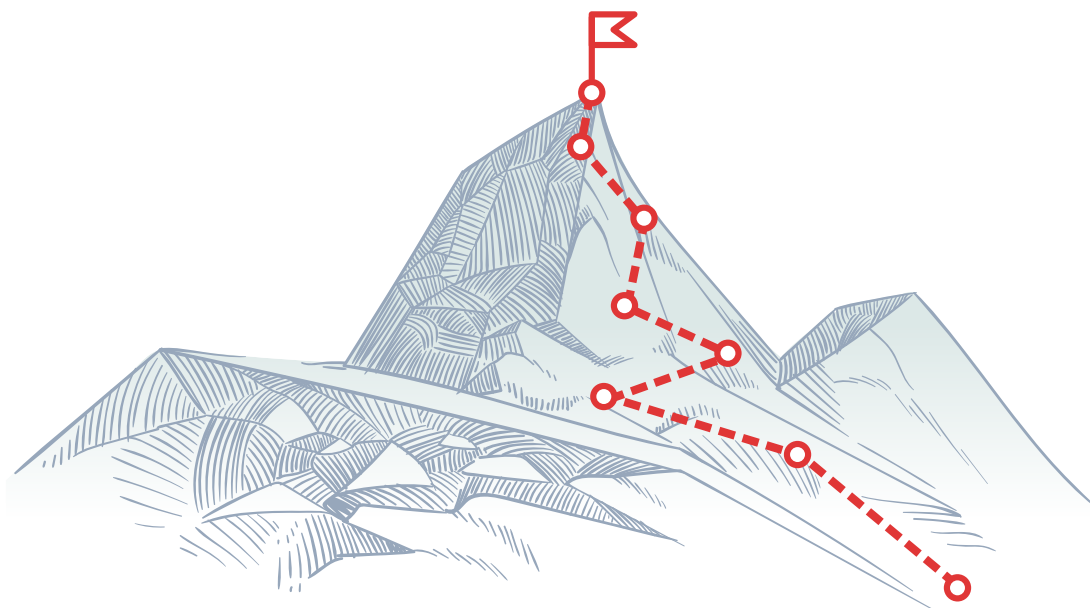

Tio framgångsfaktorer för lyckade it-projekt

I den här essän presenteras **sex kriterier** som kännetecknar lyckade it-projekt, samt **tio framgångsfaktorer** för att uppnå sådana. Essän avslutas med en **checklista** som kan användas som en enkel hälsoindikator för ett eller flera projekt.



Inledning

För att en organisation ska vara effektiv krävs i regel användbara och robusta it-system. Det finns tyvärr alltför många exempel på system som varken är användbara eller robusta, och där medarbetare får lägga tid och energi på att kämpa mot eller överlista systemen istället för att understödjas av dem.

Oavsett kvalitet uppstår inte ett it-system av sig självt. När det gäller större system som påverkar arbetsmiljön och effektiviteten för ett stort antal medarbetare, eller på annat sätt är kritiska för verksamheten, är det vanligt att de införskaffas inom ramen för ett projekt.

Användbara och robusta it-system förutsätter därmed lyckade it-projekt, vilket är temat för den här essän.

De tio framgångsfaktorerna

1. **Acceptans för att det är svårt**
2. **Projektledning och projektförutsättningar**
3. **Mod att avbryta och börja om**
4. **Rätt kompetenser genom hela projektet**
5. **Gruppen, rummet och verktygen**
6. **Beställansvar och rätt kontraktsform**
7. **Beaktande av hela systemlivscykeln**
8. **Behoven och kvaliteten i centrum**
9. **Parallell verksamhetsutveckling**
10. **Tydligt avslut och överlämning**

Sammanhang, utgångspunkt och viktiga begrepp

Med *it-projekt* avses i essän projekt som genomförs inom en organisation i syfte att förbättra organisationens verksamhet genom nya eller förändrade it-system. (Om det bedrivs projekt som inte direkt eller indirekt syftar till att förbättra något är det hög tid att fundera på vad som pågår i organisationen.)

Med *organisation* i föregående stycke avses exempelvis myndigheter, företag och föreningar, liksom större eller mindre enheter inom dessa organisationer. Med *verksamhet* alternativt *verksamheten* avses den verksamhet de nyss nämnda organisationerna bedriver för att fullgöra sina syften och åtaganden utifrån exempelvis regleringsbrev, förordningar, stadgar och bolagsordningar.

Ett antagande i resonemangen är att det finns en beställarsida som företräder verksamheten samt en utförarsida som ansvarar för projektgenomförandet. Dessa två sidor kan finnas i samma organisation likaväl som i ett kund och leverantörsförhållande.

Typer av it-projekt och dess resultat

Ett it-projekt kan vara av *integrationstyp* (sätta ihop och konfigurera befintliga komponenter eller system), *utvecklingstyp* (ta fram nya komponenter eller system) eller en kombination av dessa. De ingående komponenterna kan bestå av både hård- och mjukvara.

I essän används huvudsakligen begreppen *lösning* eller *system*¹ för projektens primära resultat. Ett system kan omfatta både produkter (exempelvis mjukvara) och tjänster.

Avgränsning

Slutligen: it-projekt och systemutveckling är svåra åtaganden. Det finns fler framgångsfaktorer än dessa tio, och beskrivningen av respektive framgångsfaktor skrapar enbart på ytan. Exempelvis utelämnas detaljerna och svårigheterna inom mjukvaruutveckling i det närmaste helt och hållet. Ramverk och processer för att skapa förutsägbarhet och minskat personberoende berörs inte heller. (Den som vill skapa förutsättningar för att göra framgångsrika it-projekt till regel snarare än undantag rekommenderas att ta del av den kunskap och best practice som tillhandahålls av CMMI Institute², INCOSE³ och PMI⁴.)

Förhoppningsvis är essän ändå till nytta, just på grund av det översiktliga helhetsperspektivet.

¹ Mer om systembegreppet finns på <https://www.svenskkravterminologi.se/terminologi/#systemperspektivet>.

