

## 2.1 Prosjektledelse

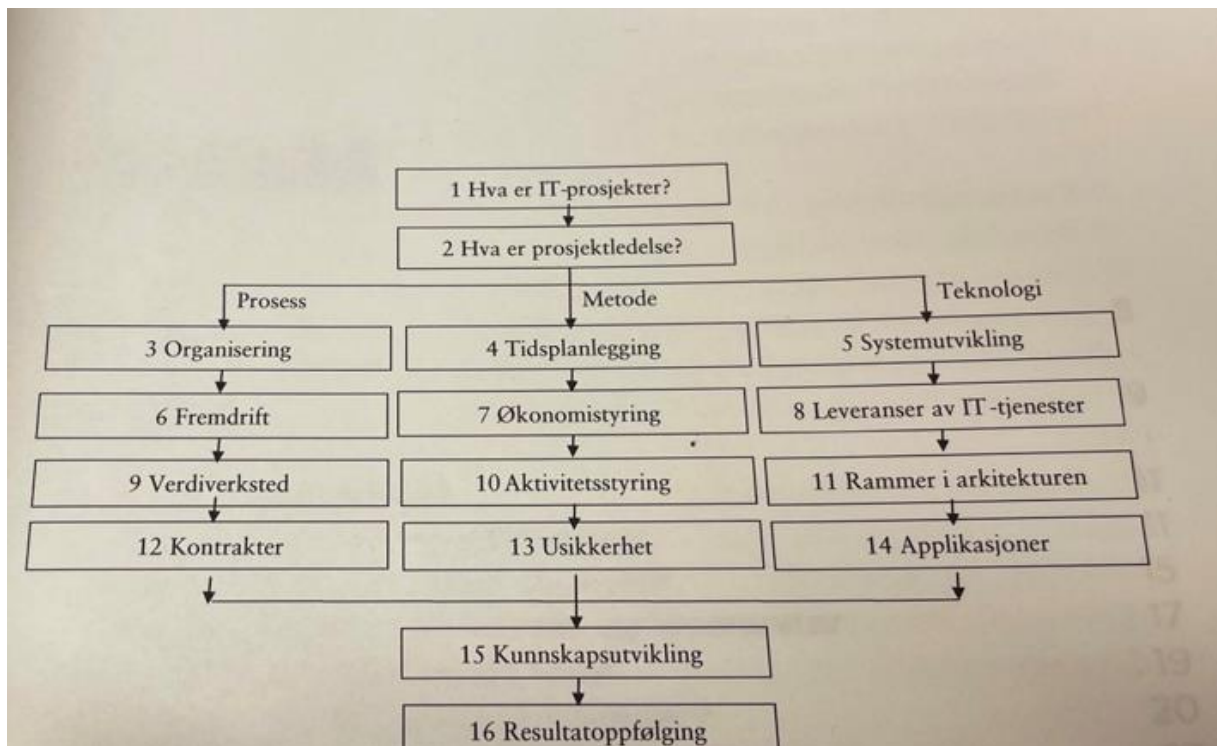
Hvordan få finansiert, organisert og besluttet prosjekt med stor grad av usikkerhet og høy risiko.

Vi forutsetter i dette kompendium at vi ikke snakker om standard integrasjoner som skal installeres eller settes opp, men løsninger som inneholder grader av utvikling eller script. Selv om en standard-integrasjon også kan være ett krevende prosjekt, er det spesielt utvikling som skaper usikkerhet, risiko og utfordringer.

Bok «IT og Prosjektledelse» av Anne og Petter Gottschalk har flere gode beskrivelser av sentrale elementer i IT Prosjekt. Vi skal ta for oss deler av boka, mens vi retter fokus på viktige elementer innenfor integrasjoner. Når en virksomhet jobber med et stort prosjekt, er fokuset ofte overdrevent rettet mot å finne den tekniske løsningen. Planlegging er nemlig en minst like viktig del av prosjektet, og da snakker vi ikke bare om en aktivitetsplan, men prosesser, metodikk og teknologi, samt formål og resultatoppfølging.

Usikkerhet er prosjektets største utfordring. Derfor anbefales risikoanalyse som viktig beslutningsgrunnlag, både før oppstart og gjennom hele prosjektutførelsen.

Dette kapitlet tar for seg 16 av elementene fra boka «IT og Prosjektledelse», og kommenterer sammenhengen mellom bokas tema og integrasjonsprosjekter.



### 2.1.1 IT-Prosjekter

Integrasjonsprosjekt er tiltak som iverksettes for å forandre bruken av informasjonsteknologi. Det er et gangs foretakende med gitte mål og avgrenset omfang, bestående av aktiviteter som er begrensninger med hensyn til tid og kostnader. Generelt har et prosjekt en bestemt start og bestemt slutt, bestående av faser, som utvikler seg progressivt og skal nå leveranser eller andre målsettinger.

Prosjekter kan ha mange formål og resultater:

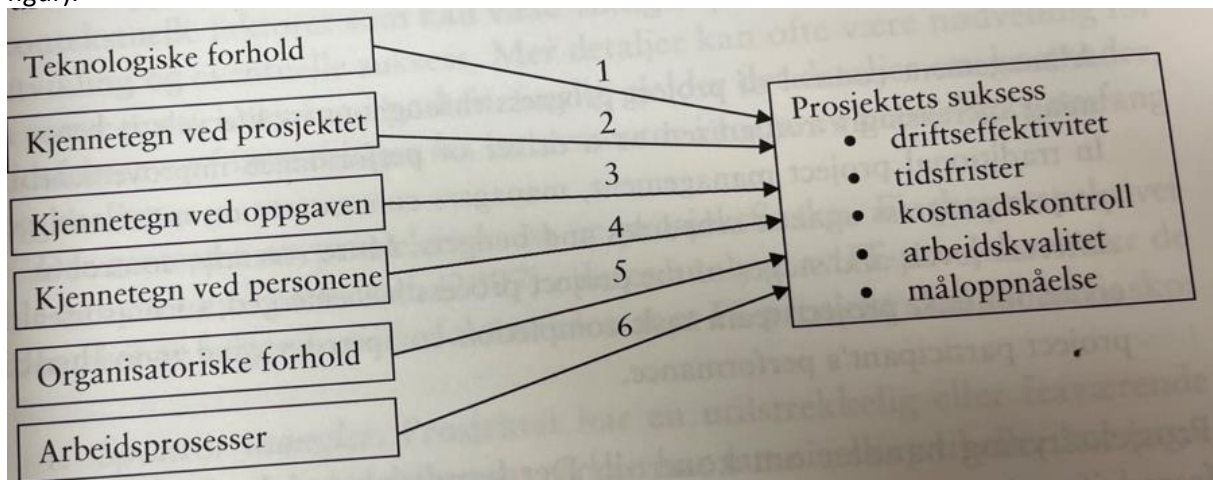
- *Oppgaveresultat*  
I et teknologisk perspektiv skal prosjektet produsere noe som er nyttig og fungerer bra
- *Prosessresultat*  
I et sosialt perspektiv skal prosjektet produsere prosjektdeltakere som er fornøyd med resultatet og som har økt sin kompetanse
- *Organisasjonsresultat*  
I et organisatorisk perspektiv skal den nye teknologiske løsningen bidra til økt verdiskapning i virksomheten.

