



Prosjektstyring

Innhold:

- Prosessmodeller og prosjekter
- Prosjektplanlegging, inkl. tidsplanlegging
- Estimering og risikostyring



Forelesningene er bla basert på ...

Sommerville-boka

Steve McConnell: Software Project Survival Guide. ISBN 1-57231-621-7.

Hans Mikkelsen og Jens O. Riis. Grundbog i Prosjektledelse. ISBN 87-89477-10-3.

Svein Arne Jesssen. Prosjektadministrative metoder. ISBN 82-00-22551-8.

Harald Westhagen. Prosjektarbeid. ISBN 82-00-03341-4.

Viktig: Forelesningene vil ikke følge Sommerville-læreboka i rekkefølge og framstilling. Det er heller ikke tid til å dekke alle temaene som tas opp i pensum => Mye selvstudium.



Målsetninger

Etter å ha deltatt på forelesningene om prosjektstyring/utviklingsmodeller bør dere:

- Kjenne til og forstå viktige begreper innen systemutviklingsprosesser og prosjekter
- Ha et godt utgangspunkt for å kunne analysere fordeler og ulemper med ulike organiseringer av systemutviklingsarbeid
- Kunne kombinere kunnskap om systemutviklingsprosesser med kunnskap om prosjektstyring
- Kjenne til et utvalg av teknikker for å planlegge, estimere og styre systemutviklingsprosjekter

=> Denne kunnskapen skal **BRUKES** og **EVALUERES** i den obligatoriske prosjektoppgaven for å få **KUNNSKAP** over til **FERDIGHETER**.



Prosjektstyring



Hva er et prosjekt?

- lat. "projectus" - noe som er kastet frem
- Et prosjekt er en arbeidsoppgave med følgende egenskaper:
 - Engangsoppgave
 - Definert mål og kan skilles ut som eget styringsobjekt
 - Skal gjennomføres innenfor bestemte tids- og kostnadsrammer
 - Ofte tverrfaglig og krever koordinert innsats fra flere personer eller organisasjonsenheter.



Hva er en prosessmodell?

- "Kjært barn har mange navn":
 - livssyklusmodell
 - systemutviklingsmodell
 - systemutviklingsmetode
- Prosess (Sommerville): "A set of activities and associated results which lead to the production of software products."
- Prosessmodell (Sommerville): "Abstract representation of a software process" (from a particular viewpoint).
- Er det noen forskjell på en modell og en metode?
- Prosessmodeller kan i forskjellig grad være deskriptive eller normative." Hva menes?



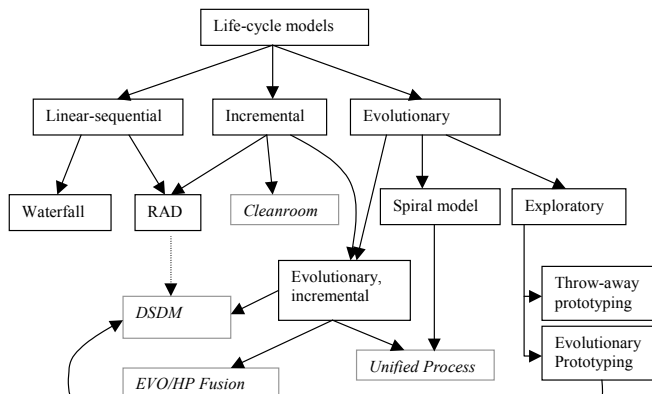
Prosessmodell - I

Typen prosessmodeller for systemutvikling (Sommerville):

- vannfall
- evolusjonær
- inkrementell
- spiral
- gjenbruksbasert



Eksempel på alternativ klassifisering





Prosessmodell - II

- Faser som inngår i (nesten) alle prosessmodeller:
 - Forstudium/"Feasability study"/"Pre-study" (Hvilke muligheter har vi?)
 - Kravspesifikasjon (Hva skal produktet gjøre?)
 - Design (Hvordan skal produktet bygges?)
 - Programmering ("Bygging")
 - V&V (Validering og verifikasjon) / Testing
 - » Har vi bygd riktig system/produkt? (validering)
 - » Har vi bygd systemet/produktet riktig? (verifikasjon)
 - Videreutvikling/Vedlikehold



