysen.

Vad man ytterligare kan göra är att kontinuerligt utsätta projektet för en kritisk granskning. Granskningen bör göras av en person som svarar för organisationens intressen och skall inte vara en intressent i projektet.

Utgångspunkt i self-justification-, prospect- och agency theory (Keil 1995a)

Med utgångspunkt i self-justification-, prospect-, och agency theory tar Keil upp fler angreppssätt. Self-justification theory handlar om det personliga behovet av rättfärdigande vilket måste dämpas för att förhindra överengagemang. Ett sätt är att utvärdera

beslutsprocessen istället för utgången av ett visst beslut. Det handlar om att minska det personliga ansvaret för beslutsfattaren och istället fokusera på processen. Ett annat sätt är att minska konsekvenser som associeras med misstag. I många organisationer kan det ha en avgörande effekt på din karriär eller ditt fortsatta arbete om du begår ett misstag. I organisationer där man på ett annat sätt tolererar att personalen gör vissa misstag minskar risken för project escalation. Ytterligare ett angreppssätt har att göra med det faktum att risken för ett irrationellt beslutsfattande ökar om samma person hela tiden tar besluten om fortsatta resursallokeringar. Vad man bör göra är att dela upp ansvaret och låta fler ta beslut om fortsatta investeringar så att samma person inte gör detta hela tiden.

Med utgångspunkt i prospect theory handlar angreppssätten om att tränas i beslutsfattande. Projektledare bör upplysas om dels de kognitiva aspekter som påverkar ett beslutsfattande och dels om företagsekonomiska principer som bl. a. ”sunc-cost”, historiska kostnader som inte påverkar ett finansiellt beslut vid utvärdering av olika alternativ. Vidare handlar det om att få beslutsfattaren att implementera ett arbetssätt där man i beslutsituationer kan identifiera och analysera alternativ till den redan inslagna vägen. Framförallt bör ekonomiska kostnader för alternativa vägar tas fram för att fungera som vägledning i beslutsfattandet.

Project escalation som förklaras av agency theory motarbetas genom att förbättra informationssystem så att högre chefer lättare kan följa projekt och dess status. Hela agent/principalproblemet försvinner om alla högre chefer har tillgång till all information som berör projektet. Liksom ovan med utgångspunkt i self-justification theory minskar risken för eskalering om konsekvenstänkandet som associeras med misstag kan minskas.

Theory	Rekommendationer för att undvika eskalering
Self-Justification Theory	<ul style="list-style-type: none"> • Utvärdera beslutsprocesser mer än resultat av beslut • Reducera konsekvenserna som associeras med misstag • Separera initiala och påföljande beslut som rör en viss handlingsplan
Prospect Theory	<ul style="list-style-type: none"> • Träna beslutsfattare att handskas med kognitiv snedvridning som påverkar beslutsfattande • Ignorera sunc-cost vid beslut om resursallokering • Tänk över alternativa handlingsvägar
Agency Theory	<ul style="list-style-type: none"> • Förse högre chefer med information om arbetet hos de underordnade • Reducera konsekvenserna som associeras med misstag

Tabell 1. Rekommendationer för att undvika eskalering baserat på tre teorier (Omarbetad efter Keil 1995a s. 25)

2.5 Learning failure

Lyytinen och Robey (1999) ger en annan förklaring till runaway projects. IT-organisationer har en oförmåga att lära sig av sina egna misstag och därför fortsätter många projekt att prestera dåligt utan att någonting görs. Man har misslyckats med att lära sig effektiva sätt att lösa problem med projektrisker och växande projekt. I många fall har dessa ineffektiva metoder pågått så länge att organisationer har blivit obenägna att förändras. I en miljö där man hela tiden har stora problem och presterar undermåligt blir detta tillstånd till slut ”det normala”. Organisationen har då lärt sig att misslyckas. Learning failure betyder alltså två saker:

- Organisationen har misslyckats att lära sig.
- Organisationen har lärt sig att misslyckas.

Många studier kring problem i IT-projekt utmynnar enligt Lyytinen och Robey ofta i rapporter som presenterar lösningar på problemen med hjälp tekniker som riskhantering, högre grad av användarmedverkan etc. Trots detta är det få organisationer som verkligen ändrar deras sätt att arbeta vilket tyder på att teknik och verktyg i sig räcker inte för att organisationen skall lära sig. Den lärande organisationen hämtar sin kunskap både externt och internt. För många saknas intern kunskap vilket leder till att den externa är det enda alternativet. Dock är det så att extern kunskap har utvecklats i andra organisationer och är sällan applicerbar fullt ut i den egna. Ofta tar man till sig kunskap som stämmer väl överens med organisationens egna värderingar och den ger sällan någon ökad konkurrenskraft p.g.a. att den även finns tillgänglig för andra organisationer. Dock gör tillgången på extern kunskap i IT-sammanhang att man t. ex. kan hålla jämna steg med den teknologiska utvecklingen. Intern kunskap är oftast mer relevant för den egna organisationen just på grund av att den är internt genererad. Det finns en bred tillämpning av hur intern kunskap genereras, allt ifrån informella möten i korridoren till standardiserade formulär som utvärderas och sammanställs. Användarteorier (theories in use) är enligt Lyytinen och Robey alla de kausala samband mellan handling och verkan i IT-utveckling. Användarteorier är alla de instrument som används för att nå fram till det önskade resultatet t. ex. objektorienterad design, prototyping, rollbesättning etc. Detta kan sägas utgöra en modell för vad man inom en organisation använder sig av i arbetet och har tillit till. Användarteorier skapas genom intern och extern kunskap och ändras genom att nya idéer kommer in i organisationen. De är ofta lätta att ta till sig och använda men varierar ofta mellan olika projekt. Lyytinen och Robey menar att varje projekt som genomförs i en organisation bör fungera som ett experiment där användarteorierna testas och justeras. Genom det anpassas handlingar och användarteorier efter den interna kunskap som genereras varje gång ett projekt genomförs och organisationen lär sig hela tiden.

Framförallt finns det fyra hinder för effektiv inläring i IT-organisationer (Lyytinen & Robey 1999):

Begränsad organisationsintelligens	Organisationsstruktur
Hinder för inläring	Hinder i utbildning

Tabell 2. Hinder för effektiv inläring i IT-organisationer

En organisations möjlighet att lära sig begränsas (begränsad organisationsintelligens) av framförallt fyra saker:

- All inläring påverkas av begränsningar i att behandla information och att lära sig av erfarenheter (make sense of experience). Informationsöverflöd är t. ex. ett hinder för organisationer att lära sig, det finns alltför mycket att lära och alltför mycket information att bearbeta.

- Hög omsättning i IT-organisationer gör att mycket av kunskapen inte tas om hand. Det kan till och med vara svårt att förmedla kunskap mellan två projekt p.g.a. tidspress eller de speciella förhållande som råder.
- Personalen förblindas av redan etablerade institutionella tanke- och handlingsmönster. Den ”formande kontexten” utgör en guidning över olika handlings- och beteendemönster som är ”standard” i organisationen och blockerar personalens försök att bilda sig nya kognitiva strategier.
- Organisationens användarteorier saknar validitet på grund av avsaknaden av testning och utvärdering. Den lärande organisationen bör hela tiden utföra experimentella tester och utforska data i syfte att förbättra arbetssätten. Ofta saknas kunskap i organisationer om hur man ökar den vetenskapliga validiteten i lärandeprocessen.

Ett betydande hinder för inläring är fixeringen vid att man måste lyckas. Organisationer har många incitament för att lyckas men inte för att göra misstag och information kring de misstag som görs försvinner ofta fort ur organisationens minne i rädsla för att göra samma misstag igen. Information som skulle kunna användas i syfte att testa rådande användarteorier går på så sätt förlorade. När ändå kunskap genereras i samband med misstag tycks den användas i defensivt syfte, det vill säga i ett undvikande syfte. Ett defensivt beteende leder dock sällan till ett lärande i positiv och utvecklande mening i det långa loppet.

Den organisatoriska uppbyggnaden kan också utgöra ett hinder för lärande. Barriärer mellan avdelningar gör att användbar information kan stanna inom en begränsad yta inom organisationen. Användarteorier blir på det sättet betydligt färre för alla och chansen att kunna utvärdera dem på ett bra sätt minskar. Traditionellt sett har IT-avdelningar hamnat utanför resten av organisationen vilket ofta leder till att bara ”tekniska” användarteorier används vid systemutveckling. Ett sätt att minska nyttan med lärande är att lyfta ut ”lärandekunskap” och separera den från kärnverksamheten vilken sker när man t. ex. lyfter ut och skapar en speciell IT-avdelning. Lärandeprocessen skall byggas in i verksamheten för att få bästa effekt.

Utbildning till systemutvecklare karaktäriseras av speciell teknisk utbildning. Det har länge varit förknippat med ”datorutbildning” vilket har gjort systemutvecklare till tekniker. Tekniker har ett snävare synfält som passar dåligt in i den typen av systemtänkande man har i en affärsverksamhet.

Lyytinen och Robey påvisar att en organisation som hela tiden misslyckas i lärandeprocessen får till slut svårt att identifiera alternativ och satsa på dem. Rådande fundament och handlingsmönster blir till slut användarmyter och prestationsbrister skylls på externa faktorer istället för på den egna processen. Ofta inträder ett defensivt beteende där organisationen ställer sig fientlig emot nya idéer eller utvärdering av rådande användarteorier. Enligt Lyytinen och Robey finns det framförallt tre myter som råder i organisationer i en sådan situation vilket förhindrar inläring.

- Den teknologiska fixeringen betyder att allt går att lösa med mer och/eller bättre teknik. Ett annat drag är att envist vidhålla de effektiva och användbara med vissa utvecklingsmetoder eller verktyg trots att erfarenheten av den/de säger något annat.