² CMMI står för *Capability Maturity Model Integration*. CMMI Institute är en avknoppning från CMU SEI. (<https://cmmiinstitute.com>).

³ International Council on Systems Engineering (<https://www.incose.org>) är bl.a. involverade i systemlivscykelstandarden ISO 15288.

⁴ Project Management Institute (<https://www.pmi.org>). Se till exempel PMBok – Project Management Body of Knowledge.

Innebörden av ett lyckat it-projekt

Innan framgångsfaktorerna presenteras närmare är det lämpligt att definiera innebörden av ett lyckat it-projekt. Ett möjligt synsätt är att ett projekt som levererar uppsatta mål i tid och enligt budget per definition alltid är lyckat, oavsett om det resulterande systemet är användbart eller ej. Detta synsätt är uppenbarligen inte helt optimalt ur ett verksamhetsperspektiv.

Givet essäns utgångspunkt är det istället möjligt att relatera måttet på framgång till verksamhetsnyttan. Att formulera en slagkraftig och ändå heltäckande definition på en mening är dock svårt. Istället föreslås sex konkreta kriterier som alla måste vara uppfyllda:

Kriterier för [resultatet av] ett lyckat it-projekt

- K1** Den resulterande produkten eller tjänsten har en hög grad av användbarhet^a för dess avsedda användningsområde.
- K2** Alla intressentmål^b och tvingande krav^c är omhändertagna på ett för respektive kravställare acceptabelt sätt.
- K3** Införandet av produkten eller tjänsten sker i tid för att kunna genomföra planerade verksamhetsförbättringar.
- K4** Produkten eller tjänsten innebär både upplevda och mätbara förbättringar för den nyttjande verksamheten och dess användare.
- K5** Produkten eller tjänsten påverkar inte andra delar av verksamheten eller andra system på ett oavsiktligt negativt sätt.
- K6** De resurser, inklusive arbetstid, som förbrukats eller nyttjats under projektet, samt som prognostiseras behövas under produktens eller tjänstens fortsatta livscykel står i proportion till verksamhetsförbättringen.

a) Med **användbarhet** avses Gulliksens och Göranssons definition:

den utsträckning till vilken en specificerad användare kan använda en produkt för att uppnå specifika mål med ändamålsenlighet, effektivitet och tillfredsställelse, i ett givet användningssammanhang.¹

b) Med **intressentmål** avses definitionen i Svensk kravterminologi:

Mål som beskriver essensen av systemet. Dessa är av sådan dignitet att om de inte uppfylls, kommer systemet helt eller till stor del att sakna värde för den eller de intressenter som äger målen.²

c) Med **tvingande krav** avses definitionen i Svensk kravterminologi:

Krav som inte är förhandlingsbara till sin existens eller uppfyllnad.²

¹ Användarcentrerad systemdesign, Studentlitteratur 2002

² Svensk kravterminologi utgåva 2018-09-09-1836U

Kriterierna kan vid första anblicken framstå som ofullständiga. Till exempel saknas explicita kriterier för it-säkerhet. Detta, liksom andra kvalitetsaspekter ingår dock implicit i flera av kriterierna:

- Ett system som inte är tillgängligt när det behövs, eller vars information är korrupt är inte användbart (**K1**).
- Ett system som inte håller måttet säkerhetsmässigt kommer att kräva omfattande resurser för att lappa och laga det efter att det tagits i drift (**K6**). Det kan också innebära att viktiga funktioner inte får användas (**K1** och **K4**).
- För många verksamheter torde det finnas både intressentmål och tvingande krav som reglerar säkerheten (**K2**). Två källor till tvingande krav är exempelvis säkerhetsskyddslagen och dataskyddsförordningen ("GDPR").

Acceptans för att det är svårt

För över 30 år sedan skrev Frederick P. Brooks sin berömda artikel "No Silver Bullet" där han konstaterar att: "[...] *building software will always be hard. There is inherently no silver bullet.*" Sedan dess har ett antal utvecklingsmetoder som alla presenteras som den perfekta lösningen publicerats, liksom ett stort antal forskningsrapporter om misslyckade it-projekt och de bakomliggande orsakerna till dessa misslyckanden. Det förefaller därmed som att Brooks tes fortfarande håller.

Uppenbart är att det än idag är svårt att konstruera och integrera användbar och robust mjukvara i informationssystem som är till nytta för verksamheten. Trots det bedrivs alltför många it-projekt på ett sätt som ignorerar både den teoretiska och empiriska kunskapsutvecklingen inom området.

Statistiken talar sitt tydliga språk. Om det var lätt skulle det inte finnas så många misslyckanden att analysera för forskarna. Innebörden av den första framgångsfaktorn är därför **"acceptera att it-projekt och mjukvaruutveckling är svårt, och agera och utifrån detta faktum."**

It-projekt och mjukvaruutveckling kräver kunskap och färdighet

Jämfört med exempelvis brobyggande är programvaruteknik en mycket ung vetenskap som bara går tillbaka till 1960-talet. Givet mjukvarans abstrakta natur, och den låga tröskeln till att skapa program, förefaller det finnas en föreställning om att den som kan skriva ett fristående program eller en app på fritiden lika enkelt kan skapa verksamhetskritiska tillämpningar som ska leva och nyttjas i många år.

Samma föreställning verkar gälla för andra kritiska kompetenser kopplade till framgångsrika it-projekt, nämligen projektledning och kravhantering. Även för dessa är tröskeln låg för att genomföra aktiviteter som i vid bemärkelse faller inom områdena, samtidigt som de negativa följderna av otillräckliga kunskaper och färdigheter är stora eller i värsta fall katastrofala. Dessutom är det svårt för den som inte själv besitter tillräcklig kunskap att utvärdera andras kompetens innan det är för sent.