Enkel faseinndeling:

- *Oppstart*  
Mål, krav, oppgaver og ansvar
- *Planlegging*  
Tidsplaner, budsjett, ressurser, bemanning og risikostyring
- *Gjennomføring*  
Statusrapporter, endringer, kvalitet og prognose
- *Avslutning*  
Opplæring, dokumentasjon, frigi ressurser og erfaringsoverføring

## Suksessfaktorer

Boka «IT og prosjektledelse» viser til Aladwan (2002) sin analyse av IT-prosjekter hos 84 virksomheter. Seks faktorer viste seg å være avgjørende for hvordan det gikk med prosjektene: (ref figur).



Prosjektets suksess blir målt gjennom driftseffektivitet, tidsfrister, kostnadskontroll, arbeidskvalitet og måloppnåelse.

1. *Teknologiske forhold* målt gjennom teknologier og metoder som har nytteverdi både for det aktuelle prosjektet, operativ drift og nye prosjekt.
2. *Kjennetegn ved prosjektet* har avklarte mål og midler.
3. *Kjennetegn ved oppgaven* har ett avklart omfang. Det er på forhånd avklart tydelig hva oppgaven dreier seg om
4. *Kjennetegn ved personene* er kompetanse hos deltakere, leverandører og mottakere.
5. *Organisatoriske forhold* dreier seg om ledelsens sitt engasjement i prosjekt.
6. *Arbeidsprosesser* dreier seg om evnen til å løse problemer.

Analysen til Aladwani viste at arbeidsprosesser (6), tekniske forhold (1) og organisatoriske forhold (5) var mest avgjørende for suksess. For ett integrasjonsprosjekt, hvor utfordringer må løses underveis og en har et uklart bilde i starten, er det svært avgjørende hvilke prosjektdeltakere og samarbeidspartnere en velger. Etter dette avgjøres prosjektets suksess av at det benytter best egnet teknologi, og at beslutningstakere har eierskap.

Skal en få beslutningstakere, «stakeholders», som er engasjerte og motiverte for å bidra, må de oppleve at de faktisk kan kontrollere prosjektet. De må få servert utfordringer, konsekvenser, alternativer og løsningsforslag fremfor teknisk(e) problemstilling(er). Aktiv bruk av risikoanalyse, det vil si hvordan avvik påvirker kvalitet, fremdrift og kostnader. Ledelsen må ta ansvar for avvik hos leverandører og legge til rette for at prosessendringer i virksomheten kommer på plass. Prosjektstyring vil dermed handle om kontroll på fremdriften i prosjektet, hvor det som vektlegges som fremdriftsparameter kan variere fra prosjekt til prosjekt.

## Fiaskoprojekt

Det motsatte av suksess er prosjektfiasko, hvor tilsvarende analyser av fiasko kjennetegnes med:

### 1. *Sponsorer mangler*

*Prosjektet har en utilstrekkelig eller fraværende støtte fra toppledelsen som verken har tatt eierskap til eller har engasjert seg i nødvendige ressurser til prosjektet.*

Dette er en utfordring da integrasjonsprosjekt gjerne blir en fjern og teknisk sak for ledelsen og brukerne, en abstrakt løsning ingen ser og opplever på annen måte enn ved driftsstans og alvorlige feil. Ett BI (Business Intelligens) eller kundesystem (CRM) har fancy rapporter og brukervennlige oversikter som engasjerer beslutningstakere. Derfor må effektivisering, kvalitet og nytteverdier fundamenteres hos sponsorer ved investering i tekniske løsninger.

### 2. *Ressurser er fastlåst*

*Hele prosjektet må planlegges på forhånd, og budsjettet må overholdes uansett hva som skjer underveis.*

Når risikoen for sprekk på tid og kostnader er svært høy for integrasjonsprosjekt, blir dette en nesten håpløs situasjon i et prosjekt der utfordringer og løsninger i stor grad må tas fortløpende. Best er kanskje å ikke iverksett prosjektet under låste rammebetingelser.

### 3. *Kompetanse mangler*

*Prosjektet mangler deltakere som har nødvendig kunnskap, ferdigheter og håndtere prosjektet i praksis.*

Dette gir seg stort utslag når nye forutsetninger oppstår, teorier som viste seg ikke å fungere. Avvik i kvalitet og tidsplan for løsningen fra leverandører og endring av behov oppstår fortløpende i integrasjonsprosjektet. Den som skal lede prosjektet slikter ved manglende teknisk forståelse og ikke minst at aktørene ikke forstår konseptet i løsningen.

### 4. *Omfanget er uavklart*

*Kravspesifikasjoner blir ignorert fordi teknologien allerede er bestemt.*

Selv om nyere teknologi it-strategisk er å foretrekke, vil fastlåste rammebetingelser ødelegg for funksjonell løsning. Overdreven sikkerhet en i praksis ikke trenger er eksempel på teknologi som kan hindre løsninger som kan ha større verdi enn konsekvensene ved at sikkerhetsbrudd.

### 5. *Dynamikk mangler.*

*Endringer underveis er nesten umulig.*

Typisk pga fastpris-kontrakter, basert på kravspekk som låser løsningen. Havner i konflikt med leverandører som ikke kan tilpasse løsninger på kort sikt eller avvike fra planlagt løsning.

## 6. Relasjoner mangler

*Prosjektet mislykkes i å tilfredsstille behov for informasjon og dialog.*

Når oppdragsgiver, ledelse eller kunde mangler tillit til prosjektet og forståelse for uforutsette situasjoner som oppstår, blir det vanskelig å finne alternative løsninger, gjerne betegnet som «workarounds». Hvis bakgrunn for reduksjon i kvalitet/funksjonalitet, kostnadssprekk og forsinkelser ikke blir godt nok kommunisert, vandrer prosjektdeltakerne i et minefelt av farer for intern og ekstern konflikt. Evnen til å forutse og forhindre fiasko forutsetter kompetanse og erfaring.