- Organisationsmyten betyder att allt hopp läggs i organisationsstrukturen och att i den finns en lösning på aktuella problem. En omorganisering genom t. ex. outsourcing eller teamarbete är ett vanligt sätt att tackla problem men som kanske inte är den bästa metoden.
- Myten om silverkulan (silverbullit) är den tredje vanliga myten och benämner en omotiverad tro på att det i framtiden (nära förstående) kommer en kraftfull lösning på alla IT-problem och ”får alla misstag ojordra”.

2.6 Projektstyrningsmodeller i systemvetenskaplig litteratur

De projektstyrningsmodeller som har studerats och presenteras här har hämtats uteslutande från systemvetenskaplig litteratur dvs. ej från artiklar eller rapporter av något slag. Valet av litteratur har skett relativt godtyckligt av författaren med viss hjälp av handledare från institutionen.

Ian Sommerville tar upp projektstyrning av systemutvecklingsprojekt i den 6:e upplagan av Software Engineering (2001). Många av de misslyckade projekt som har genomförts har misslyckats p.g.a. dålig styrning. De styrningstekniker som har använts har främst hämtats från andra ingenjörsgrenar vilka har fungerat dåligt för att styra systemutvecklingsprojekt. De främsta skillnaderna mellan styrning av systemutveckling och annan utveckling är:

- Till skillnad från andra byggnationer som t. ex. skeppsbygge är produkten ogripbar vilket gör det svårt att följa utvecklingsarbetet. Chefer får lita på den dokumentation som produceras för att följa utvecklingsprocessen.
- Det finns ingen standardiserad utvecklingsprocess som det finns i andra ingenjörsgrenar. Systemutveckling är en ny företeelse och det krävs många år innan man har testat och utvecklat bra utvecklingsmodeller.
- Stora systemprojekt är oftast engångsföreteelser och skiljer sig åt från gång till gång vilket gör det svårt att överföra erfarenhet från ett projekt till ett annat.

Projektstyrning är ett stort ämne och lämpligheten bestäms ofta av projektspecifika faktorer som organisation runt projektet och produkten som skall utvecklas. Enligt Sommerville går det gå dock att utkristallisera tre viktiga projektstyrningsaktiviteter nämligen projektplanering, schemaläggning och riskhantering.

Planeringsprocessen tar sin utgångspunkt i projektets inramande begränsningar, alltså leveransdatum, budget, personaltillgång etc. tillsammans med andra projektfaktorer som storlek och rollbesättning. Processen är iterativ och pågår ända tills projektet är avslutat. Ny information blir tillgänglig hela tiden och då måste planen omvärderas. Projektets leverabler och milstolpar definieras och sätts in i planen innan planeringsprocessen går in i den iterativa delen. I den iterativa delen sker schemaläggning och utefter den initieras de aktiviteter som skall göras. Efter att projektet har pågått en tid granskas processen och projektets uppskattade ramfaktorer granskas och revideras. Schemat uppdateras och projektets begränsade faktorer

försöker att utökas. I den iterativa delen sker också eventuell omförhandling av leverabler. För varje iteration sker en utvärdering av projektet i syfte att hitta brister som kan leda till försening eller att mål inte uppfylls.

Milstolpar utgör en slutpunkt för en aktivitet och stakar ut vägen till målet. Vid varje milstolpe bör det ske någon form av formell output (leverabler) t. ex. en rapport som beskriver var man är i processen samt om det finns några problem.

Schemaläggning innebär att projektet som helhet delas upp i aktiviteter som läggs i en sammanhängande ordning. Aktiviteternas tids- och resursåtgång uppskattas och ordnas så att resurser kan utnyttjas på ett effektivt sätt. Man måste identifiera kritiska aktiviteter dvs. sådana som andra aktiviteter är beroende av och lätt kan fördröja hela projektet. Tidsuppskattning är mycket svårt och det är ofta som för lite tid avsätts för aktiviteter. Ett sätt att uppskatta tidsåtgång är att först se hur lång tid en aktivitet skulle ta om ingenting går fel och sedan lägga på tid för att lösa uppkomna problem. Vid schemaläggning används stapeldiagram och aktivitetsnätverk som hjälp för att visualisera beroenden och turordning. I ett aktivitetsnätverk visualiseras alla aktiviteter i ett projekt och ställs i tids- och beroenderelation med andra aktiviteter. Genom att uppskatta aktiviteter varaktighet kan man dels räkna ut hela projektets längd samt se var den kritiska linjen är. Den kritiska linjen utmärker de på varandra påföljande aktiviteter som tar längst tid att utföra i projektet och utmärker då alltså hela projektets längd. Visualiserande aktivitetsmodeller kan kompletteras med tillgängliga resurser vilket gör det lättare att allokeras ut dessa på aktiviteter.

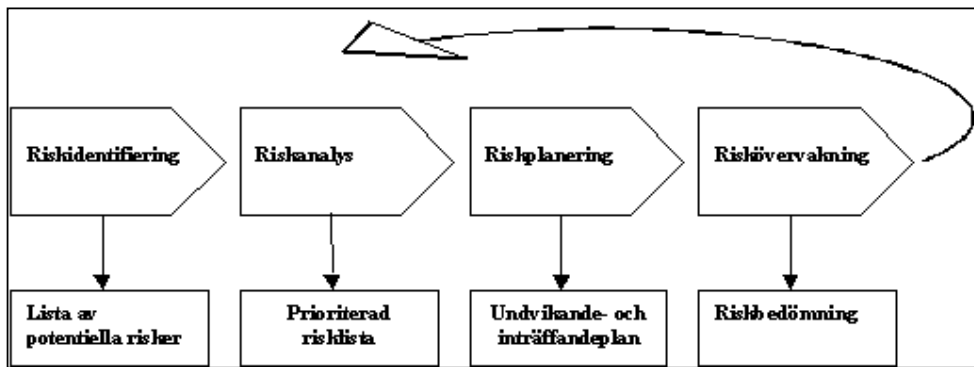
Riskhantering är en mycket viktig uppgift för projektledningen och innefattar att förutse, motarbeta uppkomsten och minimera effekter av risker. Riskanalysen skall dokumenteras och skall stå med i projektplanen. Det går att utkristallisera tre kategorier av risker och vad de påverkar:

- Projektrisker påverkar schema eller resurser.
- Produktrisker påverkar produktens kvalitet eller prestanda.
- Verksamhetsrisker påverkar organisationen som utvecklar produkten.

Riskanalys är en iterativ process som pågår under hela projektet och innehåller flera steg:

- Riskidentifiering: Identifiera de risker som finns.
- Riskanalys: Bedöm möjligt inträffande och konsekvens av ett inträffande.
- Riskplanering: Planer för att undvika eller minimera effekten.
- Riskövervakning: En ständig utvärdering av risk och omarbetade riskplaner.

Varje steg i den iterativa processen resulterar i en output vilket visas av figur 2.



Figur 2: Riskhanteringsmodell (Omarbetad efter Sommerville 2001 s. 85)

Royce (1998) förespråkar en modell där framförallt fem olika steg utmärker modern styrning av systemutvecklingsprojekt.

- Utvecklingsprocessen skall utgå ifrån arkitekturen, dvs. design och integrering, till skillnad från äldre angreppssätt där man utgick ifrån kraven. Det kräver en noggrann balansering mellan driftskrav, arkitektur och projektets livscykelplan innan resurser kan allokeras för fullskalig utveckling.
- Etablera en iterativ "livscykelprocess" där riskkonfrontation sker tidigt. Med dagens stora och sofistikerade IT-system är det inte möjligt att definiera, designa och utveckla hela lösningen i en sekvens. Man måste genomgå flera iterationer där problem och lösningar utvecklas eftersom. Genom det blir det också lättare att balansera alla inblandades intressen i projektet. Riskhantering skall etableras så tidigt som möjligt för att undvika stora problem längre fram i projektet.
- Övergå till designmetoder som stödjer en komponentbaserad utveckling. Komponenter är stycken med käll- eller exekverbar kod med ett bestämt beteende och interface. "Komponenttänkande" måste etableras för att reducera mängden kod som måste genereras av människor samt att komma bort ifrån specialbeställda "smala" lösningar.
- Skapa en miljö som kan hantera förändringar. Dynamiken kring iterativ utveckling framtvingar en kraftfull objektiv styrning.
- Utöka möjligheten för användningen av förändringsverktyg som stödjer "Round-Trip engineering". Round-Trip engineering är det administrativa stöd som krävs för att automatisera och synkronisera information i olika format (kravspecifikationer, modeller, källkod etc.). Utan detta grundläggande stöd är det mycket svårt att ge klara riktlinjer där förändring uppmuntras snarare än undviks.

Royce har identifierat tio av de vanligaste projektrisker som förekommer i den traditionella (äldre) utvecklingsmodellen. Den ovan angivna projektstyrningsmodellen innehåller naturliga verktyg som hanterar alla de traditionella riskerna.

- ♦ Upptäckt av fel i den senare delen av projekt vilket leder till mycket omarbete

- ◆ Utslitning av nyckelpersoner
- ◆ Bristfälliga utvecklingsresurser
- ◆ Motarbetande intressenter
- ◆ Tvingande inpassning av teknologin
- ◆ Utdragen kravinsamling
- ◆ Analytiskt stillestånd
- ◆ Bristfällig prestation
- ◆ Överbetoning på artefakter
- ◆ Bristfällig funktionalitet

McConnell (1998) presenterar ”steg för steg” anvisningar för projektstyrning som är mycket lätta att följa. Varje kapitel avslutas med ett överlevnadstest där man kan gå igenom ett antal aktiviteter eller krav som skall vara gjorda eller uppfyllda för att projektet skall lyckas. Det rekommenderas att följa en viss projektprocess (”Staged delivery”) där projektet går igenom ett antal etapper där det i varje etapp skall ske ett antal aktiviteter. En viktig aktivitet är att dokumentera vad som händer i varje etapp. Dokumenten läggs till projekthistoriken vilken skall kunna användas efteråt att spåra allt som har inträffat under projektiden. Den första etappen innefattar ett utvecklande av systemkonceptet vilket är ett klarläggande av bl. a. projektvision och vilken finans- och maktbild som omfattar projektet. Olika planer skall vara utformade t. ex. riskhanteringsplaner. I nästa etapp utvecklas systemkraven genom användarintervjuer och prototyping. Överlevnadstestet säger bl. a. att projektteamet måste ha identifierat nyckelanvändare som skall ingå utvecklingsarbetet. Testet varnar oss också för t. ex. ett för tidigt avslutande av prototyping innan användarna är nöjda.