Det stora problemet är alltså inte att teorier eller metoder saknas, utan att det finns för få personer som utbildas inom dem. Att för få utbildas beror bland annat på att det inte ställs krav på att projektmedlemmar ska ha kunskaperna. En möjlig förklaring kan vara att it-projekt, till skillnad från mer fysiskt handgripliga ingenjörsvetenskapsdiscipliner, inte anses vara en tillräckligt svår utmaning. Misslyckade it-projekt leder oftast inte heller till spektakulära nyhetsrubriker på samma sätt som byggnader som rasar till följd av konstruktionsfel.

Myt: "Vem som helst kan skriva krav [i naturligt språk]"

Det är sant, på samma sätt som att vem som helst kan göra en byggnadsritning. Frågan är hur bra resultatet i nästa steg blir.

Naturligt språk ("vanlig text") är en av de svåraste notationerna att skriva entydiga (tolkas lika av alla läsare) och verifierbara krav i.

Att identifiera vilka krav som motsvarar verksamhetens faktiska behov är en konst i sig. Därefter ska dessa uttryckas och förmedlas på ett sätt som projektmedlemmarna, utvecklarna och övriga intressenter förstår.

För att lyckas med ovanstående aktiviteter krävs teoretiska kunskaper kombinerat med praktiskt erfarenhet.

Således, den som agerar utifrån att it-projekt är den utmaning som forskning och empiri visat kommer förmodligen att fokusera på att hitta rätt kompetens, samt lägga mer tid på de tidiga faserna i projekt- och systemlivscyklerna. I gengäld kommer deras projekt att avslutas i tid och till totalt lägre kostnad.

Projektledning och projektförutsättningar

Till skillnad från en bra sås reder (eller leder) inte ett projekt sig självt. Projektledning som kompetensområde får sällan den uppmärksamhet det förtjänar, men utan en kompetent projektledare krävs i det närmaste ett mirakel för att uppnå ett framgångsrikt it-projekt.

En grundförutsättning är därför en projektledare som behärskar både teori och praktik kring att leda framgångsrika it-projekt. Beroende på projektets storlek och komplexitet kan denne i sin tur behöva en eller flera delprojektledare.

Det räcker dock inte med en duglig projektledare. För att nå de **sex kriterierna** för ett framgångsrikt it-projekt måste projektledaren också få tillräckliga resurser samt uppbackning från beställaren:

- 1) Ett **projektledningsdirektiv** som reglerar vad projektet ska leverera, samt vilka resurser och mandat projektledaren har för att åstadkomma detta. Projektdirektivet lägger grunden för en gemensam målbild både inom och utanför projektet.
- 2) En **styrgrupp** som godkänner grindbeslut och fasövergångar i projektlivscykeln, och som projektledaren kan vända sig till för exempelvis prioriteringar när så behövs.
- 3) En **projektsponsor** som är beroende av och engagerad i projektets leveranser. Sponsorn behöver ha tillräckligt med inflytande i organisationen för att kunna skydda projektet och ge projektledaren arbetsro, samt se till att nödvändiga resurser görs tillgängliga för projektet.

Myt: "Projektledning är inget som kräver teoretiska kunskaper, man lär sig genom att leda projekt"

Även denna myt har en viss grad av sanning i sig. Det är fullt möjligt att leda projekt utan att ha kunskaper i projektledningsmetodik.

Frågan är dock hur många misstag och misslyckade projekt som behövs innan en individ på egen hand har tillägnat sig en bråkdel av den best practice som finns tillgänglig hos exempelvis PMI, CMMI Institute och INCOSE. *(Länkar till ovan nämnda org. finns i slutet på sidan 2.)*

Mandat, resurser och möjlighet att leda

För att projektledaren ska kunna ta ansvar för resultatet måste denne ha ett reellt mandat att leda projektet:

- **Projektdirektivet gäller.** Projektets mål och leverabler får inte ändras på annat sätt än genom uppdatering av direktivet, vilket sker i samråd med projektledaren. Därefter kommunicerar projektledaren förändringarna till projektgruppen och externa intressenter.
- De **resurser** som ställs till projektledarens förfogande **får inte flyttas** från projektet utan att projektledaren informeras. En sådan omprioritering medför nästan alltid att projektet försenas eller blir dyrare, vilket den som beslutar om förändringen måste ta ansvar för.
- **Styrgruppen följer** projektets utveckling, **samt styr genom att fatta grindbeslut.** De styr däremot inte arbetet på detaljnivå gentemot specifika projektmedlemmar.

Projektledaren leder projektet, men har ett ansvar att hålla styrgrupp, projektsponsor och övriga intressenter informerade om utfört arbete, tidplan, ekonomi och risker. Projektledning handlar lika mycket om att koordinera och följa upp aktiviteter som att kommunicera och hantera förväntningar inom och utanför projektet. Projektledaren är beroende av, och måste förtjäna, förtroende från samtliga inblandade.

Mod att avbryta och börja om

Vissa projekt fortsätter att rulla på trots att många inblandade inser att det inte har förutsättningar att lyckas. Detta kan bero på en för optimistisk tidplan, för lite resurser, eller helt enkelt att verkligheten förändrats så att resultatet från projektet inte längre behövs.

Samtidigt, ju längre ett projekt har pågått och ju mer resurser det förbrukat, desto svårare är det att lägga ner det och börja om från början. Både organisatorisk och personlig prestige kan stå på spel, inte minst i sådana lägen där projektet redan vid ett eller flera tillfällen fått extra resurser för att "få det på banan igen".

Visst gör det ont när it-projekt brister

Det kan finnas flera bidragande faktorer till att projekt som passerat "bäst före" tillåts fortsätta och i värsta fall också övergår till skenande kritiska projekt (se rutan till höger). Till exempel:

Psykologiskt motstånd: *"Om vi avslutar projektet kommer allt arbete att ha varit förgäves; alla dessa utgifter kommer ha varit till ingen nytta!"*

Organisatoriska förutsättningar: *"Vi kommer inte att få chansen att göra om det, trots att vi nu vet hur vi borde ha gjort istället."*

Lösningen är ett projektklimat som gör det lätt att ta med lärdomar och leverabler till nästa projekt.