Er prosessmodell viktig? Eksempel 1 - Microsoft

- Utgangspunkt: Hacker-kultur. Prosessmodell var i realiteten "Code-and-fix" (mer enn 50% testere + mye feil + overskridelser i tid og kostnad).
- Enorme leveranseproblemer som følge av økende kompleksitet i produktene.
- Problem: Hvordan få en høyere kvalitet og forutsigbarhet i leveransene samtidig som man tok hensyn til "Microsoft"-kulturen og ikke mistet dyktige programmerere? (Man ville ikke bli så strukturerte som IBM-programmererne)
- "Løsning": Endret prosessmodell (ikke nye verktøy eller språk) til "synch-and-stabilize" (inkrementell utviklingsmodell med "daglig" (!!)) integrasjon og test av kode i bygg-fasen).
- Resultat: En bedre modell for utvikling av svært komplekse systemer med programmerere som kunne fortsette å "hacke", men nå med mindre sannsynlighet for at ny funksjonalitet ikke lot seg integrere uten masse feil. (men, fortsatt mer enn 50% testere!)



Er prosessmodell viktig? Eksempel 2 - TRESS-90

- Trykdeetatens store EDB-satsing på 90-tallet.
- SINTEFs eksterne prosjekt revisjon viste bla at:
 - Bygg-fasen ble startet uten å være i nærheten av en stabil design.
 - Reell brukermedvirkning på en del viktige områder, som brukergrensesnitt, kom først i bygg-fasen (men likevel vannfallsmodell!)
 - Interne prosjektavhengigheter dårlig håndtert.
 - Ingen "kritisk sti" => alle aktiviteter ble oppfattet som like (lite) viktige for å holde leveransetidspunktet
 - Kvalitetssikring uten myndighet.
 - Personer ble tatt ut av prosjektet på avtalt dato, uavhengig om oppgaven var ferdigstilt eller ikke.
- Reell gjennomføring var langt fra sekvensiell (vannfalls), men kontrollfunksjoner, avtaler og innkjøp syntes å gå ut fra at kartet (valgt prosessmodell) var mer riktig enn terrenget (reell prosess). Konsekvensen var at prosjektet kom ut av styring.



Prosessmodell vs prosjektarbeid

- Sammenhenger:
 - Faser i prosjektmodellen blir til "overordnede aktiviteter og milepæler" i prosjektplan (men prosjektplan vil typisk inneholde flere og mer detaljerte aktiviteter)
 - Relasjon mellom faser i prosjektmodellen blir til Gantt-diagram eller lignende i prosjektplan.
 - Innhold i faser i prosjektmodellen blir til leveranser, dokumentasjon, verifikasjon og validering, risikostyringsaktiviteter m.m. i prosjektplan
- Forenklet sagt: Prosessmodellen vil ofte være organisasjonens valg av felles arbeidsmåte for å sikre gjenbruk av erfaring og forutsigbare leveranser. Prosjektplanen er prosjektleders virkemiddel for å sikre leveransene under rammebetingelsene som settes av prosessmodellen. Konflikter oppstår dersom disse to målene ikke samsvarer (ref. "skunk projects" og "byråkrati").



Prosjektplanlegging



Prosjektplanen

- Mulig prosess for utarbeidelse av prosjektplan (iterativ prosess):
 - Finn person(er) som kan utarbeide prosjektplan (bør inkludere den kommende prosjektleder - eierskap til plan er viktig)
 - Forstå så mye som mulig av kravene til leveransene (kravspesifikasjon og implisitte krav), samt andre viktige rammer for prosjektet (f eks leveransefrist og kostnadsrammer). Vurder om rammevilkår er tilfredstillende.
 - Velg prosessmodell ut fra rammevilkår (marked, risiko for endring av krav, kundemodenhet, kompetanse i organisasjonen,).
 - Definer milepæler, leveranser og aktiviteter, samt avhengigheter mellom disse.
 - Estimer tids og timeforbruk per aktivitet
 - Analyser om tilgjengelig ressurser finnes
 - Vurder risiko og forhandle om nødvendig leveranseinnhold eller andre rammevilkår.
 - Dokumenter resultatene av prosessen i et dokument som skal tjene som styringsdokument og kommunikasjonsinstrument.