McConnell skriver att i designfasen, som är iterativ, skall varje iteration behandlas som ett projekt i projektet. I varje iteration genomförs planering, design, konstruktion, test och leverans av mindre delar av systemet. Eftersom mindre projekt inte är lika utsatta för risker delas alltså hela projektet in i delprojekt.

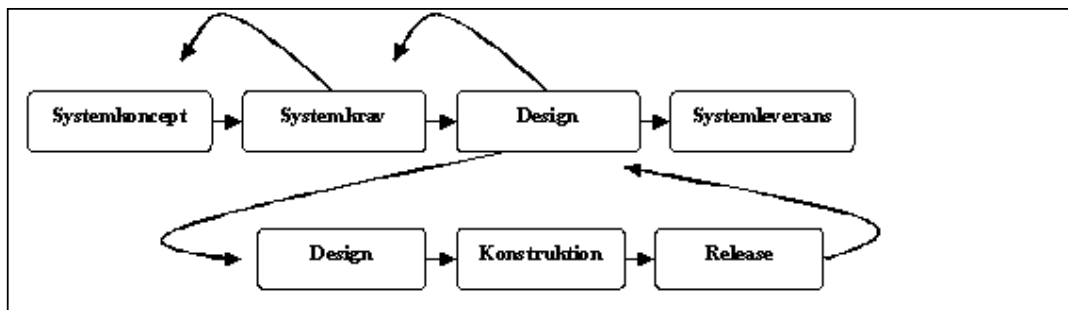
I planeringsfasen genomförs en övergripande planering för hela fasen vilket bl. a. innefattar uppdatering av systemkrav, detaljerad design, konstruktion (kodning), testutformning etc. Fasens milstolpar skall också fastställas (miniatyrmilstolpar för att de är mindre än vanliga milstolpar) vilket skall resultera i en komplett lista. Om milstolpar missas frekvent bör projektledaren omkalkylera utvecklingstiden och flytta fram deadline. Andra åtgärder är att sänka systemkraven, rensa arbetsmiljön så att utvecklarna kan fokusera lättare eller omfördela arbete.

I konstruktionsfasen sker kodning. En viktig aspekt är kvaliteten vilket säkerställs genom en kodningsstandard. Syftet med en standard är att få hela systemet att se ut som det är kodat av samma person vilket är till stor hjälp när t. ex. utvecklare skall läsa varandras kod. Integrationsproceduren är till för att föra in nyligen utvecklad funktionalitet in i huvudsystemet. Att kontrollera vilka systemändringar som behöver göras sker under denna fas då användare börjar sätta sig in i de delar som är levererade.

Testfasen sker samtidigt som konstruktionsfasen. Koden testas för att hitta fel som utvecklare kan åtgärda. Testning utförs hela tiden för att hålla uppe en hög kvalitet och till skillnad från vissa andra projektmodeller kommer inte ett stort test i slutet innan leverans där alla fel upptäcks.

Att hela tiden tvingas att konstruera systemdelar som skall levereras omgående minskar risken för undermålig kvalitet. I releasefasen går hela produkten som skall levereras igenom så att den håller den höga kvalitet som är förknippat med leverans. En checklista med 21 punkter går igenom och kontrolleras.

Fasavslutning innebär en möjlighet att styra om handlingsplanen och lära sig av det som hittills har skett. Detta är ett bra läge att samla ihop och undersöka de förändringsförslag som finns samt att gå igenom de uppskattningar som har gjorts. Projektfaktorer som t. ex. omfattning, produktivitet och prestandakrav går igenom för att se om förutsättningarna har ändrats. Det är också viktigt att utvärdera det som har presterats kontra projektplanen för att bekräfta att allt sker som planerat.



Figur 3. Modell över "Staged delivery" (Omarbetad efter McConnell 1998 s.163)

3 Material och Metod

3.1 Datainsamling

Vid sidan om teoristudien har en empirisk studie skett under perioden. Den avser de förhållanden och den samlade syn på projektstyrning som finns inom utvecklingsgruppen med avseende på runaway projects. Insamling av data på Försäkringskassan har främst skett genom observationer och intervjuer. Observationer har genomförts kontinuerligt under hela studietiden och har skett på gruppmöten och kundmöten. Fyra intervjuer har genomförts vilka i snitt har tagit ca 40 minuter. Tre av de intervjuade ingår som utvecklare i gruppen och en intervju har skett med chefen för regionens IT-avdelning. Halvstrukturerade intervjufrågor har använts och intervjuerna har till stor del haft karaktären av en diskussion (se bilaga 1 för intervjufrågor). Vid sidan av de formella intervjuerna har ett flertal mindre informella intervjuer ägt rum främst med personer i utvecklingsgruppen.

3.2 Intervjuer

Easterby-Smith, Thorpe och Lowe (1991) påvisar att innan metod för materialinsamling väljs måste studiens mål vara fastställt. Detta gäller även intervjumetoden trots att den av många anses vara den ”bästa” metoden. Det finns ett brett spektra av hur det i praktiken går till, från frågeformulär med exakta svarsalternativ till rena diskussionsforum där mycket lite är styrt på förhand. I den här studien har halvstrukturerade intervjuer tillämpats (de formella intervjuerna, 4st.) dvs. det är relativt öppet, inom ämnesramen, för den intervjuade att svara på frågan. Några svarsalternativ ges inte. Intervjuaren har i många fall ställt följdfrågor och bett den intervjuade att precisera eller utveckla sitt svar. Intervjufrågor har utformats med utgångspunkt i de teorier som används i studien.

Halv- eller ostrukturerade intervjuer är lämpliga när det är viktigt att förstå den intervjuades verklighetskonstruktion. Dennes tankar och åsikter som utgör en grund för det ämne som skall diskuteras. Vidare är det en lämplig metod vid tillfällen då forskaren måste tränga in och förstå den intervjuades ”värld” för att kunna påverka (Easterby-Smith et al. 1991). Det finns klara svårigheter med att utföra denna typen av intervjuer vilket mer eller mindre påverkar resultatet. Till och med det mest elementära som att förstå de svar som ges kan vara svårt. Det gäller att kunna se svaren ur respondentens synvinkel samt att tolka dem i dess kontext. Det är också viktigt att vara medveten om den sociala interaktion som sker mellan den som intervjuar respektive den som intervjuas. Samtalet kan styras, frågor och svar kan tolkas genom den sociala påverkan människor har på varandra genom kläder, tonval, gester, kroppshållning etc. (Easterby-Smith et al. 1991).

3.3 Observerande deltagare

Metoden med observerande deltagare har sina rötter i etnografisk forskning. Forskare levde i avlägsna byar och studerade främmande stamkulturer. Eftersom alla organisationer har en egen kultur liksom avlägsna stamkulturer har sin egna kultur (sociala spelregler etc.) är det ett applicerbart sätt att studera organisationer. Det är viktigt för observeraren är vara medveten om vilken roll han spelar och det finns flera olika rollbenämningar. En sådan rollkategorisering är researcher as employee, research as explicit role, interrupted involvement och observation alone (Easterby-Smith et al. 1991).

Research as the explicit role beskriver bäst den typ av roll som har använts under forskningsarbetet som ligger till grund för den här rapporten. Den observerandes roll som forskare görs klart för alla anställda på arbetsplatsen vilket öppnar för möjligheter att röra sig inom organisationen och få tillgång till information. Forskaren befinner sig i organisationen varje dag under en period och kan observera, intervjuar, medverka i arbetet etc. (Easterby-Smith et al. 1991).

3.4 Metoddiskussion

Studiens mål är att undersöka och beskriva fenomenet runaway projects i dess verkliga miljö samt att föra en diskussion kring vilket hanteringsstöd som ges i olika projektstyrningsmodeller. Den etnografiska metoden med observationer och intervjuer har valts för att kunna komma innanför "skalet" på den studerande organisationen. Det är överhuvudtaget svårt att få personer att blotta problem och brister i sitt arbete. Man måste ha en möjlighet att kunna kontrollera den information som ges. Risken är annars att det blir en form av försvarsjargong och man kommer aldrig åt själva kärnan i problematiken. I detta sammanhang är det viktigt att studera hur man gör, inte hur man säger att man gör. Befinner man sig som forskare på en arbetsplats och under en period dagligen har tillgång till de personer som berörs är det lättare att skapa förutsättningar för det.

Runaway projects är ett komplext fenomen och många beskrivande och förklarande faktorer kan inte lyftas ur sitt sammanhang för att tolkas. Organisation vari studien genomförs har sin egen kultur vilken måste studeras och tas hänsyn till. Man måste få med sig hela bilden av det som skall undersöka för att kunna dra trovärdiga, och framförallt inte dra förhastade, slutsatser.

Nackdelen med metoden är att den är relativt svår att tillämpa. En klar svårhet, vilket till viss del förklaras av oerfarenhet, har varit att exakt få ut det resultat ur intervjuerna som var tänkt. Som intervjuare har man en bild av vilka aspekter som ämnet innefattar och vad som bör behandlas och den intervjuade har en bild vilka inte alltid stämmer överens.