"Det skenande kritiska projektet"

Kännetecknande för det skenande kritiska projektet är att det har pågått så länge och konsumerat så mycket resurser i form av tid och pengar att det till varje pris måste fortsätta, just för att det redan har kostat så mycket.

Förutom att det troligen aldrig kommer att leverera enligt förhoppningarna tar det fortsatt stora resurser i anspråk, samt tränger undan andra projekt och förbättringsåtgärder.

Värdefulla erfarenheter eller bortkastat arbete

Oavsett kompetens och förutsättningar kommer det hända att projekt inte lyckas på första försöket. Frågan är om man ser det som bortkastat arbete, eller att man vunnit värdefulla lärdomar på vägen mot slutmålet. Några rekommendationer för den som eftersträvar det sistnämnda:

- Uppmuntra ett projektklimat där det är naturligt att våga prova och där ett avbrutet projekt ses som en möjlighet att dra lärdomar samt åtgärda felaktiga antaganden eller förutsättningar.
- Var hård mot projekten. Ett projekt som inte uppfyllt uppsatta kriterier för beslutsgrunden till nästa fas i livscykeln får inte gå vidare. Avsluta projektet om det inte går att nå nästa beslutsgrund eller slutmålet med acceptabel mängd utökade resurser.
- Ha rutiner för att snabbt kunna börja om ett projekt med reviderade ingångsvärden. Tillåt och uppmuntra det. Se till att avtal med leverantörer möjliggör avslut vid förutbestämda tidpunkter (se FRAMGÅNGSFAKTOR 6) samt fortsättning med en ny leverantör.
- Använd en väldefinierad process för att löpande utvärdera projekten och snabbt omhänderta nya lärdomar i både pågående och kommande projekt.

Den organisation som åstadkommer ovanstående kommer över tid att vara betydligt mer innovativ och framgångsrik än den som ser varje i förtid avslutat projekt som ett misslyckande vars resultat och lärdomar helst ska glömmas bort så snabbt som möjligt.

Rätt kompetenser genom hela projektet

Skapandet av ett användbart it-system kan liknas vid byggandet av en järnvägstunnel. Liksom vid ett tunnelbygge krävs specialister med rätt utbildning och erfarenhet, och aktiviteterna måste koordineras så att rätt expertis kommer in vid rätt tillfälle. Exakt vilka kompetenser som behövs, liksom antalet individer inom varje område, beror givetvis på projektets omfattning och komplexitet.

Utöver projektledning (se FRAMGÅNGSFAKTOR 2) finns det dock några kompetensområden som är särskilt kritiska. Detta gäller för både utvecklings- och integrationsprojekt. Dessa områden sticker dessutom ut genom att de måste omhändertas redan från början och därefter fortsatt hanteras genom hela projektlivscykeln:

Kravhantering

Kravhantering kan ses som en stödjande disciplin som möjliggör samverkan mellan övriga kompetensområden. Kravspecialisterna ser till att de verksamhetsbehov som it-systemet ska understödja identifieras och omsätts i nedtecknade, precisa, spårbara, verifierbara och entydiga krav och modeller som både verksamhetsföreträdare och tekniska specialister förstår.

Systemarkitektur

Arkitekturspecialisterna håller ihop helheten, både in- och utvändigt. Utvändigt gäller det att säkerställa att den nya lösningen fungerar tillsammans med redan befintliga system. Invändigt måste alla delsystem och komponenter fungera tillsammans när de sätts ihop. Det invändiga arkitekturarbetet är vitalt för att åstadkomma en förvaltningsbar produkt med låg komplexitet.

Användbarhet och it-säkerhet

Varken användbarhet eller säkerhet uppstår av sig självt eller kan byggas på i efterhand utan negativa konsekvenser. När båda perspektiven är med från början och integreras i systemarkitekturen kan däremot system som är både användbara och säkra uppnås. Säkerhets- och användbarhetsspecialisterna ansvarar bland annat för acceptanskriterier och verifieringsmetoder för kraven inom respektive kompetensområde.

Verifiering och validering (VoV)

Verifieringsspecialisterna kontrollerar att de specificerade kraven uppfylls, det vill säga att lösningen utformas på rätt sätt. Valideringsspecialisterna kontrollerar att den kompletta lösningen uppfyller verksamhetens faktiska behov. Både verifiering och validering sker löpande genom projektet för att kunna upptäcka och åtgärda avvikelser innan det är för sent. Ju tidigare en avvikelse upptäcks, desto billigare är det att åtgärda den.

När specialisterna inom ovanstående kompetensområden får möjlighet att arbeta tillsammans genom hela projektet blir effekten dessutom större än om de arbetat var och en för sig.

För de flesta projekt kommer förmodligen ytterligare specialister att behövas. Systemlivscykelstandarden ISO/IEC/IEEE 15288:2015 innehåller till exempel 29 större processer som alla innebär krav på särskilda kompetenser och metoder. För mjukvaruutveckling tillkommer ytterligare kompetenser. En viktig uppgift för projektledaren, och en framgångsfaktor för projektet, är därför att identifiera vilka kompetenser som behövs genom projektet och se till att de är tillgängliga vid rätt tillfällen.

Gruppen, rummet och verktygen

Det har redan konstaterats att lyckade it-projekt kräver flera olika specialistkompetenser. Det räcker dock inte att anställa eller hyra in dem; specialisterna behöver också rätt förutsättningar för att kunna utföra sina arbetsuppgifter så effektivt som möjligt.

Den tätt sammansvetsade gruppen

Ett it-projekt är i någon utsträckning en kamp mot klockan. Det gäller att hinna leverera resultatet innan de identifierande behoven eller de tekniska lösningarna blir inaktuella.

Arbete i allmänhet och samarbete i synnerhet kräver närvaro. En grupp blir inte sammansvetsad om den har svårt att hitta gemensam kalendertid eller lämpliga lokaler för att träffas.