Tidsplanlegging i prosjekter

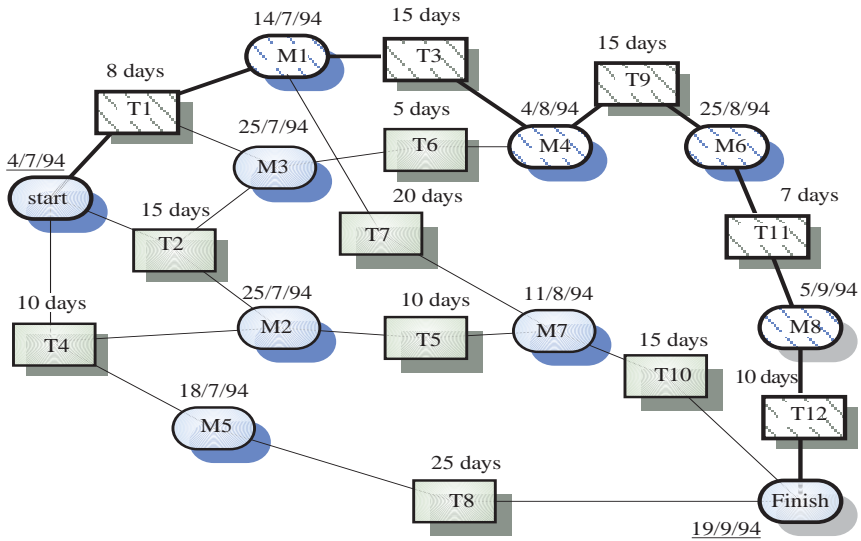


Arbeidsoppgaver – varighet og avhengigheter

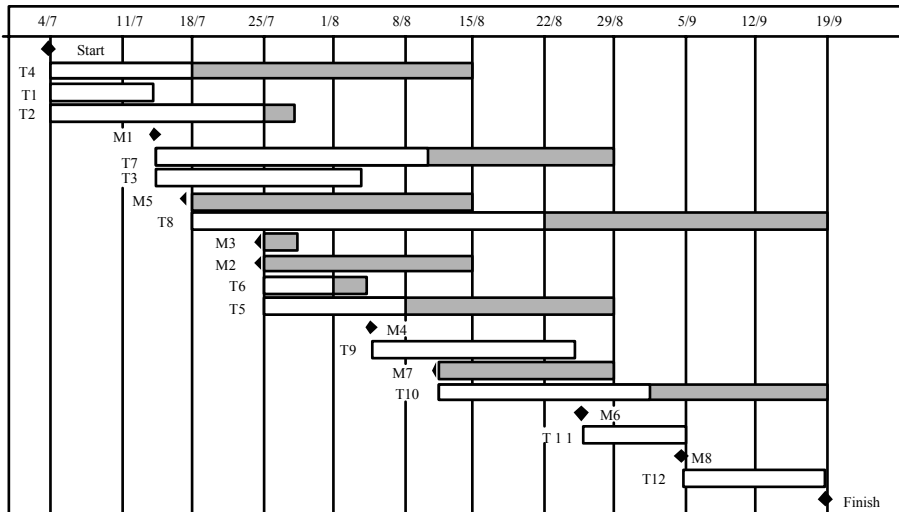
Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1
T4	10	
T5	10	T2, T4
T6	5	T1, T2
T7	20	T1
T8	25	T4
T9	15	T3, T6
T10	15	T5, T7
T11	7	T9
T12	10	T11



Aktivitetsnettverk

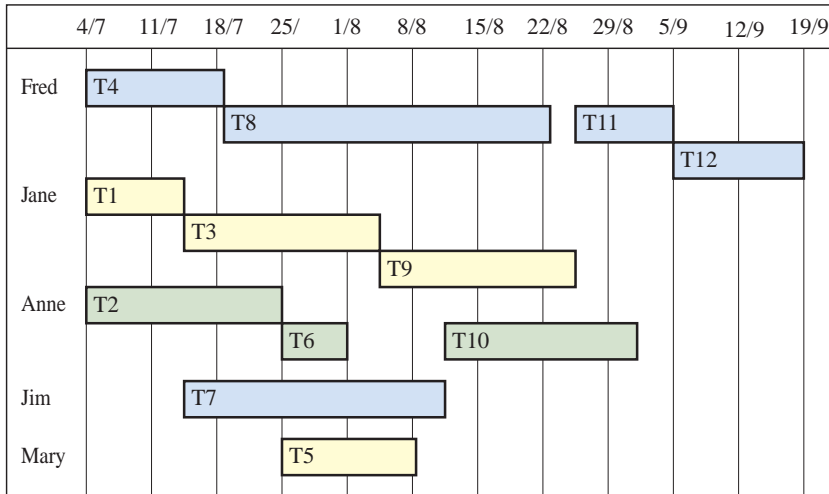


Gantt-diagram





Person-allokering



Støtteverktøy

- MS Project (for mindre prosjekter)
- Project Workbench PMW (for store prosjekter)
- Se oversikt over noen andre verktøy:
<http://www.wior.uni-karlsruhe.de/bibliothek/Project/>