### 2.1.2 IT-Prosjektledelse

Viktigste personen i et prosjekt er prosjektlederen. I prosjektet vil denne personen ha flere roller; Vedkommende vil for eksempel være den som tar beslutninger, skaffer ressurser, motiverer deltakerne, osv. Oppgavene dreier seg om å planlegge, budsjettere, organisere teamarbeid, oppfølging, iverksette endringer, og kommunisere med aktører/interessesentre. Satt på spissen, skal en god prosjektleder for ett krevende prosjekt skaffe til veie ressurser en gjerne ikke har til disposisjon når det virkelig trenges.

*«Et hypotetisk eksempel er at du skal lede en fødsel som viser seg å kreve keisersnitt, men tilgjengelig kirurg skal følge sin datter opp kirkegulvet. Du skal som prosjektleder overbevise ham om at jobben der og da er viktigere, sørge for at vielsen går sin gang uten far til stede og at fødselen blir vellykket».*

En prosjektleder kan med andre ord ikke alltid forvente å være populær hos alle til enhver tid.

Det er ingen fasit, men det finnes ulike teorier om hva som kjennetegner en god prosjektleder for integrasjoner; være seg Posisjon (stilling), Personlighet (karisma, intelligens samarbeid, osv.), Fagorientert eller Situasjonorientert. *«Satt på spissen skulle den ideelle Prosjektleder for integrasjon vært en person med dr.grad innen programvareutvikling, ledelse og psykologi, samt erfaring fra forskjellig prosjektstillinger i 10 virksomheter og vært 25 år gammel».* Behovet for fagkunnskap hos prosjektleder avhenger av størrelse på prosjektet og kompetanse i resten av prosjektteamet. Er det snakk om 100-1000 timers prosjekter, bør prosjektleder hos utførene part ha sterk kompetanse innen programvare-utvikling, helst programmert en del selv, kunnskap om integrasjonsteknologier og evne til å håndtere risiko-elementet i ett prosjekt.

Mottakende part, intern/ekstern kunde, har gjerne også en innkjøper eller prosjektleder, hvor kompetanse, samarbeidsevne og risikohåndtering kan være like viktig for suksess som hos leverandøren. Prosjekter som er suksess i én virksomhet, kan like godt møte veggen i en annen, men tilsvarende virksomhet innenfor samme bransje.

Lederstiler kan være instruerende, rådgivende, deltakende, delegerende, osv. Innen IT-prosjekter vil det være behov for prosjektleder som kan variere mellom de forskjellige lederstilene fra situasjon til situasjon. Det skyldes at prosjekter består av en endringspreget hverdag hvor det oppstår nye arbeidsoppgaver de ikke kjenner, og at motivasjonen av ulike grunner går opp og ned. For å lykkes med situasjonsbestemt ledelser er det blant annet viktig at prosjektlederen forstår deltakernes situasjon og kompetanse. Prosjektlederen må ikke ha holdninger som forhindrer endringer i lederstil, men derimot kunne vise fleksibilitet og tilpasningsevne. En vanlig utfordring for integrasjonsprosjekt er at leverandør av kritisk API-komponent ikke kan levere det som er avtalt, være seg på tid eller kvalitet. Evnen da til å presse leverandøren, i noen tilfeller gå bakveier eller gjennom teamet finne kreative alternativer (workarounds) vil ofte være prosjektleder sin suksessfaktor.

### 2.1.3 Organisering

Organisasjonsstrukturen, dvs. måten aktører og deltakere er satt sammen og bygd opp på, har som formål å sørge for at aktiviteter blir koordinert og samstemt, samtidig som aktørenes handlinger blir kontrollert. Strukturen består av alle de reglene og rutinene som er satt opp i prosjektorganisasjonen, som kan deles inn i 3 elementer:

- *Kompleksitet* avgjør graden av oppdeling og differensiering av oppgaver
- *Formalisering* avgjør av spesifisering og definering av arbeidsoppgaver gjennom regler og prosedyrer
- *Fordeling* er graden av beslutningsmyndighet sentralt og desentralt



Integrasjon er i alle hovedsak et kunnskapsbasert prosjekt, hvor byråkrati og krevende beslutningsprosesser kveler kreativiteten. På den annen side, er de krevende arbeidet en forutsetning for å kunne få tilgang til ressurser, kapital og gjennomslag for beslutninger som er avgjørende for løsning, fremdrift og kostnader.

### 2.1.4 Tidsplanlegging

Planlegging handler først og fremst om å bestemme retning til prosjektet, slik at deltakerne vet hva de skal gjøre, når de skal gjøre det og hvilke ressurser som skal brukes for å levere en suksessfull løsning. For mottaker/kunden er tiden ofte kritisk, prosjektet må bli ferdig innen en gitt dato, for eksempel produksjonssetting av netthandel eller nytt ERP-system. Deltakelse i planleggingsprosessen skaper både engasjement, motivasjon, forståelse, løsninger og forpliktelser hos deltakerne.

Hensikten med å legge tidspress på integrasjonsprosjektet kan være å:

- sikre at leveransen blir levert innen en tidsfrist som kan forsvare utgiftene
- ha grunnlag for å styre mot fastsatte mål
- få bedre innsikt i oppgavene som skal utføres
- oppnå at arbeidsoppgaver og ressurser samordnes, koordineres og avtales
- sikre beste mulig ressursutnyttelse
- få økte muligheter til å se konsekvensene ved endrede forutsetninger
- skaffe bedre grunnlag for diskusjoner for de som blir berørt
- bedre mulighet for å motivere prosjektets medarbeidere

Når en skal velge mellom kvalitet, tid og kostnad, er det ett faktum at tid og kostnad sprekker for integrasjonsprosjektet fordi faktorene er ukjent, mens løsningene kun kan benyttes dersom kvaliteten er over et minimumsnivå. Hvis for eksempel ikke grunndata-synkronisering mellom ERP og Netthandel er av høy kvalitet, har en ordre-overføringen liten verdi. At integrasjonen kommer på plass kan være en forutsetning for at hovedprosjekter kan gjennomføres. Dermed blir en nærmest tvunget til å akseptere tid- og kostnadsoverskridelser.

Endringer i forutsetning underveis, eller svikt i leveranser, kan få så dramatiske følger at det ikke lenger er mulig å fastsette en realistisk tidsplan eller ferdigdato før en har kontroll på situasjonen. Slik sett er bruk av Milepæls-plan ofte bedre egnet i denne type prosjekter. Da tradisjonell tidsbestemt aktivitetsinndeling har liten hensikt i et dynamisk og gjerne kreativt integrasjonsprosjekt, bør en søke andre metoder for planlegging, beskrevet i eget kapittel 2.2 Overordnet planlegging.