Bäst resultat på kortast tid uppnås när en mindre grupp nyckelkompetenser får möjlighet att tillsammans fokusera på uppgiften. För att kunna fokusera krävs att projektet är den huvudsakliga och högst prioriterade arbetsuppgiften för dem.

Effektiviteten ökar dessutom markant om medlemmarna är lokaliserade i närheten av både varandra (samma korridor eller motsvarande) och ett dedikerat projektrum.

Projektrummet

Ett dedikerat projektrum med stora whiteboard-tavlor skapar möjlighet att diskutera och visualisera tankar på ett sätt som i dagsläget är svårt att ersätta med tekniska lösningar.

Rummet blir också en plats där prototyper kan tas fram, förvaras och utvärderas. Ett dedikerat projektrum ökar även effektiviteten eftersom projektmöten kan hållas med kort varsel när problem eller innovativa lösningar på dessa uppstår.

Genom att ha projektmålen samt den kritiska linjen uppsatta på en av väggarna skapas en gemensam målbild och uppfattning om vad som är mest prioriterat för stunden.

Rätt verktyg för rätt uppgift

Det är nog få som förväntar sig att en byggnadsentreprenör ska bygga ett konserthus enbart med hjälp av gammaldags handverktyg. Det är möjligt att det går, men det skulle säkerligen ta lång tid och bli väldigt dyrt. Samma sak gäller för it-projekt; det *går* att göra det med endast ordbehandlare, kalkylprogram och e-post, men det är inte särskilt effektivt och det medför stora risker. Några exempel¹ på viktiga kategorier av verktyg återges nedan:

- 1) Projektplaneringsverktyg, som visualiserar status, allokerade resurser samt resurskonflikter, WBS (Work Breakdown Structure) inklusive vad som är utfört, kritisk linje, risklogg, mm.
- 2) Kravhanteringsverktyg, där krav, acceptanskriterier och verifieringsstatus finns tillgängligt.
- 3) Kommunikationsverktyg, som möjliggör kommunikation med projektmedlemmar eller leverantörer på distans, både i realtid och asynkront.
- 4) Dokumenthanteringssystem, som versionshanterar befintliga elektroniska artefakter samt underlättar framtagandet av nya.

¹ Utöver dessa exempel tillkommer specifika verktyg för mjukvaruutveckling. I enlighet med vad som redan nämnts i inledningen går detta dock utanför omfattningen av den här essän.

Beställansvar och rätt kontraktsform

I föregående framgångsfaktorer har det konstaterats att it-projekt är svåra åtaganden som är beroende av rätt kompetens för att nå framgång. Ibland finns den rätta kompetensen utanför den egna organisationen. I så fall är det viktigt att säkerställa att den tilltänkta leverantören faktiskt har den efterfrågade kompetensen, samt att denne ges förutsättningar för att kunna göra ett bra jobb.

Leverantören och kontraktet

Huvudprincipen är att skapa långsiktiga relationer med kompetenta leverantörer, men utan att låsa in sig och bli för beroende av någon av dem.

- Lägg tid och energi på att hitta metoder att välja en kompetent leverantör istället för att försöka åstadkomma "idiotsäkra" specifikationer som inte kan misstolkas. Det sistnämnda är dömt att misslyckas av flera skäl, däribland Dunning-Krugereffekten¹.
- It-projekt och mjukvaror är av naturen unika. (Finns resultatet redan innan start är ju arbetet redan färdigt.) Därmed är både projekt och systemutveckling förknippade med stora osäkerheter, vilket gör att så kallade fastpriskontrakt innebär stora risker för både beställaren och leverantören. Även utan Dunning-Krugereffekten är sannolikheten stor att båda sidorna underskattar tidsåtgången, och det som blir kvar att kompromissa om är oftast kvaliteter som användbarhet, säkerhet och förvaltningsbarhet.

Utforma därför kontraktet utifrån ett etappvis utförande med möjlighet till avslut vid förutbestämda tidpunkter samt fortsättning med en annan leverantör. Även om optionen att avveckla en bemannad organisation i förtid kan medföra ett prispåslag så är det bättre än att tvingas fortsätta med ett samarbete som inte fungerar. (Se även FRAMGÅNGSFAKTOR 3.) Säkerställ också att kontraktet är kompatibelt med hur projektet i övrigt ska bedrivas, till exempel för att omhänderta framgångsfaktorerna 7, 8 och 9.

Kommunikation och delaktighet

- Dokumentera bakgrund och syfte för varje behov och krav. Leverantören kan inte läsa tankar. Spårbarheten som uppstår underlättar även för beställaren, och det kommer sannolikt visa sig att det finns flera olika uppfattningar om syfte och bakgrund även på beställarsidan.
- Var tillgänglig för leverantörens frågor, samt delaktig i övrigt. Är kontaktvägarna för krångliga, eller om svaren dröjer för länge, kommer leverantören troligen låta bli att fråga och istället gissa vad verksamheten behöver. Sådana gissningar blir inte alltid rätt.
- Om en agil utvecklingsmetod används behöver beställaren se till att personal med rätt kompetens och mandat är på plats hos leverantören under stora delar av projektet.
- Även om leverantören är anlitad som expert för att lösa beställarens behov finns det ibland designval där beställaren från början vet att endast en specifik lösning accepteras. Kommunicera detta, men var lyhörd ifall leverantören säger att det får konsekvenser.

¹ Effekten kan exemplifieras med en leverantör som har otillräcklig kompetens, men som är omedveten om detta och dessutom överskattar sin förmåga att förstå kraven och producera en lösning. Risker är ett överoptimistiskt anbud avseende både tid och kostnad. Om dessutom beställaren överskattar sin förmåga att skriva bra krav är katastrofen ett faktum.

Beaktande av hela systemlivscykeln

Syftet med ett it-projekt är typiskt att leverera ett nytt eller förändrat system. Projektet avslutas därmed i regel strax före eller efter det att systemet har införts i verksamheten. Samtidigt går systemet in i sin längsta och viktigaste fas, nämligen *användningsfasen*¹. Om inte hela systemets livscykel beaktas från början är risken stor att användningsfasen också blir den i särklass dyraste fasen.