4 Resultatredovisning

Resultatet av intervjuerna och studierna redovisas i en viss ordning som framförallt bestäms av den struktur som finns på intervjufrågorna. Vissa frågor är hopslagna och bildar en rubrik med lite modifikation efter vad som bedöms höra ihop. Resultatet är till största delen omformulerat för att passa en naturligare form av redovisning. I vissa fall sker en koncentration på samstämmighet mellan resultatet från de olika intervjuerna samt från de observationer som gjorts och i vissa fall sker en koncentration på olikheter. De intervjuade/observerade har haft möjlighet att läsa igenom resultatet innan publikation i syfte att påvisa uppenbara missförstånd och felaktigheter. Resultatet presenteras i följande ordning:

Resultat av etnografisk studie:

- En kort genomgång över hur man idag jobbar med projektstyrning i gruppen
- Två projektbeskrivningar samt en diskussion kring erfarenhet av runaway projects
- Tillgänglighet av projektinformation
- Personligt ansvar för projekt och projektresultat
- Metoder för återkoppling och lärande
- Diskussion kring stöd för hantering i projektstyrningsmodellen

4.1 Projektstyrning i utvecklingsgruppen

Den främsta uppgiften som gruppen har är att tillfredsställa behov av processtöd som uppstår i verksamheten. I många fall, men inte alla, resulterar behovet i att gruppen utvecklar IT-stöd. En förstudie sker alltid då det finns oklarheter kring de behov som har uppkommit och när man inte har ett fullständigt underlag för en beställning. Vid en förfrågan där ett sådant underlag finns sker en utvärdering av användarnytan med ett eventuellt IT-stöd samt vilka kostnader ett utvecklingsprojekt alternativt ett inköp av en sådan produkt blir. Det vanliga är dock att det krävs en förstudie vilket sker enligt en iterativ metod där behoven identifieras och förfinas i varje iteration parallellt med att kraven växer fram. Den iterativa metoden innehåller tre steg:

- Behovsanalys syftar till att höja medvetandenivån kring de behov som har gett upphov till förstudien vilket sker genom intervjuer och arbetsplatsanalyser. Samtidigt försöker man identifiera och definiera projektets aktörer och intressenter. Aktörer är direkt inblandade i projektet medan intressenter endast berörs av projektet.
- Prototyping används för att ytterligare specificera behoven vilket skall utgöra en grund för de systemkrav som skall tas fram. Olika scenarier byggs upp runt en prototyp för att kunna kartlägga alla tänkbara aktiviteter i processen.
- Kravframställning sker i samförstånd mellan beställaren och utvecklaren och ligger till grund för applikationens funktionalitet, tillgänglighet, grafik etc.

När förstudien är klar skall det finnas ett tillräckligt underlag (kravspecifikation) för en utredning om användarnytta och projektkostnad. Om ett beslut sedan tas om utveckling initieras utvecklingsarbetet vid sidan om en kvalitetssäkringsprocess (tidigare certifieringsprocess). Kvalitetssäkringsprocessen innefattar genomgång och dokumentation av följande delar:

- Systemunderlag dvs. kravspecifikation samt beställare och beslutstagare
- Systemdokumentation dvs. teknisk specifikation
- Drifrutiner
- Användaranvisning
- Installationsanvisning
- Säkerhet / behörighet
- Testprotokoll / pilotprotokoll
- Systemplan dvs. information om förvaltningsorganisation och systemägare

Varje del i kvalitetssäkringsprocessen skall gås igenom och dokumenteras. Ett system som saknar delar i processen går inte in i skarpt läge innan det är gjort. Utvecklingen bygger på korta leveranser vilket innebär att projektet snabbt skall leverera en första version som kan börja användas så tidigt som möjligt. Efter en första systemleverans förfinas produkten fram till slutleverans av systemet. Tanken med korta leveranser bygger på idén att snabbt skapa nytta i verksamheten samt att få användare att utveckla förbättringsförslag vilket är lättare när det finns en version att jobba i. Vid början av varje utvecklingsprojekt skall en förvaltningsorganisation utformas. Förvaltningsorganisationen tar över systemet efter leverans och ansvarar därefter för följande områden:

- Den tekniska plattformen där systemet finns vilket innebär underhåll och prestandasäkring på hårdvaran samt server- och databasadministration.
- Kravinsamling från användare innebär att systematiskt samla in krav på systemförändringar som uppkommer under systemets livstid och förmedla dessa till utvecklingsgruppen. Mindre krav åtgärdas omgående vilket också gäller systemfel. Vid andra tillfällen måste en helt ny process startas vilket då innebär att alla steg går igenom igen samt att nya beslut måste tas om ett nytt utvecklingsprojekt.
- Förvaltningen ansvarar också för att eventuella ändringar i förutsättningar runt systemet, t. ex. lagändring.

En viktig aspekt i projektstyrningen är att man markant vill åtskilja utvecklingsprocessen och förvaltningsprocessen. Detta gör man med anledning av att markera de nya förutsättningar som råder. Ett överlämnande av systemet markerar att projektets utvecklingsfas är avslutad och nya beställningar i samma system behandlas genom att gå igenom hela processen igen.

4.2 Erfarenhet av Runaway projects

Intervjuresultatet visar att problematiken inte är ny för personalen i utvecklingsgruppen. Mycket av de faktorer som bildar begreppet runaway projects känns igen och det är lätt att associera till tidigare projekt som gruppen har haft.

Fallbeskrivning 1

En av de intervjuade beskriver ett projekt som belyser problematiken kring att kompetensbrister kan få projekt att glida iväg. Utvecklingsgruppen befann sig vid det tillfället precis i början av ett tekniskifte och man hade bestämt sig för att utveckla en relativt stor applikation med webbt teknik dvs. en applikation som exekveras på en webbserver och till vilken åtkomst sker genom webbläsaren. Det tekniska resultatet blev till slut lyckat men projektet hade försenats kraftigt. Det var först efter nästan tre år som applikationen fick den design som var tänkt från början. Det är inte uppskattat hur mycket tid som har lagts ned på projektet men det är betydligt mer än vad som var tänkt. Framförallt konstateras tre brister som kom att ha avgörande betydelse för projektprocessen.

Det första gäller brist på kunskap i den teknik som skulle användas. Mycket tid gick åt att pröva sig fram i den nya tekniken och man lärde sig mycket på relativt kort tid, men ändå blev det inte riktigt bra. Utveckling av version 2 av applikationen är i dagsläget precis avslutat och det är först nu som all funktionalitet finns. Resultatet av det som har utvecklats på senare tid har blivit bättre och det har gått betydligt snabbare att utveckla p.g.a. att man har utvecklat den tekniska kunskapen under tiden.

Det andra gäller brist på kunskap i projektmetodik. Detta fick personalen också (vid sidan om produkttekniken) lära sig under projektets gång. De projektmallar som användes hade utvecklats centralt av RFV vilka man då stödde sig på. I dessa specificerades det mesta som skulle göras i ett projekt, vilka dokument som skulle upprättas, rollbesättning, vilka som skulle hållas ajour med projektets status etc. Det uttrycks som tur att beställaren i detta fallet hade viss IT-kompetens vilket gjorde att kravspecificeringen blev så pass bra som den blev.

Den tredje bristen är projektets avslutande vilket inte hade specificerats tillräckligt tydligt. Överenskommet var att projektet skulle leverera med korta mellanrum för att kravbild till viss del skulle klarna under projektets gång. Mycket av det som skulle levereras levererades också och beställaren var relativt nöjd med hur projektet fortskred. Det var inte bestämt exakt när projektet skulle anses vara avslutat och beställaren fortsatte att ställa krav på att nya funktioner skulle utvecklas och att vissa ändringar av tidigare funktioner skulle ske. Projektgruppen hade varit sysselsatt med detta ganska länge och andra uppdrag började komma in vilka påbörjades och pågick parallellt. Beställaren var inte helt tillfredsställd med systemet som nu fungerade "hyfsat", men projektet tappade fart och blev liggande utan att det var korrekt avslutat.

Fallbeskrivning 2

Förhållanden i ett annat projekt beskriver svårigheten med att stoppa projekt som tycks ha fått ett eget liv. Sammanlagt har en viss applikation i Försäkringskassans verksamhet omkring 4000 användare i flera olika län. Applikationen har levt ett långt liv och en projektgrupp håller ständigt på med utveckling av den. Systemet innefattar en teknik som redan för länge sedan var omodern och beskrivs så här:

...ett system som har levt sitt eget liv som ingen har kontrollerat och ingen vet vad det har kostat att utveckla eller vilken nytta det har gjort och nu sitter vi där med en bomb.

Projektet, som inte ligger på IT-avdelningen i Västra Götaland, har tidigare försökt stoppas av förespråkare ur utvecklingsgruppen och flera beslut har tagits att utvecklingen skall avbrytas vilket inte har skett. Det finns risk att systemet inte kan föras över på den nya plattformen och man har nu sex månader på sig att utveckla ett ersättningssystem. Den största anledningen till att projektet tycks ha fått ett eget liv är att det har skapats en organisation kring det som hela tiden hittar förslag på förbättringar vilka också utvecklas utan att beräkna vilken nytta de ger. Användare beställer av IT-avdelningen utan formella krav på beslut från högre instanser eller beställningsspecifikationer. Detta skapar en situation där man aldrig lyfter upp frågan om man skall utveckla produkten eller göra förbättringar överhuvudtaget. Det blir en ond cirkel och en viss frustration kan anas i uttrycket:

...sådana grupperingar har en egen dynamik som är väldigt svåra att stoppa.