### 2.1.5 Systemutvikling

Systemutvikling er metodisk og målrettet arbeid for å lage informasjonssystemer, hvor det handler om å utvikle et system basert på spesifikasjoner eller behov. I dette kompendiet ser vi ikke på «plug & play» standard-integrasjoner, men løsninger som inneholder en grad av utvikling i form av programmering, være seg script eller tilpasninger av nye funksjoner, moduler eller nye applikasjoner. Det er utvikling, enten i selve integrasjonsløsningen, eller i form av utvidelser av grensesnitt/API hos system-leverandøren, som er en av hovedårsakene til at integrasjonsprosjekt har høy risiko. Risikoen kommer også av at grensesnitt/API hos systemleverandører kan være premature løsninger med lange leveringstider, samt abstrakt kunnskap som er begrenset i mange virksomheter og hos systemleverandører.

Systemutvikling omfatter flere aktiviteter:

- *Problemanalyse* er arbeidet med å forstå problem som skal løses og muligheter med ny teknologi
- *Kravarbeid* er å fastlegge kravene til løsningen i form av funksjoner, når og hvilke data skal utveksles
- *Utforming* går ut på å bestemme overordnet konsept og systemstruktur, samt bryte strukturen ned i delsystem, moduler eller funksjoner.
- *Realisering* går ut på å skape den tekniske løsning gjennom design og programmering
- *Implementering* går ut på å sette løsningen i drift, gjerne med parametersettinger
- *Testing* går ut på å sjekke at løsningen oppfører seg som planlagt, inkludert volum og unntak
- *Produksjon* er overgangen til at løsningen er i operativ bruk for pilot eller alle brukere.
- *Modifisering* innebærer egenskaper og funksjoner i løsningen som må justeres ut fra praktisk bruk.



Bruk av tradisjonell Fossefallsmetodikk, tatt i fra bygg- og anleggsbransjen, hvor en sekvensielt følger aktivitetslisten for komplett løsning, blir ansett som den største feilen en kan gjøre ved utvikling av løsninger hvor en ikke er i stand til å detalj-spesifisere kravene. Utvikling av integrasjonsløsninger er en prosess hvor aktivitetene fra problemanalyse til testing, i noen tilfeller produksjon og modifisering, må kunne skje fortløpende i prosjektet. Katastrofe-prosjektene kjennetegnes ofte ved at spesifikasjoner er sendt til lavkostland for programmering.

Mer om utviklingsmetodikk i kap 2.6.

### 2.1.6 Fremdrift

Et prosjekt må følges opp regelmessig i utførelsen slik at prosjektledelse, eventuelt styringsgruppe og ledelse/oppdragsgiver/sponsor, er oppdatert på avvik som kan gi utslag på tid, kostnader eller kvalitet. I en integrasjonsløsning er det gjerne en liten brikke i ett stort puslespill som kan være helt avgjørende, også kalt «*showstopper*». Denne typen avvik kan medføre at hele prosjektet må settes på vent eller kanselleres, uansett når og hvor den oppstår. Konsekvensene må analyseres og risikohåndteres, hvor normale fremdriftsplaner og status gir liten mening.

Utfordringen for de som skal ta beslutninger i forhold til prosjektet er å forholde seg til adferden og måten prosjektleder rapporterer på fremdrift, avvik og status. Er prosjektleder selektiv i rapporteringen, enten optimistisk eller pessimistisk i forhold til den faktiske situasjonen, kan det ende med feil beslutninger. Optimistisk rapportering benyttes gjerne for å skape entusiasme, beholde ressurser og få følgeprosjekt, mens en pessimistisk rapportering benyttes for å få støtte i vanskelige saker og flere ressurser. Prosjektleder kan spille på begge strengene fra sak til sak, men må gi full alarm når en ser at prosjektet kan gå på et skjær slik at ledelsen skal ha muligheter får å styre unna en katastrofe.



Typiske saker er at teorier, komponenter eller teknologier som var forutsetninger for at prosjektet skulle fungere ikke ble levert eller levert som forventet. Videre er det typisk at aktører har misforstått løsningen eller har ikke funksjoner som dekker behovet. Eksempel kan være at prosjekt-teamet trodde løsningen kunne ta imot på et meldingsformat det viser seg løsning ikke støtter, eller at innholdet i meldingen mangler avgjørende informasjon. Det kan være at ordretransaksjonen mangler et bestillingsnummer, hvor behovet for nummeret fremkom først under pilot-produksjon, dvs. at det ikke ble avdekket i løsningsbeskrivelsen. Prosjektleder sin evne til raskt å kunne løse opp i slike krisesaker er ofte den viktigste egenskap til den som skal styre integrasjonsprosjekt.

### 2.1.7 Økonomistyring

Kostnadshåndtering er en stor utfordring i prosjekter, spesielt for integrasjonsprosjekt. Slike prosjekter handler først og fremst om kostnader da det ikke skapes inntekter eller besparelser mens det pågår, med mindre det kan splittes i faser som produksjons-settes. Kostnader måles i penger og betales for å få tilgang til ressurser, slik som arbeidskraft, utstyr og programvare-lisenser.

En sentral del av prosjektledelse er styring av kostnader, som omfattes av fire prosesser:

- *Ressurserestimering* går ut på å bestemme hvilke ressurser som trengs, og i hvilket kvantum for å kunne planlegge.
- *Kostnadsestimering* går ut på å bestemme grove anslag på kostnader for de forskjellige ressursene som trengs i prosjektet, hvor resultatet av prosessen er kostnadsestimater.
- *Strukturering* av kostnadene handler om å tildele kostnadsestimater i henhold til valgt prosjektinndeling, hvor resultatet av prosessen blir en kostnads-basis som grunnlag for finansiering og beslutninger.
- *Kostnadskontroll* omhandler kostnadsstyring i form av oppfølging av påløpte kostnader, avvik, reviderte budsjett og håndtering av kostnadsavvik.



## Innkjøp

Behovet for innkjøp avhenger av hva en må anskaffe av ressurser, utstyr, materiell og arbeidskraft fra interne kostsenter eller leverandører i form av innkjøp og kontrakter. Eksempel er server-kapasitet, lisenser og konsulenter.

## Estimering

Om tidsplanlegging er krevende, er estimering av timeforbruk og budsjettering enda verre når en skal utvikle løsninger med ukjente faktorer. Av metodikker som er forsøkt, er det som regel å lytte til utviklere med lang erfaring som fungerer best. De angir da gjerne ett best case estimat og «*multipliserer dette med PI*», forutsatt at alt går som normalt.