Beakta särskilt det faktum att ett stort fokus på låga projekt- och utvecklingskostnader ofta (oavsiktligt) förskjuter kostnaderna till användningsfasen. En sådan prioritering kan till exempel uppstå implicit till följd av hur ekonomiska medel fördelas internt.

I enstaka fall, exempelvis för akuta temporära lösningar, kan det vara befogat att införa ett it-system med höga drift- och förvaltningskostnader. Temporära lösningar har dock en otrevlig tendens att bli kvar betydligt längre än man tänkt sig.

Oavsett vilken prioritering som görs, är det viktigt att den är tydligt i både projektdirektivet och i systemets kravmängd. Om ingen explicit prioritering görs kommer den att göras implicit beroende på faktorer som antingen är person- eller kontraktsberoende.

Obehagliga överraskningar?

Nedan visas några exempel på problem som kan uppstå när användningsfasen glöms bort under konceptframtagande och kravställning.

- ✗ Dyra årliga licenskostnader.
- ✗ Omotiverat stort behov av driftpersonal.
- ✗ Komplicerat (och därmed dyrt) att göra förändringar eller förbättringar.
- ✗ Svårt att hålla systemet säkert.
- ✗ Hårdvarukomponenter som tidigt blir EOL (End of Life) och därför måste bytas ut.

Undvik svarta kostnadshål

- Fuska inte i konceptfasen. Åttio procent av systemets totala livscykelkostnad bestäms i regel under de tidsmässigt första tio procenten i projektet. Se till att det finns en systemlivscykelmodell som beskriver systemets faser inklusive fasernas längd i tid.
- Se drift- och förvaltningsorganisationerna som viktiga intressenter. Dra nytta av deras kompetens i både kravställningen och den löpande verifieringen. Se särskilt till att de involveras vid val av hård- och mjukvarukomponenter.
- Eftersträva låg komplexitet och hög modularitet. Enkla och standardiserade gränssytor både inom systemet och mot omvärlden underlättar integration såväl som förvaltning.
- Förr eller senare ska systemet avvecklas. Den information som finns i systemet kan behöva migreras eller arkiveras, och hårdvarukomponenter kan behöva återvinnas. Krav på hur detta ska gå till behöver fångas upp tidigt.

Ovanstående text avser kostnader och problem som är relaterade till systemet i sig självt. För att undvika indirekta kostnader hos den nyttjande verksamheten till följd av ett system med låg grad av användbarhet behöver essän i sin helhet beaktas.

¹ Namnet kan variera beroende på vald systemlivscykelmodell.

Behoven och kvaliteten i centrum

Den slutliga verifieringen av ett system kan visa att samtliga krav är uppfyllda, det vill säga att systemet är byggt helt enligt specifikationen. Samtidigt kan valideringen¹ komma fram till att systemet är mer eller mindre oanvändbart för den organisation som ska ta emot och använda det. Det som då har hänt är att de ställda kraven inte motsvarar de verkliga behoven.

Grunden läggs i konceptfasen

Genom att från början tydligt beskriva vilka behov systemet ska realisera åstadkoms en stabil grund för utveckling såväl som för verifiering och validering. En sådan grund består av olika delar, som tillsammans kan kallas för ett *systemkoncept*. Tre viktiga beståndsdelar i systemkonceptet återges nedan:

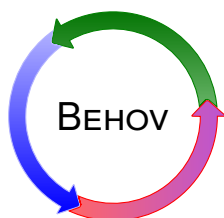
- 1) En övergripande beskrivning av systemet som tydligt definierar var systemgränsen går. Specificera förväntningar på kringliggande och eventuellt stödjande system, samt antaganden om omvärlden eller verksamheten som måste gälla för att systemet ska uppfylla sitt syfte.
- 2) Användningsscenarier och andra relaterade arbetsprodukter² som beskriver hur systemet kommer att användas av verksamheten. Dessa måste vara tillräckligt detaljerade för att kunna utvärderas genom rollspel av den personal som ska utföra uppgifterna med det nya systemet.
- 3) Krav och acceptanskriterier på och för kvaliteten inom olika områden som exempelvis användbarhet, förvaltningsbarhet och säkerhet inklusive tillgänglighet.

Teknikvarning!

Att skriva detaljerade tekniska krav tidigt i konceptfasen kan kännas konkret och roligt, men det riskerar att kraftigt begränsa nyttan med systemet. Behoven först, sedan tekniken!

Genom systemkonceptet får en kompetent utvecklarorganisation förutsättningar för att skapa en bra teknisk lösning. Därefter gäller det att löpande verifiera det framväxande systemet gentemot konceptet, samt (lika löpande) validera både konceptet och systemet gentemot verkligheten.

Löpande validering och verifiering fram till överlämning



- Följ upp att det framväxande systemet realiserar behoven i systemkonceptet, och att uppfyllandegraden av kvalitetskraven ökar.
- Följ samtidigt upp att antagandena och konceptet i övrigt fortfarande är giltiga. Både omvärlden och verksamheten kan ha förändrats.
- Utvärdera det framväxande systemet hos de blivande användarna.

Används en projekt- och utvecklingsmetodik där etappvisa leveranser införs och används skarpt av verksamheten sker valideringen i viss mån automatiskt. För projekt som inte kan drivas på det sättet kan istället prototyper, både i papper och mer realistiska varianter användas i den löpande valideringen. En annan bra metod är att så tidigt som möjligt ta fram och provköra manualer och utbildningsmaterial.

¹ <https://www.svenskkravterminologi.se/terminologi#validering-verifiering>

² <https://www.svenskkravterminologi.se/terminologi#arbetsprodukter>

Parallell verksamhetsutveckling

Målet med ett nytt it-system bör vara att verksamheten ska bli bättre på något sätt, exempelvis genom att minska onödig administration eller att kunna utföra uppgifter som inte tidigare var möjliga. För att uppnå sådana mål behöver ofta arbetsmetoderna utvecklas tillsammans med it-systemet. Detta innebär ytterligare utmaningar, men utdelningen är väl värd det extra arbetet.