Uppkomna behov hos användare som hela tiden resulterar i nya utvecklingsprojekt utan att ifrågasätta nyttoeffekten skapar en evighetsituation som tar väldigt mycket resurser. Har man väl tillfredsställt ett behov uppstår det alltid nya och det är svårt för personer som inte har ett strategiskt tänkande att avgöra om resurser skall satsas. Varje beställning måste lyftas upp till högre nivåer i organisationen till personer som har en bredare syn på verksamheten. På högre nivåer i organisationen finns bättre förutsättningar att se vilka beroenden som skapas samt att analysera ekonomiska aspekter vilket gör att det blir lättare att prioritera mellan projekt. Ett problem för användare eller utvecklare är att bedöma nyttan med det som efterfrågas. Många handläggare har ingen chans att se till helheten utan ser bara sina egna problem som de naturligtvis vill ha lösta.

Nya metoder ska börja användas för att formalisera beställningsprocessen. RFV har utvecklat metoder för att se på de ekonomiska aspekterna i en prioriteringssituation, hur man gör en ekonomisk uppföljning, hur mycket tid och resurser det tar, vilka effekter det blir av projekt etc. Detta skapar bättre förutsättningar för att hejda projekt som det här.

Övrig erfarenhet

Något som alla tycks se som en orsak till att projekt hela tiden drar ut på tiden utan att avslutas är problematiken kring kravspecifikationer. Det arbetssätt som man har valt att arbeta efter förutsätter att kravbilderna utvecklas under tiden projektet pågår. Det är en medveten strategi att ha det så och en förutsättning för de användare och beställare som ingår. Systemkompetensen är i allmänhet låg hos beställare vilka måste få något konkret att utgå ifrån varefter man utvecklar de krav som skall ställas på applikationen. En risk med denna

iterativa utvecklingsmodell, där man levererar mindre delar i korta intervall, är att mer och mer behov upptäcks längs vägen och utmynnar i ytterligare krav som måste tas hänsyn till. I vissa fall, och det har blivit vanligare den senaste tiden, säger man nej till sådana krav. Vad man istället gör är att senarelägga dem till en annan version. Dock har det funnits, och finns fortfarande, en viss policy att vara så tillmötesgående som möjligt vilket resulterar i en del s.k. korridorbeställningar där man säger ja till utveckling av ytterligare funktioner.

Korridorbeställning kan beskrivas så här:

Efter man har haft ett möte så kommer någon tillbaka och säger: "det är så här vi vill ha det". Detta är svårt att styra, är det mindre grejer som man tycker att det spelar ju ingen roll så slinker sådant här med, alltså större grejer som man inte har sett vidden av som kan ta mycket tid att göra. Tidsplanen för de andra aktiviteterna sätts på spel.

På frågan om varför man tar på sig för mycket svarar en person:

Jag tror att det är en inställning om att man skall försöka hjälpa kunden, servicekänslan, det är lätt att göra missbedömningar, man är för snäll helt enkelt, det gäller att vårda kundrelationen lite grand, det får ju inte bli ett byråkratiskt system som innebär att man inte kan göra någonting utan att dra igång ett helt projekt som är resurskrävande, det där är en avvägning och bedömning som man måste göra varje gång...

Ytterligare en risk med att använda prototyping som utvecklingsmetod är att den första modellen blir ganska komplex dvs. det som man skall utgå ifrån och skapa systemkrav har redan löst mycket av det ursprungliga problemet. Man börjar alltså jobba med ett färdigt system och fortsätter att bygga på mer och mer utan att man ser slutet på projektet. För att motarbeta detta måste den som utvecklar redan från början börja skissa på vad som kan innefattas av en första version samt fånga upp de krav som ligger utanför detta och lägga detta på en senare version. Det är viktigt att kraven samlas upp på ett korrekt sätt så att beställaren känner att man lyssnar och att de ser att det kommer med i nästa version.

Det är svårt att få en tydlig bild över vilka indikationer man kan få på att projekt rullar på utan egentlig kontroll. I intervjuerna tar flera återigen upp diskussionen med kravspecifikationer. En person uttrycker det som en magkänsla som är svår att definiera men som bara infinner sig ibland.

Det är svårt att säga att man har tydliga indikationer på..., man kan ha sina aningar, det känns i magen att det här känns tveksamt om vi får ut den nyttan, och har man då ett väldigt tryck ifrån beställaren att faktiskt ändå göra det här då är det svårt att faktiskt få tid att sätta sig ner och göra en ordentlig analys, det är svårt att vara så motståndskraftig, ofta är det ganska angeläget om att sakerna kommer till stånd, det är lätt att göra saker som är bra att ha men inte nödvändiga.

Rollbesättning har också diskuterats under intervjuerna vilket flera tar upp som en orsak att projekt kan få problem. En aspekt som redan har nämnts tidigare i rapporten handlar om kompetensen hos den som beställer. Ett möte med utvecklingsgruppen resulterar oftast i en mängd ny kunskap om vad som faktiskt är möjligt att göra. Samtidigt är man från början inte helt säker över vilka behoven är och vad man vill tillfredställa. När kunskapen om IT växer under processen växer också kunskapen om vilka behov man har vilket är bra, bara man sätter

en gräns någonstans. En gräns på vilket omfång projektet skall ha måste bestämmas i inledningsfasen. En önskan är att beställaren har mer vetskap om vad den vill ha innan kontakt tas med utvecklingsgruppen.

Ytterligare en intressant faktor, som hänger ihop med kompetens, handlar om att byta motpart under projektets gång. Vid ett pågående projekt gick nya personer in i beställarrollen vilka inte alls hade den kompetensen som de tidigare hade. Projektet pågick ändå som vanligt vilket skapade en svår situation. Framförallt blev det svårt genom att personen i utvecklingsgruppen fortsatte med det språk som hade varit standard hittills under projektiden. Dock hade de nya beställarna inte alls den kompetensen samtidigt som nya krav kom in i bilden och mycket fick göras om och mycket fick gås igenom igen. I efterhand ser man att projektet borde ha startas om på nytt då förändringen skedde men det var svårt att ana att det skulle betyda så mycket. Rollbesättning kan också handla om en del ”mjuka frågor” dvs. hur man är som person och vilka ”mjuka kvaliteter” man har. Det kan t. ex. vara påfrestande för en del att ingå i projekt på grund av arbets sättets natur. Arbete i projektform skiljer sig från vanligt linjearbete och det krävs ofta en stor arbetsinsats.

4.3 Tillgänglighet av projektinformation

Det är en relativt samlad bild av vad gruppen anser om tillgängligheten av projektinformation. I det lilla sammanhanget, inom utvecklingsgruppen samt gruppens närmaste chef, har alla en god möjlighet att ta del av information om projektstatus i de pågående projekten. Gruppen har kontinuerliga veckomöten då varje projekt går igenom i syfte att informera resterande gruppmedlemmar om hur långt man har kommit samt att ta upp om det har uppstått problem som måste lösas. Vid sidan om mötena ligger dokumentation om alla projekt tillgängliga på intranät som ytterligare en informationskanal. Dock finns det kritik mot den form dokumenten har. Dokumenten ligger som vanliga textfiler i olika kataloger vilket inte känns så lättillgängligt som det borde vara. Det blir upp till varje person att själv gå in i dokumenten och söka efter den information som man vill åt. Ett alternativ kunde vara att presentera viss standardinformation om t. ex. var i processen projektet befinner sig så att det gick lättare att ta åt sig, inte bara för utvecklingsgruppen utan även för andra intressenter i projektet. Det kanske viktigaste är ändå att kunden har information om projektet. Denna informationsöverföring sker naturligt vid olika möten som utvecklare kontinuerligt har med kunden. Vid sidan om kunden och/eller beställaren finns en mängd olika aktörer och intressenter i projektet. Aktörer berörs direkt av projektet och intressenter berörs indirekt. Ett problem är att många inte är intresserade av det de berörs av dvs. de tar inte till sig den informationen som finns, många tycker t. ex. att intranätet borde vara mycket mindre än vad det är idag. Det kan ligga i ämnets natur (IT) eller att man har tillräckligt med saker att hålla reda på att det upplevs ett svalt intresse för den projektinformation som publiceras. Så här säger en person:

Och IT är mycket speciellt för det är en ny bransch och tyvärr är det många som inte har en aning om vad IT är för något, och det finns en sådan där kunskapslättja, man vill inte heller veta något...

Ytterligare ett problem som kanske är svårare att handskas med är det problematiska med att hitta de som berörs av projektet. Till en viss del kan det vara lätt att se aktörer men inte

intressenter, men i många fall är det till och med svårt att identifiera aktörer. Ju större ett projekt är desto mer beroenden skapar det och desto mer personer berörs. Vid större projekt brukar man skapa en hel projektorganisation med en styrgrupp, projektledning etc. men vid små projekt svarar oftast en person som projektledare för alla samtidigt pågående. En tendens har varit att det är svårare att hitta personer eller tjänster som berörs negativt vilket kan få stora konsekvenser för projektet. En person säger så här efter en sådan upplevelse:

...i det här fallet folk som snackade mycket skit om systemet och påverkade negativt, och det där hade vi fixat om vi hade räknat med dem från början.

En annan aspekt är att mycket information går utanför de formella informationskanalerna. Mycket utbyte, kanske den viktigaste, sker över kaffebordet vilket i sin tur kan förstärkas många gånger innan det kommer till de personer som berörs. Denna typ av informationsspridning sker slumpmässigt dvs. om spridningen sker överhuvudtaget eller vad innehållet är vilket kan uttryckas så här: ”Man sitter alltså med rätt personer...”