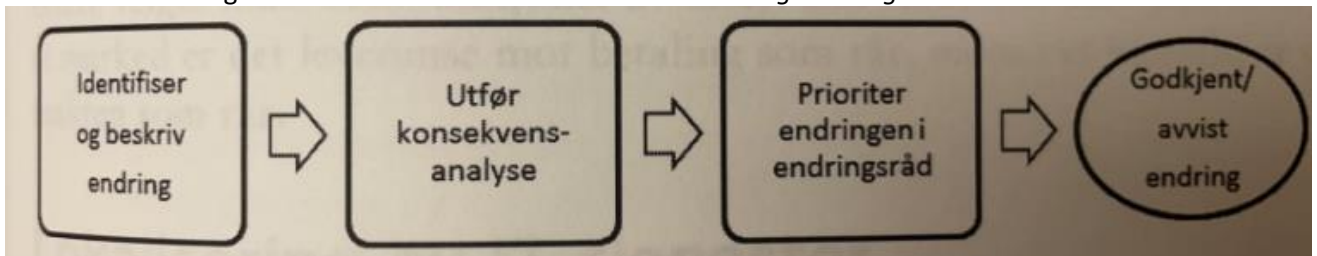
Ut over «skudd fra hofta estimat» fra de med lang erfaring, fins det metoder for estimering hvor de mest vanlige er følgende:

- *PERT*. Forventet tid = optimistisk \* 4 + pessimistisk / 4
  - $15 * 4 + 40 / 4 = 60 + 10 = 70$  timer
- *Estimeringsmodeller*. Disse modellene lages gjerne basert på designdokument, hvor en teller antall prosesser, grensesnitt, funksjoner, osv. som blir berørt, samt vektig av disse
- *Estimering basert på erfaring*. Her kan erfarne ressurser estimeres, men hvis oppgaven senere blir utført av uerfaren ressurs må omfanget re-estimeres.
- *PI* Grovt estimert + PI
  - $20 * PI = 15 * 3,14 = 47,1$  timer

## Kostandsoppfølging

Prosjektleder må ha jevnlig oppfølging av kostnader i prosjektet, normalt med rapportering til styringsgruppe/ledelse/kunden hver måned på totalkostnader. Den mest variable delen av kostnadene skyldes de uforutsigbare timeforbruket, som må følges opp fortløpende, gjerne ukentlig godkjenning. Timene kan iht. valgt prosjektinndeling rapporteres som opprinnelig estimert, re-planlagt, påløpt og gjenstående, registrert i Excel eller Time/Prosjektsystem (JIRA, e.l.)

Avvik fra prosjektplan kan oppstå pga. kvalitet, endrede priser på planlagt innkjøp, utvidet omfang eller feil-estimering. Prosessen med å håndtere avvik kalles *Change Management*



Endringsønsker ut fra nye funksjoner, krav, muligheter, osv. som kommer underveis fra oppdragsgiver, vil normalt medføre økte kostnader og økte inntekter. For å holde rede på dette og sikre inntekter på endringene må prosjektleder opptre som selger, noe som igjen gir muligheter for god fortjeneste på et prosjekt med stramt budsjett eller fastpris. Samtidig, må prosjektleder kanskje opptre som innkjøper også, dersom endringene medfører økte kostnader fra system/underleverandør. Også her benyttes begrepet PI, dvs. multiplisere tilbudsprisen med ca. 3,14 da kunden

ikke har bedre alternativ. I ett integrasjonsprosjekt blir det ofte som mye endringer underveis at en fastpris er uheldig for alle parter.

En annen utfordring, vil være å avtale endringsordrer i en pågående prosess hvor en har fastpris/estimer basert på tillit eller måltall (delingsmodell på +/- avvik). Hvis ikke en får avsluttet og delvis godkjent planlagt løsning før en starter på endringen, blir det vanskelig å avgjøre om feil og overskridelser skyldes opprinnelig løsning eller endringen. På den annen side, vil det å fullføre en uønsket løsning også være uheldig for alle parter. Mer om dette under 2.5 Systemleverandører.

### 2.1.8 Leveranse

Det er ikke lenger bare konsulenttimer, hardware og software som leveres av IT-leverandørene, men også en flora av andre tjenester innen for eksempel drift av all infrastruktur, kommunikasjon og integrasjonsportaler. Integrasjonsløsningen må forholde seg til disse tjenestene og aktørene, hvor API må installeres hos driftsleverandøren etter å ha godkjent sikkerhetsløsningen.

Eksempel på slike SKY eller driftstjenester kan være:

- *Infrastructure as a Service (IaaS)* – alt av utstyr for å støtte opp under driften leveres av en ekstern leverandør, hvor kunden betaler for bruk og omfang.
- *Platform as a Service (PaaS)* – virksomheten outsourcer utrulling og oppgradering av programmer, hvor rammeverket for å utføre dette leveres av en leverandør.
- *Software as a Service (SaaS)* – det de fleste anser som en fullverdig sky-tjeneste, hvor kunden betaler for bruken av programmene i form av antall transaksjoner eller bruker-lisenser
- *Integration Software as a Service (ISaaS)* – rammeverk og driftsløsning for integrasjoner, normalt API til API
- *Hybrid* – kombinasjoner av disse tjenestene

### 2.1.9 Verdiverksted

En bedrift skaper verdi for sine kunder og eiere ved utføre aktiviteter. Bedriftene skiller seg fra hverandre i hva de gjør ved at de har forskjellig verdiskapningslogikk, delt i:

- *Verdikjede* - klassisk produksjon ala Glamox
- *Verdiverksted* – løser kunders problem ved å være samarbeidsdyktig og tilpasse sine aktiviteter og teknologi, ala Radiumhospitalet
- *Verdinettnettverket* – utfører formidlingstjeneste ala Telenor

IT-tjenester handler mye om Verdiverksted i form av problemforståelse, løsningsalternativer, løsningsvalg, løsningsgjennomføring og resultatevaluering. I et integrasjonsprosjekt er det fatalt hvis aktører og deltakere ikke har forstått problemstillingen, målsetning og løsningskonseptet. Når systemleverandører bytter ressurser i ett prosjekt eller flytter oppgaven til andre enheter, vil det gjerne medføre katastrofale konsekvenser hvis de som overtar ikke har forstått løsningen, noe en tar for gitt.

### 2.1.10 Styring av IT-prosjekt

Med «styring av arbeid i prosjektet» menes at informasjon må samles inn om fremdrift, beslutninger må tas og tiltak må iverksettes for å sikre at fremdriften holdes, og løsningene blir i henhold til oppdaterte forventninger. Styringen må sikre at mål nås og holde kursen slik at avstand mellom plan og virkelighet blir minst mulig. Styring er en kontinuering prosess som vil pågå til prosjektet er avsluttet.