Prioritering, samordning och mandatfördelning

Om rutiner och processer ska förändras till följd av det nya systemet är det nödvändigt att behovet är prioriterat av ledningen för verksamheten i fråga. Ledningens stöd behövs bland annat för att rätt personer ska kunna avsätta tillräckligt mycket tid för att delta i arbetet.

För att åstadkomma nödvändig samordning bör minst en verksamhetsföreträdare ingå i it-projektets styrgrupp, och det måste vara tydligt (bland annat reglerat i projektdirektivet) vad it-projektet respektive verksamhetsutvecklingen ansvarar för.

Kommunikation, insyn och delaktighet

Förändring, eller bara ryktet om en möjlig kommande förändring, medför oro hos dem som upplever sig kunna bli påverkade. Med tydlig information om hur det faktiskt kommer att påverka dem, tillsammans med god kommunikation i övrigt kan oron minimeras och till och med övergå i entusiasm.

- Var tydlig med vad både it-systemet och verksamhetsförändringarna handlar om, och informera regelbundet om hur arbetet fortskrider. Bjud in till, och sprid via videoinspelningar, demonstrationer av det kommande systemet. Visa hur det kan användas för att underlätta det dagliga arbetet. Berätta hur lärdomar och återkoppling från utvecklingsarbetet används för att förbättra det framväxande systemet och de kommande arbetsmetoderna.
- Etablera test- och referensgrupper med representanter från de delar i organisationen som i slutändan ska använda det nya systemet. Involvera dessa i utvärderingen av det framväxande systemet såväl som av det nya arbetssättet. Utöver kvalitetssäkringen skapar det ambassadörer runt om i organisationen som kan bidra till att minska oron.

Utöver delaktighet och tillgång till relevant information är det slutligen viktigt att användarna av systemet får tillräcklig utbildning i både systemet och det förändrade arbetssättet *innan* något av dem införs.

Stegvis införande med återmatning och förbättring

För nya it-system som också innebär förändrade arbetssätt finns mycket att vinna på att hämta inspiration från agila utvecklingsmetoder. Detta gäller inte bara för systemet utan också för de nya arbetsrutinerna.

Genom att utveckla och införa det nya systemet etappvis kan nya arbetssätt införas, utvärderas och förbättras parallellt med it-systemet. På så sätt minskar risken att det tänkta arbetssättet i slutändan visar sig vara inkompatibelt med systemet, alternativt inte alls fungera i verksamheten.

När medarbetarna tidigt kan börja nyttja systemet uppstår snabbt nya förslag på hur både systemet och arbetssättet kan förbättras. Om dessa återmatas och används i system- och verksamhetsutvecklingen ökar delaktigheten och engagemanget hos medarbetarna samtidigt som slutresultatet blir bättre.

Tydligt avslut och överlämning

Ett projekt är enligt gängse definition en i tid avgränsad uppgift med ett specifikt (unikt) syfte. Med andra ord, alla projekt har ett slut. Det måste vara tydligt för alla inblandade när i tiden slutpunkten ligger. Detta oavsett om projektet avslutas i förtid eller efter att målen i projektdirektivet uppnåtts.

Varför ett tydligt avslut är viktigt

- För projektmedlemmarna är det viktigt med ett tydligt avslut. Alla ska veta att arbetet är slutfört och att de istället kan fokusera på nästa projekt eller annan verksamhet.
- För den nyttjande verksamheten är det viktigt att veta att projektet inte längre existerar, och att fortsatt förändrings- och förvaltningsarbete sker enligt ordinarie rutiner.
- För de mottagande drift- och förvaltningsorganisationerna är det viktigt att veta från när i tiden de nya it-systemen är deras ansvar.
- För projektägare och sponsorer är det viktigt att veta att projektet är avslutat, och inte minst med vilket resultat.
- För resursägare, oavsett om de stått för experter, lokaler eller utrustning, är det viktigt att få reda på att resurserna kan omfördelas till andra aktiviteter.

Avslutande aktiviteter

Innan projektet avslutas formellt är viktigt att allt är färdigt, så att det inte uppstår extra uppgifter i efterhand. De före detta projektmedlemmarna är troligen redan fullteknade med annat arbete, vilket betyder att engagemanget för och därmed resultatet av exempelvis erfarenhetsåterföring kommer att vara begränsat.

Exempel på avslutande aktiviteter

- Säkerställ att drift- och förvaltningsorganisationerna bekräftat att de tagit över ansvaret för det levererade systemet
- Genomför avslutande utvärdering och erfarenhetsåterföring
- Lämna tillbaka eventuellt kvarvarande ekonomiska medel och avsluta konton
- Paketera och överlämna relevant information för arkivering och rensa resten
- Lämna tillbaka utrustning, avsluta it-konton och it-tjänster och se till att lokaler är utrymda och städade. Se helt enkelt till att inga lösa trådar är kvar för någon annan att ta hand om.
- Kommunicera till alla inblandade att projektet är avslutat

Sist men inte minst, fira det framgångsrika it-projektet!

Checklistan

Checklistan nedan innehåller tre viktiga förutsättningar för respektive framgångsfaktor. För varje tom ruta i ett givet projekt ökar risken markant att de sex kriterierna för ett framgångsrikt it-projekt inte uppfylls.

Acceptans för att det är svårt (FAKTOR NR 1)

- Det finns förståelse för att det är skillnad mellan småskalig hobbyprogrammering och professionell mjukvaruutveckling.
- Det finns förståelse för att mjukvaruutveckling till sin natur är en svår uppgift som kräver både utbildning och erfarenhet.
- Det finns förståelse för att genande i kurvorna (bristande tid för förarbete och förutsättningskapande arbete) i början av projektet riskerar att få allvarliga konsekvenser senare.