4.4 Personligt ansvar för utgången

Formellt sett skall utvecklingsgruppen agera som ”en (1) person” utåt mot t. ex. beställare. Det finns alltid en risk att en kund ser det som att den person som har haft huvudansvaret för utvecklingen är den som levererar. Denna problematik har diskuterats mycket under bildandet och forandet av gruppen och IT-avdelningen. Om gruppen agerar som en grupp så bemöts man också som det. Detta kan man utveckla genom att få in ett tankesätt att hjälpa och backa upp varandra, dela på arbetet etc. Ambitionen är att komma bort ifrån det personberoende som har funnits tidigare. Hur man känner inför en produkt som man håller på att utveckla eller ett projekt som man ansvarar för är till stor del personligt. Dock finns det ett arv från en tidigare hierarkisk organisationskultur som inom vissa avdelningar lever kvar. Denna kultur har lett till att man är mycket rädd för att göra fel på grund av att man kan bli personligt utpekad och få stå till svars för de misstag som begås. Dock ser man på sig själva i utvecklingsgruppen som att man har kommit bort ifrån mycket av den här typen av tankegångar. Vissa känner ett personligt ansvar mer än andra. En av de intervjuade ser en klar skillnad mellan de projekt som man är personligt engagerad i och övriga projekt som pågår i gruppen. Lojaliteten ligger dock alltid hos IT-avdelningen och man känner sig stolt om något har gått bra. Diskussionen blir lite tvetydlig hos en annan person:

...men det kände inte jag som något personligt nederlag. När man har gjort vad man har kunnat men inte når ända fram, då är det andra faktorer som avgör.

Men å andra sidan:

...det är ju inte bara tiden man lägger ned i projekt utan man engagerar sig ju verkligen i det man håller på med och man tar till sig både i med och motgångar, en bra projektledare påverkar projektet mycket.

Vad man känner personligen inför ett visst projekt eller resultat kan också handla om ”mjuka värden”. Vissa personer ser bara en härfin skillnad mellan sin yrkesroll och sin personliga roll. De med- eller motgångar som yrkesrollen drabbas av drabbar också den personliga

rollen. Å andra sidan finns det personer som skiljer rollerna åt och har en förmåga att agera prestigelöst när det kommer till att styra om eller avbryta projekt.

4.5 Metod för återkoppling och lärande

Ingen i utvecklingsgruppen har en bestämd uppfattning om det verkligen sker någon återkoppling eller eftertanke när projekt har haft tendenser att glida iväg. Mycket av diskussionen kring projekten sker på möten som hålls på veckobasis och det är också här som den här typen av frågor tas upp. Det finns en tendens att lägga ned relativt mycket energi när det uppstår problem i något projekt. En sådan situation kan få flera av gruppmedlemmarna engagerade och man hjälper gärna till i syfte att lösa situationen. Dock är det sällan som det sker en återkoppling och utvärdering om varför något gick snett från början. Denna typ av diskussion sker mer informellt och det finns inget bestämt om tidpunkt eller att det skall ske i vissa former. En av de intervjuade påvisar de positiva effekter det arbetssätt gruppen har på återkoppling och lärandeprocessen. Aktivitetsplanen där varje aktivitet listas går igenom på regelbunden basis vilket innebär att personalen måste avlägga en statusrapport om varje projekt. Detta kan göra att man tvingas in i en diskussion om varför någonting inte fungerar. Även om det inte formellt ligger i arbetsrutinerna sker det vid behov en utvärdering av gruppens projektstyrningsmetodik. Det är framförallt en person som arbetar med förbättringar av den metodologi som används. Diskussion kring det arbetssätt man har sker i huvudsak på de utsatta veckomötena. De ämnen som diskuteras tas sedan upp med gruppens chef vilket i sin tur för en diskussion med IT-rådet. IT-rådet skall representera verksamheten och där ingår chefer från olika områden i Försäkringskassan i Västra Götaland.

4.6 Stöd i projektstyrningsmodellen

Datumstyrd utveckling

Framförallt framhålls vikten av att arbeta med datum i utvecklingsarbetet. Den typ av iterativt utvecklingsarbete som används kräver att man ser hårt på t. ex. leveransdatum. Det finns en stor risk med att inte sätta ett leveransdatum eller att inte respektera datumet tillräckligt. Väldigt många av de projekt som har dragit ut på tiden och som man har tappat kontrollen på har saknat bestämda tidpunkter för leverans. I regel är inte tidsplanen bestämd i den iterativa delen i utvecklingsarbetet. Det upplevs som svårt att sätta ett datum för t. ex. en slutleverans om inte alla avgränsningar är gjorda. Är uppdraget väl avgränsat så vet man vilka delar som skall levereras och då är det lättare att sätta upp en tidsplan med en deadline för projektet. Inträffar det något under projektets gång som gör att leveransdatum måste senareläggas tas en diskussion med beställaren om detta. Går det inte att flytta ett datum så finns möjligheten att ta in externa konsulter. Det finns en policy att projekt skall bestå av minst två utvecklare som båda skall vara insatta i utvecklingsarbetet. Genom det minimeras sårbarheten vid t. ex. sjukdom och verkar för att leveransdatum hålls.

Projektspråk

Språket kan också vara en nyckel till att behålla kontroll över projekt. Ett gemensamt språk som förstås av alla i projektet skapar förutsättningar för att eliminera missförstånd samt för att

komma överens kring olika lösningar. Det uppstår många situationer som är jobbiga och slitsamma p.g.a. kommunikationssvårigheter. Det kan handla om varför man bör lägga ned projekt eller varför man bör satsa på projekt. Svåra situationer kan också uppstå vid diskussion kring hur mycket eller vilken funktionalitet som skall finnas i systemet. Det är ibland svårt att komma fram till en gemensam bild över kravspecifikationen och vilka avgränsningar som skall göras. Det får till följd att det blir svårt att datumbestämma projekt. Ett gemensamt språk skulle underlätta för beställare och utvecklare att komma närmare och skapa en, från två håll, accepterad bild över problemområdet. Det har uppstått slitningar mellan olika grupper inom Försäkringskassan i samband med utvecklingsarbete. Både slitningar mellan olika regioner och mellan grupperingar inom regionen vilket till viss del förklaras av avsaknaden av ett gemensamt språk.

Det mesta av utvecklingen sker mot linjen dvs. mot handläggare och utredare och det kan ibland uppstå en klyfta mellan linjen och projektet. Språket påverkar detta förhållande både direkt och indirekt. Direkt genom att verka som ett redskap för att skapa en gemensam verklighetsbild och indirekt genom att förankra projektbeslut i linjen vilket sker genom representanter (chefer) från linjen. Det upplevs ibland som problematiskt att göra sig förstådd i frågor som rör IT i dessa sammanhang. Samtidigt krävs det att besluten förankras genom representanter p.g.a. att projektmedlemmar inte har den makten. Ett gemensamt språk krävs också för att samordna aktiviteter och utnyttja resurser på bästa sätt. Vid ett flertal tillfällen har det skett missförstånd om vad som sedan tidigare är utvecklat av t. ex. andra försäkringskassor och vad det egentliga behovet har varit. Utvecklingsarbeten har av misstag startas upp eller lagts ned p.g.a. den typen av svårigheter. Så här uttrycker sig en person:

Linjechefer skall vara med i styrgruppen så att det som kommer från projektet förankras i linjen, det är svårt att fixa, inte bara i vår grupp, utan i det större perspektivet också (...)
Många projekt som gör samma sak och inte kommunicerar med varandra, och linjen driver sådant som man har sagt nej till (...) mycket viktigt att bädda in linjen i projektet så att resultatet blir förankrat vilket betyder inte bara att man godkänner utan att man har förstått vad projektet säger.

Tillit till varandras yrkeskunskap kan också öka genom att man språkmässigt befinner sig på samma plan. Genom språket skapas en förståelse för varandras yrkesroll vilket leder till ett ökat förtroende. Ibland måste en part helt lita på en annan just p.g.a. dess yrkesroll. En utbildad systemvetare med lång erfarenhet måste få viss handlingsfrihet när det gäller frågor som rör IT, vilket också gäller för en person med lång försäkringserfarenhet när det gäller försäkringsfrågor.

Projektdokumentation

Flera av de intervjuade framhåller vikten av att dokumentera genom hela projektet. En gemensam dokumentation för hela projektet kan fungera som ett kontrakt. Där står kravspecifikation, leveransdatum och avgränsningar och uppstår det oklarheter längre fram i projekt kan man gå tillbaka och se vad man gemensamt har bestämt. Dokumentation tillämpas numera med högre frekvent och med bättre kvalitet än vad man har gjort tidigare i utvecklingsgruppen. Projekt pågick under lösare former tidigare vilket kunde leda till att projektets avslutande blev otydligt. Om man ville gå tillbaka och titta på beslut så fanns oftast inte det vilket gjorde det svårare att få projekt som hade pågått för länge i en ”felaktig”

riktning under kontroll. Genom att införa ett gemensamt dokumentsystem kan stegen som skall genomgås i projekt förutbestämmas. För att gå in i en ny fas i projektet krävs att vissa faktorer är uppfyllda. Är de inte det så kan projektet inte gå vidare. Dokumentation gäller även i ingångsskedet av ett projekt dvs. för att ens starta upp ett projekt så måste vissa delar bl. a. en väl formulerad beställning och rollbesättning vara klart. Ett sådant arbetssätt tvingar också in beställaren i ett visst tankesätt där det gäller att formulera behov och beräkna nyttoeffekter vilket är en bra grund för ett lyckat projekt. Mycket enkla saker kan också påverka kontrollen av projektet. Vanliga missförstånd kring facktermer eller begrepp kan undvikas genom dokumentation där begreppen som används i projektet förklaras.

Dock upplever många att dokumentera inte är så lätt och man måste ställa höga krav på den om den skall uppfylla sitt syfte. Det måste vara tydligt så att alla förstår. I projekt ingår personer med olika yrkesbakgrund och alla skall kunna enas om innehållet. Det arbetet som läggs ned på dokumentation måste hela tiden avvägas mot den nytta som man får ut. Det är lätt att kväva kreativitet genom att skapa merarbete. Informationen måste också vara tillgänglig för alla i projektet så att man hela tiden har möjlighet att kontrollera eller läsa på.