Gjennom rapporter, oppfølgingsmøter og samtaler med prosjektdeltakere får prosjektleder informasjon om status, hvor fokus må ligge på avvik fra planer og tilhørende tiltak. Prosjektleder må avgjøre om tiltakene eller hindringer krever avklaring hos styringsgruppe, ledelse, leverandør eller kunde. I ett krevende integrasjonsprosjekt er dette normen, ikke noe som *kan* oppstå, men *vil* oppstå.

Det en da står overfor er vanligvis negativt avvik i tid eller økte kostnader, som gir følgende valg-alternativer:

- *fortsette prosjekt som før*
- *endre tidsfristene*
- *senke ambisjonsnivået*
- *velge andre alternative løsninger*
- *forenkle, omgå problemet (workarounds)*
- *tilføre nye ressurser*
- *sette bort arbeid*
- *jobbe mer effektivt*

Det viktig for en prosjektleder er at mest mulig av denne typen beslutning er tatt av andre, men gjerne basert på utredning og anbefaling av prosjektgruppa. Bruk av risikoanalysen er et godt verktøy for en slik utredning, mens en ser konsekvensene i et totalperspektiv.

Endringer i prosjektets omfang kan ha positive og negative følger. De positive følgene kan være løsninger som er billigere, mindre tid-krevende og har høyere kvalitet. Får prosjektet derimot negative følger, kan det bety høyere kostnader og forsinkelser. For prosjektleder blir det en stor utfordring dersom det ikke er lagt inn rikelig med rom for endringer i tidsplan og rammer for kostnader, gjerne satt av som budsjettpost for tillegg eller funksjoner som kan salderes i løsningen («nice to have»).

For å redusere effekten av endringene kan det etableres et system for endringskontroll med følgende innhold:

- *Opprinnelig løsning, plan eller status som endringer refererer til*
- *Endrings skjema eller aktivitet (issue)*
- *Prosedyre for evaluering av konsekvensen av endringen*
- *Beslutningsorgan som godkjenner, finansierer eller forkaster endringen*
- *Prosedyre for å informere berørte parter om endringene*
- *Oppfølging og kontroll av iverksatte endringer*

### IT-Verktøy

Projectplace startet i 1998, og var en av verdens første SaaS-løsninger med versjonsstyring av dokumenter og program-kode som kan deles blant aktørene.

GITHUB og GITLAB er tilpasset programvare-prosjekter, inkludert enkel planleggingsfunksjon. I kap. 2.2 viser vi integrasjonsprosjekt planlagt og styrt med GITHUB.

### 2.1.11 Rammer i strukturen

Arkitekturen setter rammene for hvordan et IT-system skal bygges, hvor løsningen er utformet av to typer arkitekter:

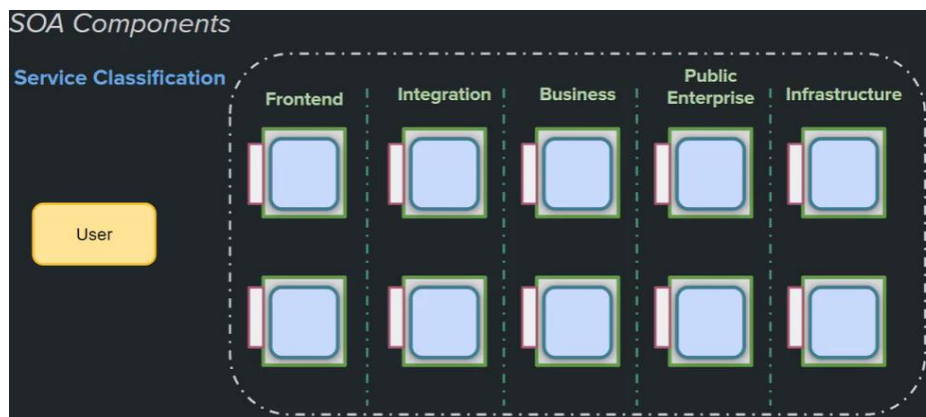
- *Teknisk arkitektur – tar stilling til teknologivalg e.l.*
- *Funksjonell arkitektur – har oversikt over hva system skal utføre*

I et integrasjonsprosjekt må av og til få innsikt i funksjonelle deler av system som er berørt av løsningen, for eksempel se hvordan grunndata blir vedlikeholdt eller ordrer blir opprettet i et ERP-system. Skjerm bilde-print av et kunderegistreringen kan være godt nok, samt hvilke innhold i oppslagstabeller (valuta, land, område, osv.)

Den tekniske arkitekturen er avgjørende for hvordan integrasjoner skal implementeres, hvor integrasjonsgrensesnitt er spesielt viktig. Hvis systemet ikke har egnede grensesnitt for dette må en analysere arkitekturen som da kan bestå av følgende:

- *fysisk lag – bokser, nettverk, etc*
- *logisk lag – SQL-database med kun tabeller eller SQL-logikk (triggere eller prosedyrer)*
- *applikasjonslag – forretningslogikken*
- *presentasjonslag – brukergrensesnittet*

System som gjenspeiler applikasjonslaget i integrasjonsgrensesnitt, aller helst i et synkront REST-API, er et utmerket utgangspunkt. Endringer i applikasjonslaget vil da kunne være tilgjengelig til flere formål, samt at versjonshåndtering er automatisk ivaretatt. Dessverre er systemer pr dato så gode system, hvor grensesnittene derimot er egne løsninger rett mot databasen eller via meldings-hub, noe som gir en langt svakere kvalitet på integrasjonen.



Service Oriented Architecture (SOA) kalles på norsk tjenesteorientert arkitektur, som kjennetegnes ved at komponentene i arkitekturen er løst koblet. Det betyr at de kommuniserer med hverandre over en Enterprise Service Bus (ESB) som sørger for at meldingene blir synkronisert og kommer frem til riktig komponent. Dette gir muligheter for å bytte ut komponenter i en applikasjon uten å endre på arkitekturen. Integrasjonsgrensesnittet kan typisk være en slik komponent. Et eksempel på en funksjon kan være *hentLeveringsAdresser* som kun henter adressen fra databasen, mens funksjonen *hentKundeinformasjon* henter kunde-nr, navn, postadresse, telefon og leveringsadresser. Dette er da forretningslogikk som også kan være innbakte funksjonalitet i API-et.