Projektledning och projektförutsättningar (FAKTOR NR 2)

- Projektledaren har teoretisk utbildning och praktisk erfarenhet av att leda projekt.
- Det finns ett projektdirektiv som specificerar projektledarens uppgifter, mandat och resurser.
- Det finns både en projektsponsor och en styrgrupp som är engagerade i projektet.

Mod att avbryta och börja om (FAKTOR NR 3)

- Det finns en projektlivscykelmodell med beslutsgrindar och tydliga kriterier som ska vara uppfyllda för att varje grind ska få passeras.
- Det finns rutiner för att snabbt kunna börja om ett projekt med reviderade ingångsvärden.
- Det finns en kultur som uppmuntrar medarbetare att våga prova nya sätt att lösa problem på.

Rätt kompetenser genom hela projektet (FAKTOR NR 4)

- Krav-, VoV-, IT-säkerhets-, användbarhets- och systemarkitekturspecialister medverkar från början och genom hela projektet.
- Projektledaren har identifierat, dokumenterat och säkerställt tillgång till övriga nödvändiga specialister genom hela projektet.
- Samtliga specialister har utbildning och erfarenhet som motsvarar projektets svårighetsgrad.

Gruppen, rummet och verktygen (FAKTOR NR 5)

- Projektgruppen har tillgång till ett dedikerat projektrum.
- Projektgruppen är fysiskt lokaliserad i närheten av varandra och projektrummet, alternativt har tillgång till tekniska lösningar som möjliggör motsvarande närhet och kommunikation.
- Projektgruppen har tillgång till ändamålsenliga verktyg för projektledning, kravhantering och dokumenthantering.

Checklistan, faktorer sex till tio

Beställansvar och rätt kontraktsform (FAKTOR NR 6)

- Stor omsorg läggs vid att utvärdera leverantörens kompetens och förmåga innan avtal tecknas.
- Leverantörsavtal utformas för att möjliggöra avslut vid förutbestämda tidpunkter samt fortsättning med en annan leverantör.
- Beställaren avsätter personal och tid för att svara på leverantörens frågor samt i övrigt vara tillgänglig i den utsträckning det behövs.

Beaktande av hela systemlivscykeln (FAKTOR NR 7)

- Kostnader och personalbehov för användningsfasen beaktas vid kravställning och utveckling.
- Drift- och förvaltningsorganisationerna involveras i kravställning, utveckling, verifiering och validering.
- Systemarkitekturen kännetecknas av låg komplexitet och hög modularitet.

Behoven och kvaliteten i centrum (FAKTOR NR 8)

- Det finns ett sammanhållet systemkoncept som beskriver systemet, inklusive systemgräns och stödsystem, samt antaganden om omvärlden och scenarier för hur systemet kommer att användas i den verksamhet det är till för.
- Beställarsidans fokus ligger på behov och kvalitetskrav snarare än specifika tekniska funktioner¹.
- Det finns aktiviteter och metoder för att löpande validera det framväxande systemet.

Parallell verksamhetsutveckling (FAKTOR NR 9)

- Det finns ett parallellt verksamhetsutvecklingsprojekt eller en uppgift i linjen för att utveckla verksamheten med stöd av de möjligheter det nya systemet erbjuder.
- Verksamheten avdelar tillräckliga resurser för deltagande i it-projektet.
- Både verksamhets- och it-projektet har möjlighet att ändra riktning allt eftersom nya kunskaper erhålls och omvärlden förändras.

Tydligt avslut och överlämning (FAKTOR NR 10)

- Projektet avslutas inte innan drift- och förvaltningsorganisationerna har tagit över ansvaret för it-systemet.²
- Alla avslutande aktiviteter, inklusive "städning" och avslutande erfarenhetsåterföring, genomförs *innan* projektet avslutas.
- Projektmedlemmar och intressenter får tydliga besked om att projektet är avslutat och att inga ytterligare aktiviteter kvarstår.

¹ <https://www.svenskkravterminologi.se/terminologi#overflodig-kravtyp>

² Givetvis under förutsättning att projektet inte avslutas i förtid.

Om författaren

Essän är skriven av Patrik Sternudd. Patrik är civilingenjör i informationsteknologi och har mångårig erfarenhet från it-branschen. Mer information finns på Patriks hemsida <https://patrik.sternudd.name>.

Essän har även en elak tvilling som behandlar samma tema, men från "andra hålet":

Det skenande kritiska projektet – en lathund för att behålla it-verksamheten i den digitala stenåldern

Båda essäerna finns tillgängliga på Patriks hemsida.

Avslutningsvis vill författaren skicka ett stort tack till de kunniga och kloka personer som bidragit med synpunkter och korrektur på innehållet. Ert engagemang har gjort essän bättre!

Dokumentversion 2019-08-23

Upphovsrätt och regler för spridning

Patrik Sternudd äger upphovsrätten till detta verk. Bilden på framsidan används med licens från Shutterstock.com.

Det är tillåtet att sprida verket under följande förutsättningar:

1. Spridningen av verket sker i sin helhet och utan förvanskning eller bearbetning. Det är alltså inte tillåtet att sprida delar av verket eller förändringar av det. Följande undantag gäller dock:
 - (a) Citeringar får naturligtvis göras enligt gällande praxis.
 - (b) Det är tillåtet att för internt bruk, till exempel inom en organisation eller ett projekt, sprida en eller flera utskrivna sidor av verket.
2. Spridningen syftar direkt eller indirekt till att skapa bättre it-system i Sverige eller världen, eller på annat sätt förbättra samhället. Detta inkluderar professionellt utförande av arbetsuppgifter inom relaterade områden.
3. Spridningen sker utan vinstsyfte. Det är inte tillåtet att sälja verket eller införliva större delar av det än vad som är normal praxis för citering i något annat verk eller material som säljs.

Copyright

Copyright Patrik Sternudd. You may not reproduce or distribute this document under other circumstances than what is permitted by the Swedish terms specified in the section directly above.