Aktivitetsplan

Kanske den mest framgångsrika metoden att få kontroll över projekt upplever de flesta är aktivitetsplanen. Genom den skapas en informationskanal inom utvecklingsgruppen om vilka projekt och aktiviteter som pågår eller har avslutats. Aktivitetsplanen ligger ofta till grund för dagordningen på de veckobaserade möten som gruppen har. Projekten går igenom och här finns tillfälle att ta upp problem eller fråga om hjälp. En beskrivning av en av de intervjuade:

...jag tycker att det arbetssätt som vi har hittat nu, att vi har med aktiviteterna, att man tar alltid upp dem, det finns inte projekt som rullar och går utan att vi, (paus) om vi skall förnya en deadline, den pressen behöver man, och det är klart att man kan ändra datum om man ser att detta inte håller men då har man ändå inför sig själv och gruppen motivera vad det är som har hänt och händer detta gång på gång så har man en chans att kunna fånga upp varför det går snett.

Samarbetet kring aktivitetsplanen har lett till ett mer öppet klimat i utvecklingsgruppen. Man tränas hela tiden att ta itu med problem som uppstår och det sker på ett naturligt sätt. Det var tidigare en större personlig press p.g.a. att samverkansformerna inte var utvecklade som de är idag. ”Man kände sig oumbärlig” säger en av utvecklarna och syftar t. ex. på att man sällan sökte efter kompetens i gruppen när man hade problem tidigare utan försökte lösa mycket på egen hand.

Utvärdering

Utvärdering är en av de bitar som enligt alla saknas i gruppens projekt. Flera tar även upp att det generellt är dåligt med projektuppföljning inom Försäkringskassan. Alla inser dock fördelar med att följa upp projekt, under hela projekttiden, inte bara efter avslut.

Nyttoeffekten är ett nyckelord inom utvecklingsgruppen. Nyttoeffekten visar den nytta som en utvecklad produkt tillför verksamheten. Nyttoeffekten ställs ofta i relation till satsade resurser för att kunna prioritera. Att prioritera anses vara svårt på grund av att det ofta i en förstudie är svårt att bestämma behoven som skall tillgodoses vilket leder till problem med att

fastställa en kravspecifikation. Genom erfarenhet vill man kunna avgöra hur en prioritering skall göras och erfarenhet kan fås genom att följa upp projekt. Att följa upp projekt innebär att man i efterhand beräknar nyttoeffekten och ställer den i relation till det som hade prognostiserat. Gruppen är sysselsatt med att undersöka metoder för att följa upp projekt i ekonomiska termer, dels metoder för att bestämma utvecklingskostnad och dels metoder för att beräkna den ekonomiska effekten av ett projekt. Att utforma modeller för utvärdering är svårt på flera sätt. Under utvecklingsprojekt möts man av mycket åsikter från dels beställare och dels användare, vilka oftast är positiva. Det beror till stor del på att det är en hög arbetsbelastning på de användare som utvecklingen riktar sig emot. Utvecklingsgruppen är alltså nyckeln till mycket lösningar som kan innebära en lättare arbetsbörda för dem. Ett problem för utvecklaren blir då att göra en objektiv bedömning på vilken nyttoeffekt projektet egentligen tillför. Det är alltså svårt att se kritiskt på projektet inifrån projektet. En tanke är att föra in en bedömare utifrån, t. ex. från en annan avdelning, men just nu saknas resurser för det.

En annan viktig aspekt som hänger ihop med utvärdering är att det är svårt för de personer som ingår i projekt att ha en bred syn på verksamheten. En bred syn krävs för att man skall kunna göra en riktig bedömning över nyttoeffekten. En ansats som man försöker jobba med är att lyfta upp frågor till högre nivåer till personer som ser bredare. En sådan kanal är t. ex. IT-rådet där frågor tas upp genom chefen för IT-avdelningen och utvecklingsgruppen. Processen fungerar bara delvis så som man vill ha det men det pågår förbättringar. Det breda perspektivet beskrivs så här av en person:

Det är svårt att lägga ned projekt, det ligger i projektledarens intresse att behålla det, så är vi människor. Då är det viktigt att ha en styrgrupp som har det breda perspektivet i sammanhanget och ställer krav på projektledare.

5 Diskussion

Att runaway projects är ett komplext fenomen kan man konstatera i den här studien. Många faktorer tycks påverka och det finns många aspekter att ta hänsyn till när situationer skall analyseras. Därför blir det svårt att dra nytta av normativa projektstyrningsmodeller vid en uppkomst av runaway projects. Det går inte att hantera sådana komplexa problem med ”steg för steg-anvisningar”. De projektstyrningsmodeller som har studerats erbjuder ett till viss del dåligt stöd för att skapa en miljö som skapar förutsättningar för en hantering av runaway projects. Den främsta anledningen till det är dess utopiska karaktär. De beskriver ”perfekta” tillstånd som man som projektdeltagare troligtvis har svårt att känna igen sig i, i en runaway-situation. Det saknas hanteringsmodeller för projekt som går utanför den spikraka, perfekta, vägen mot målet, som egentligen aldrig existerar i verkligheten.

En stor brist i modellerna är att det inte tas någon hänsyn till människans personliga egenskaper. Till exempel förväntas att projektledning och deltagare hela tiden skall ta rationella beslut, dvs. agera som programmerade maskiner. Vad som har framkommit under studien på Försäkringskassan är att beslut och handlingar i många fall styrs av känslor eller påverkas av sociala aspekter. Det saknas hanteringsstöd för ett irrationellt handlingssätt som orsakas av psykologiska eller beteendevetenskapliga faktorer. En positiv aspekt på McConnells (1998) ”Staged delivery” kopplat till Lyytinen och Robeys (1999) studier kring lärande är att se modellstöd för en kontinuerlig utvärdering av det arbetssätt som används. Utvärdering sker kontinuerligt under projektets gång och inte efter.

Det är svårt att på kort tid överblicka i vilken utsträckning det existerar runaway projects inom utvecklingsgruppen. Dock har det under studien framkommit att problematiken känns igen och det är relativt lätt att associera till projekt som har haft sådana tendenser. Genom att ta upp riskfaktorer som kan påverka kvaliteten på de produkter som utvecklas höjs medvetandenivån och det blir betydligt lättare att agera undvikande gentemot de risker som t. ex. förknippas med runaway projects. Keil (1995b) har identifierat och kategoriserat områden i vilka det finns faktorer som kan orsaka project escalation. Det finns i den studierade miljön faktorer som talar både för och emot en uppkomst av runaway projects. Faktorer som talar emot en uppkomst:

- Försäkringskassans verksamhet är inte vinstdrivande. Till skillnad från den direkta kopplingen till ekonomisk vinst i kommersiella organisationer finns det inget ekonomiskt avkastningskrav för gruppens projekt.
- Projekttiden är relativt kort och projekten genererar snabba vinster i verksamheten. Genom det får man snabb respons på sin insats vilket minskar osäkerhet kring projektutgång och måluppfyllelse på längre sikt.
- Med- och motgångar hanteras som grupp. Trots att det finns tendenser på att man i vissa lägen tar ett stort personligt ansvar är ändå helhetsbilden sådan. En gruppsammanhållning stärker den enskilde att ta, ibland svåra, beslut som rör projekt.

Faktorer som talar för en uppkomst:

- Osäkerhet kring gruppens roll i organisationen. Gruppen är relativt nybildad och dess position och identitet är inte helt fastställd samtidigt som man har ändrat arbetssätt. Man kan lätt bli ifrågasatt av användare när man slutar ta korridoruppdrag och vara den som ”fixar allt” utan att ifrågasätta varför man bör göra det.
- Den iterativa modellen för behovs- och kravutveckling. Analysfasen kan pågå hur länge som helst om det inte finns tydliga riktlinjer om när kraven skall vara insamlade. Det måste finnas metoder för att ena utvecklare och beställare kring systemkraven för att kunna gå in i designfasen.

Många projekt som har varit i riskzonen går att härleda tillbaka till en utdragen kravanalys. Den iterativa metoden kan lätt leda till att man rycks med i ett högt tempo där man hela tiden hittar nya systemmöjligheter. Lyytinens & Robeys (1999) teori om misslyckad lärandeprocess kan ses som en förklaring till detta. En projektstyrningsmodell där utvecklingsgruppen kontinuerligt ställde sitt arbetssätt under kritiskt granskning skulle kanske utmynna i nya metoder för kravhantering där processtiden kunde kortas ned. Keils (1995a) tankar om agency-theory kan också ge en trovärdig bild av problematiken. Det projekt som tas upp under ”Fallbeskrivning 2” kan ha pågått länge utan att personer i ledande positioner med strategisk blick haft en möjlighet att sätta sig in i situationen. Både utvecklare och användare kan utveckla mål och intressen som skiljer sig från ledningens vilket kan leda till en uppenbar risk för runaway project.

Studien visar att den projektstyrningsmodell som används av utvecklingsgruppen bara delvis stödjer hantering av runaway projects. Att belysa brister är viktigt men lika viktigt är det att fokusera på det som fungerar bra. Följande punkter belyser viktiga diskussionsämnen vid en utveckling av rådande projektstyrningsmodell.