Når en skal anskaffe system som også skal integreres, er arkitektur etter hvert blitt ett av de aller viktigste kravene, men det å få innsikt i dette kan ta tid da ikke alle leverandører er åpne på saken.

### 2.1.12 Håndtering av kontrakter

I de fleste integrasjonsprosjekter vil det forgå former for anskaffelser, enten i form av resurser, API, integrasjonsrammeverk, infrastruktur eller komplette systemløsninger inkludert i API. Jo større prosjektet er, desto mer av prosjektet vil da normalt bli rene innkjøp. I de aller største prosjektene i Norge vil normalt alt bli anskaffelser, der gode kontrakter og evnen til å håndtere leverandører blir største utfordring for prosjekt-eier. Også prosjektleder kan i slike prosjekt være innleid.



Mer om dette i *kap. 2.5 Systemleverandører*



### 2.1.13 Håndtering av usikkerhet

Usikkerhetshåndtering dreier som om å jobbe proaktivt med hensyn til problemer og uforutsette hendelser som kan oppstå. Betydningen av dette blir noe av det viktigste i et integrasjonsprosjekt hvor vi som nevnt mange ganger opplever at prosjektet består av en rekke usikkerhetsfaktorer.



Mer om dette i *kap 2.4 Risikohåndtering*

### 2.1.14 Anskaffelse av integrasjonsverktøy

I mange integrasjonsprosjekt blir anskaffelse av applikasjoner, rammeverk og API en sentral del av prosjektet, slik at en blir nødt til å gjøre utvelgelser på linje med øvrige innføringer av nye løsninger i virksomheten.

Overordnede faser i ett slikt løp kan være:

- *Analyse av behovene*
- *Kartlegge leverandører*
- *Utvelgelse av leverandør*
- *Oppfølging av leverandør*
- *Innføring av løsningen*



Behovene er et resultat av planlagt løsning, hvor gjerne aktuelle leverandører har kommet med innspill og anbefalinger. Utfordringen her ligger i å styre eventuelle utvidelser og tilpasninger for at løsningen skal kunne fungere i planlagt integrasjon.

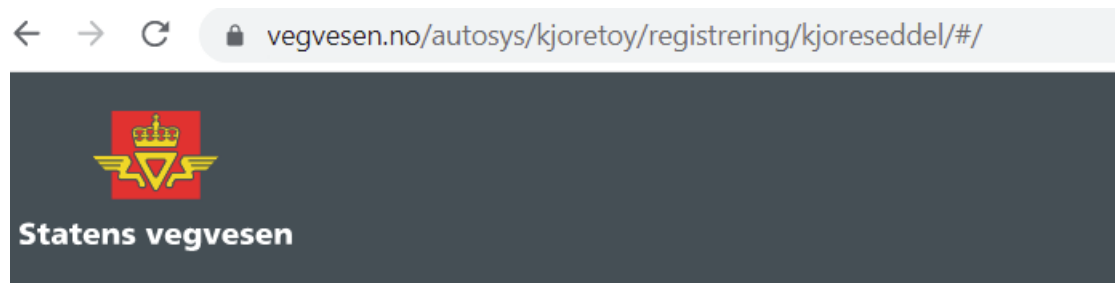
Mer om prosessen ift. Systemleverandører i *kap. 2.5*

### 2.1.15 Kunnskapsutvikling

Reich (2007) er en av mange forskere som viser kunnskapens betydning for å lykkes med IT-prosjekter. Han identifiserte viktig områder innen prosjektarbeid hvor kunnskapsbasert risiko oppstår og må håndteres. Effektiv kunnskapsledelse støtter utvikling og integrering av kunnskap, minimerer kunnskapstap og kunnskapsmangel, og fyller kunnskapshull gjennom hele prosjektet. God samhandling mellom de som sitter med kunnskapen er avgjørende en forutsetning for å gjennomføre effektive prosjekt.

#### Eksempel: Nytt system for kjøretøy og førerkort

Innledningsvis i denne boken nevnte vi et prosjekt som var mislykket – et IT-prosjekt i Statens vegvesen. Det hele startet med at Statens vegvesen i 2004 utlyste en anbudskonkurranse. Det skulle utvikles et nytt førerkort- og motorvognregister. Prosjektet startet i 2005 og ble stoppet i 2007. Da hadde prosjektet kostet 250 millioner kroner. Mangel på kompetanse i Statens vegvesen var grunnen til at prosjektet mislyktes (Schreurs et al. 2011).



Autosys er et nasjonalt motorvogn- og førerkortregister over alle kjøretøy som er registrert i Norge, og alle førerkort som er utstedt i Norge. Data fra Autosys er gjort direkte tilgjengelig for over 800 ulike eksterne interessenter, i tillegg til publikum. Over 7 millioner transaksjoner utføres mot systemet årlig. Det opprinnelige systemet ble bygd på 1980-tallet. I 2004 ble det besluttet å lage et helt nytt system basert på moderne teknologi som skulle gjøre integrasjonen med eksterne brukere enklere. Prosjektet, som ble kalt Au2sys, fikk en knusende dom i 2007 i en rapport ført i penn av konsulentfirmaet Pharos på oppdrag fra Samferdselsdepartementet. Prosjektet ble stoppet i 2007.

Her er forløpet i prosjektet (Schreurs et al. 2011):

- 2004: Statens vegvesen utlyser en åpen anbudskonkurranse. Rammeavtalen de ønsker seg, skal dekke «behovet for konsulent tjenester i forbindelse med informasjonsteknologi». Ciber og Bekk var blant dem som ble med videre.
- 2005: Arbeidet med nye Au2sys settes i gang. Prosjektet er beregnet til 250 millioner kroner over en fireårsperiode.
- 2006: Statens vegvesen deler opp gigantprosjekt og blir dermed beskyldt for å prøve å unngå nye anbudsrunder.
- 2007: En ny rapport, ført i pennen av Pharos, slakter IT-prosjektet i Statens vegvesen. Pharos jobber på oppdrag fra Samferdselsdepartementet og slår fast at prosjektet er helt ute av kontroll: «Vi anser at prosjektet ikke er under tilfredsstillende kontroll, at det ikke har et godt nok planverk, at budsjettet ikke holder mål, og at forankringen i linjeorganisasjonen ikke er god nok». Kontrollen over størrelsen på prosjektet var ikke god nok, man hadde ikke totaloversikt. Prosjektet var ikke høyt nok forankret, og det var ikke gjort godt nok grunnarbeid.
- 2009: Delprosjektet «Autosys Enkeltgodkjenning» blir formelt godkjent like før påske. Dette er den delen av prosjektet som ikke ble stoppet i 2007. Leverandørene her er Ciber og Bekk. Bekk uttrykker at de gjerne vil være med videre.
- 2010: Det gis grønt lys for videre arbeid med nye Autosys. En ny anbudsrunde utlyses. Accenture, EDB Business Partner, IBM og Steria er med.
- 2011: Steria vinner den nye Autosys-kontrakten for kjøretøy- og førerkortregister. Kontraktverdien er beregnet til rundt 500 millioner kroner. Det nye systemet skal utvikles med Java og Oracle Soa Suite. På det meste skal 40 ansatte fra Vegvesenet samarbeide for å utvikle systemet med scrum-metodikk, med kontrakten PS2000 i bunnen. Prosjektet er basert på smidig utvikling. Det starter med en prioritert produkt-kø for leveransene.
- 2014: Autosys skal være ferdig og Statens vegvesen skal da ha brukt 750 millioner kroner på prosjektet. Resultater fra prosjektet skal leveres trinnvis, først med en plattform og deretter fire funksjonelle leveranser.