- Aktivitetsplan och veckomöte är tillsammans kanske det viktigaste hanteringsstödet för runaway projects. Det blir en naturlig informationskanal och fungerar som en arena där problem tvingas upp till ytan. Det blir också en naturlig plats för att be kollegor om hjälp och omfördela arbete om det skulle behövas. Denna arena skulle kunna utvecklas för att belysa ytterligare projektfaktorer. Liksom Keil (1995b) skriver har det också framkommit under de genomförda intervjuerna att projekt kontinuerligt borde granskas objektivt av en person som inte har ett direkt intresse i projektet. Denna person kan ta rollen som ”kvalitetssäkrare” med befogenheter att inhämta all information om projektet både från levererande och mottagande part. Till viss del görs detta i IT-rådet men med en veckovis kontinuitet skulle denna modell få betydligt mer genomslagskraft. Det objektiva perspektivet går utmärkt att kombinera med ett bredare perspektiv (organisatoriskt) i en och samma roll. Genom det skapas en mekanism för kontroll av organisationens intressen.
- Att arbeta med och hålla hårt på leveransdatum upplevs som en trygghet som man kan stödja sig emot. Det är en konkret faktor och fungerar som en ”ringklocka” för både projektmedlemmar och andra att projektet bör utvärderas om det blir förseningar. Arbetssättet går att utveckla att inte bara gälla leveransdatum utan också olika datum för milstolpar. Ett datum då systemkraven skall vara klara skulle kunna vara ett stort stöd för upptäckt av projekt som håller på att glida iväg. Ett mer sofistikerat system med ”tidiga

varningssignaler” där leveranser inte möter kvalitet eller tid kan skapa mervärde genom att de just upptäcks i tid. Det blir också enklare för projektmedlemmar att ”säga ifrån” när det finns vissa stadgar/regler i projekt-ramverket som stödjer detta rent praktiskt.

- Det finns inget utvecklat arbetssätt för projektuppföljning vilket bör implementeras. Lyytinen & Robey (1999) beskriver hur viktigt det är att implementera ett kritiskt förhållningssätt mot organisationens användarteorier. Genom att följa upp projekt på ett systematiskt sätt skapas kontinuerligt kunskap kring den projektmetodik som används och hur metodiken kan justeras för att undvika runaway-tendenser. Kunskap om följande viktiga områden skulle öka vid en kontinuerlig projektuppföljning:
 - Användarnytta är viktigt att kunna bedöma för att göra prioriteringar. Genom att prioritera rätt minskar risken för att hamna i en situation där det finns bättre alternativ. Om användarnytan skall stå i centrum måste det också ske en utvärdering under projekttiden där man dels bedömer den redan inslagna vägen och dels alternativa lösningar.
 - Kravhantering upplevs som det mest problematiska under utvecklingstiden och det finns mycket att lära inom det området. Som Lyytinen & Robey (1999) skriver finns det betydligt mer intern kunskap i den egna organisationen än vad man kanske tror, det gäller bara att ta vara på den. Behovs- och kravhantering handlar mycket om att kunna få användare att förmedla information till dig som systemutvecklare dvs. det handlar om kommunikation mellan människor. Eftersom alla organisationer har sin speciella kultur och sitt speciella språk bör kunskap om hur kommunikation fungerar bäst genereras internt. Genom det ökar tillämpningsmöjligheten.
 - Kompetenshantering handlar om att bedöma och hantera kompetensbehov i projekt och gäller samtliga projektmedlemmar. Kompetensbrister hos utvecklare i den teknik som skall användas kan leda till förseningar vilket gör att projektförutsättningar ändras. Gruppen står inför ett teknikskifte vilket gör det extra viktigt att belysa. Det gäller också att kunna bedöma kompetensbehov hos användare och beställare. Kompetensbrister bör upptäckas innan, inte under, projektets gång.
- Det måste utvecklas stöd för hantering av de psykologiska aspekter som associeras med projektarbete. Alla projektmedlemmar kommer från olika arbetsmiljöer och ingår i projekt, mer eller mindre, för personliga syften. Man måste säkerställa att individuella personer inte drar projekt åt något håll endast för personliga ändamål. Keil (1995a, 1995b) beskriver att beslutfattare utsätts för påfrestningar bl. a. i form av kognitiv snedvridning vilket också sker med beslutfattare i utvecklingsgruppen. En sak som påverkar är t. ex. gruppens servicekänsla och en inställning att man alltid skall leverera något. Detta förstärks av att gruppen ännu inte har identifierat sin roll fullt ut vilket driver en vilja att visa upp sig. Ett sätt att minska trycket på beslutfattaren är att göra som Keil (1995a) föreslår dvs. att implementera en beslutsstrategi där det undviks att en person tar beslut som följer på varandra.

Denna rapport ger en inblick i fenomenet runaway projects och vilka hanteringssvårigheter som förknippas med det. Författarens oerfarenhet av etnografiska studier har varit en brist i studien men trots det har många faktorer kunnat belysas. En intressant faktor är att

förutsättningar för att undvika en runaway-situation tycks vara relativt goda i en mindre projektorganisation liksom de som bildas av utvecklingsgruppen. En spekulativ tanke är att information lättare blir allmän och det är svårare att dölja personliga syften i den mindre organisationen. Ett förslag på ämne till vidare studier är att utreda om så är fallet.

6 Slutsats

Den studerande gruppen har upplevt runaway projects. I många fall handlar det om tendenser men i minst ett fall finns det tydliga indikationer på att projektet kan klassificeras som runaway project. Trots att många försök har gjorts för att stoppa projektet har det fortsatt att leva och ta resurser i anspråk. Runaway projects är inte vanligt förekommande i utvecklingsgruppen. Till viss del förklaras det av goda arbets- och organisatoriska förutsättningar t. ex. att organisationen inte är vinstdrivande, gruppen är relativt liten och projekten är korta. Man har ett mycket bra stöd i den aktivitetsplan som används där alla gruppens projekt finns identifierade vilka noggrant går igenom och statusbestäms varje vecka. Projektuppföljning bör implementeras i gruppens projektstyrningsmodell vilket skall fungera som en kontinuerlig lärandeprocess. Projektstyrningsmodellen bör också innefatta stöd för hantering av de psykologiska effekter som associeras med projektarbete och beslutsfattande. De projektstyrningsmodeller ur systemvetenskaplig litteratur som har studerats erbjuder till viss del ett dåligt stöd för hantering av runaway projects vilket beror mycket på dess utopiska karaktär. Det är osannolikt att projektdeltagare kan känna igen sig och applicera den information som ges på den egna situationen. Vidare tas liten hänsyn till människans psykiska karaktär dvs. att känslor påverkar det arbete vi utför och de beslut vi tar.

Referenser

Easterby-Smith M., Thorpe R., Lowe A. (1991). *Management Research, An Introduction*. London: Sage

Keil M. (1995a) *Escalation of commitment in information systems development: A comparison of three theories*. Atlanta: Department of Computer Information Systems, Georgia State University

Keil M. (1995b) *Pulling the plug: Software project management and the problem of project escalation*. Atlanta: Department of Computer Information Systems, Georgia State University

Keil M. (1995c). Identifying and Preventing Runaway Systems Projects. *American Programmer, March*, 16-21

Keil M., Mixon R., Saarinen T., Tuunainen V. (1995). Understanding Runaway Information Technology Projects: Results from an International Research Program Based on Escalation Theory. *Journal of Management Information Systems*, 3, 65-85

Lyytinen K., Mathiassen L., Ropponen J. (1998). Attention Shaping and Software Risk – A Categorical Analysis of Four Classical Risk Management Approaches. *Information Systems Research*, 9, 233-255

Lyytinen K., Robey D. (1999). Learning failure in information systems development. *Information Systems Journal*, 9, 85-101

McConnell S. (1998). *Software Project, Survival Guide*. Redmond, Washington: Microsoft Press

Royce W. (1998). *Software Project Management: A unified Framework*. Reading, MA: Addison Wesley Longman Inc.

Sommerville I. (2001). *Software Engineering 6th ed*. Harlow: Pearson Education Limited

Bilaga 1 Intervjufrågor

- ◆ Finns det, eller har det funnits, projekt som kan betecknas som runaway projects eller har sådana tendenser? Beskriv kortfattat ett sådant projekt.
- ◆ Är projektinformation tillgänglig för alla som berörs?
- ◆ Känner man ett starkt personligt ansvar för projektutgången?
- ◆ Vilken typ av återkoppling/utvärdering sker efter projektslut?
- ◆ Hur kan man förhindra runaway projects i utvecklingsgruppen?
- ◆ Vilket stöd upplever du från projektstyrningsmodellen?

Bilaga 2 Tre teorier om eskalering och dess underliggande antaganden

Theory	How the Theory Explains Escalation	Typical Unit of Analysis	Assumptions	Limitations
Self-Justification theory (SJT)	Managers continue to commit resources to a failing course of action in order to justify the correctness of an earlier decision to pursue a particular course of action.	Individual	<ul style="list-style-type: none"> Escalation is seen as a psychological phenomenon that leads to irrational behavior. Condition of personal responsibility seen as prerequisite for escalation to occur. 	<ul style="list-style-type: none"> Emphasizes psychological factors, but ignores social and political factors that may be relevant. Focuses on the individual unit of analysis even though escalation appears to be an organizational phenomenon.
Prospect theory (PT)	Managers commit resources to a failing course of action because the decision is framed as a choice between losses which lead to risk seeking behavior.	Individual	<ul style="list-style-type: none"> Escalation is seen as a cognitive bias that leads to irrational behavior. Condition of personal responsibility not viewed as a prerequisite for escalation to occur. 	<ul style="list-style-type: none"> Emphasizes cognitive biases but ignores other psychological, social, and political factors that may be relevant. Focuses on the individual unit of analysis even though escalation appears to be an organizational phenomenon.
Agency theory (AT)	Managers commit resources to a failing course of action because it is in their best interest to do so due to a lack of goal congruency between the manager and his superior and the existence of information asymmetry.	Organizational	<ul style="list-style-type: none"> Escalation is seen as an economically rational phenomenon that occurs under conditions of information asymmetry and a lack of goal congruency between subordinates and their superiors. 	<ul style="list-style-type: none"> Emphasizes some organizational factors but ignores psychological and political factors.

Tabell 3. Tre teorier om eskalering och dess underliggande antaganden (Keil 1995b s.8)