Mangel på kompetanse ble brukt som begrunnelse for at prosjektet først mislyktes. Styrket IT-kompetanse ble brukt som argument for at prosjektet skulle komme til å lykkes ved andre forsøk. Statens vegvesen hadde da rekruttert ekstra IT-kompetanse for å unngå en ny IT-fiasko (Schreurs et al. 2011, 7):

- «Det er foretatt en kraftig oppbygging av it-avdelingen i Statens vegvesen. Vi ville ikke kjørt dette hvis vi ikke var i stand til det. Vi har spesielt rekruttert it-ressurser i forhold til dette prosjektet», forteller prosjektleder Jon Harald Holm i Statens Vegvesen.

Det pussige var imidlertid her at Jon Harald Holm som prosjektleder selv ikke var ansatt i Statens vegvesen, men innleid konsulent, noe som er betenkelig med hensyn til ansvars- og rapporteringsforhold.

### Kunnskap som ressurs

Kunnskap er en fornybar ressurs som brukes gjentatte ganger, og som akkumuleres i organisasjonen gjennom medarbeidernes erfaring, men kunnskap i bedriften kan ikke eksistere utenfor hodene til de ansatte i bedriften. Informasjonen blir til kunnskap når den behandles i menneskehjernen, og videre kan den blir kommunisert til andre i form av informasjon.

Kunnskap er informasjon kombinert med refleksjon (ettertanke), interpretasjon (tolkning) og kontekst (sammenheng). Informasjonen er basert på data, som er bokstaver og tall og andre symboler uten mening. Når data blir satt inn i en sammenheng som gir mening, blir de til informasjon. Vi sier at opplysninger som man samler inn for å bekjempe hvitstippkriminalitet, kan være både data og informasjon. Disse opplysningene blir til kunnskap etter som mennesker tilegner seg dem. Data kan misforstås. Dersom jeg sier at mitt tall er 61, kunne noen finne på å tro at det er vekten min, at jeg veier 61 kilo. Men det er 61 år som er korrekt oppfatning av tallet. Derfor sier vi at data er bokstaver og tall. Når disse blir satt inn i en sammenheng som gir mening, som for eksempel 61 år, blir data til informasjon. Videre, når informasjon blir koblet til interpretasjon (tolkning), kontekst (sammenheng) og refleksjon (ettertanke), sier vi at informasjonen blir til kunnskap. Når kunnskap akkumuleres over tid gjennom læring, sier vi at kunnskap blir til visdom. Dermed har vi et forståelses-hierarki som går fra data tog informasjon via kunnskap til visdom. Det er en gjengs oppfatning at datamaskiner kan lagre data og informasjon, mens bare mennesker har kunnskap og visdom.



<http://www.annsofiklefvoss.com/kunnskap-viktig-ressurs-verdiskapning-dit-school-of-management-dublin/>

### 2.1.16 Resultatoppfølging

Ethvert prosjekt avsluttes på et eller annet tidspunkt, hvis begrepet prosjekt skal benyttes, men overgang fra prosjekt til drift kan være uklar. Regelmessig prosjektgjennomgang skal sikre at prosjektleder tar stilling til når og om prosjektet er avsluttet, hvor ansvaret for videre etterarbeid bør flyttes over til linjeorganisasjonen eller systemeier. Skal prosjektet bli en innsparing eller økonomisk suksess, må en se framover. Prosessen med å høste gevinstene av løsningen starter som oftest ikke før prosjektet er overlevert, da en ser nye muligheter som kan utnyttes. Deretter kan en rulle ut løsningen i organisasjonen eller i flere enheter eller selskaper. Dessverre stopper prosessen sannsynligvis kort tid etter dette, fordi de som har overtatt eierskapet ikke evner eller prioriterer å gjennomføre tiltakene som må på plass.

Evaluering av avsluttet prosjekt og videreføring av implementering er derfor viktig.

Noen prosjekter avsluttes før de når målsetninger og kan kalles fiasko. Årsaken til dette kan være nyttig lærdom som kan ha følgende moment:

- *liten sannsynlighet for målsetninger nås, slik at det gis opp*
- *tekniske utfordringer som ikke lag seg løse med valgte verktøy, leverandører eller egen kompetanse*
- *høyere prioritet for andre prosjektet som konkurrerer om de samme ressursene*
- *lav lønnsomhet på investeringene, manuelle operasjoner dekke behovene bedre*
- *for dyrt å utvikle som selvstendig prosjekt eller som kundetilpasning*
- *markedspotensial for løsningen viser seg å være mindre enn antatt, endre forutsetning*
- *for lang tid å kommersielle resultat eller høsting av gevinster kan oppnås*
- *negative effekter på andre prosjekter eller produkter*
- *patentproblemer*
- *manglede finansiering på kostnadsoverskridelser*

Suksesskriterier for prosjektet blir gjerne målt i:

- *kostnader – at prosjektet fullføres innenfor avsatt budsjett*
- *tid – at prosjektet blir avsluttet innenfor definert sluttdato*
- *kvalitet – at leveransen har ønskede egenskaper og oppfyller de behovene som er gitt*



Å henvise kun til dette synes en i senere tid blir for snevert. Prosjekt som operahuset i Sydney hadde store kostnadsoverskridelser og mange års forsinkelse, blir likevel ansett som en verdenssuksess i ettertid. På samme måte vil vellykkede integrasjoner kunne gi så store besparelser, nye muligheter, og økt kompetanse, og samtidig være enkel å implementere i organisasjonen at det er alt en husker i etterkant. Dermed kan kvaliteten på den leverte løsningen bli så høy at tid og kostnadsoverskridelser overskygges fullstendig, slik at prosjektet blir husket som en suksess.