INF3120-oppgaven:

- I INF3120-prosjektoppgaven vil det ikke være nødvendig å bruke noe verktøy for prosjektplanleggingen. Det er imidlertid ingenting i veien for å bruke ett eller flere av disse verktøyene! Sørg imidlertid for å ikke bruke mye tid på prosjektplanleggingsverktøy. Det viktige er å skjønne prinsippene.
 - Erfaringsmessig er at de beste prosjektlederne er minst opptatt av å ha avanserte støtteverktøy for prosjektplanlegging/styring og forsøker hele tiden å holde det så enkelt som mulig.



Estimering og risikostyring



Innledning

Målsetning:

- Oversikt over typer metoder og teknikker
- Forstå bruken av metodene
- Bedre gjennomføring av prosjektoppgaven

Innhold:

- Hva er et estimat?
- Hva er risiko?
- Modeller og metoder
- Estimering og risikostyring i prosjektoppgaven.



"Oppvarming"

- Er det usannsynlig at en svært usannsynlig uheldig episode som vil forsinke et IT-prosjekt vil skje i løpet av en lengre prosjektperiode?
 - NB: Dette er kanskje den største kilden til overoptimisme i systemutviklingsprosjekter.
- Hva er andre mulige årsaker til at man som oftest er for optimistisk når prosjekter skal planlegges?
- Hva er viktige hindre for å lære av erfaring?



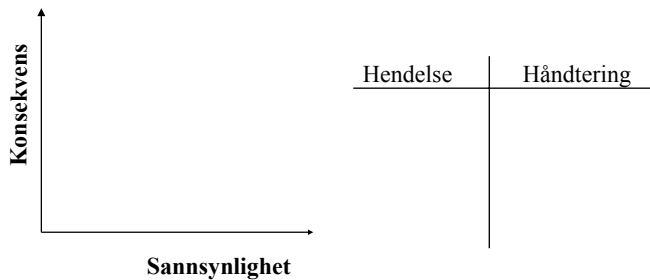
Plan = Estimat + Risikobuffer

- Estimert ressursforbruk: 50% sannsynlighet for at virkelig ressursforbruk ikke overstiger det estimerte (gitt at forutsetningene for estimatet er korrekte). Et estimat bør brukes sammen med angivelse av sannsynlige nedre og øvre grenser for ressursforbruk, risiko forbundet med estimatet og antagelser som ligger til grunn. Estimaten brukes som inndata til planlagt ressursforbruk og ressursforbruk til grunn for fastprisavtaler.
- Planlagt ressursforbruk: Det ressursforbruket som prosjektleder bruker som utgangspunkt for planleggingen. I de fleste tilfellene bør dette være høyere enn det estimerte ressursforbruket, dvs det bør være mer enn 50% sannsynlighet for å holde planen. Plan = Estimat + Risikobuffer.



Risiko

- Risiko beregnes ut fra sannsynlighet og konsekvens til en hendelse.



Øvelser

Øvelse 1:

I relasjon til prosjektene dere skal starte, hvor omtrent vil dere plassere og håndtere hendelser av typene:

- Samarbeidsproblemer
- Tekniske problemer

Er det andre hendelser som er viktige mhp prosjektets risiko?

Øvelse 2:

Hvilke forskjeller bør det være mhp håndtering av følgende risikotyper:

- "Normalvariasjon"
- Risikofaktorer vi kjenner til
- Risikofaktorer vi ikke kjenner til (uventede hendelser)
- Kaos og total redefinering

(Hvordan påvirker innflytelsen vi har på utfallet håndteringen?)



Kort innføring i risikostyrt estimering (PERT-estimering, utvidet)

- Denne metoden skal brukes i prosjektoppgaven!
- Viktige spørsmål:
 - Hva bestemmer om oppdelingen i aktiviteter er god?
 - Hva kan estimeres når? og i hvilken rekkefølge?
 - Hvorfor bør vi starte med "mest pessimistisk estimat" ?
 - Bør det legges til ekstra risikobuffer for svært usannsynlige hendelser? (og i så tilfelle, hvor stort?)
 - Når bør det reestimeres?
 - Hvordan håndtere at både kravene og tidsfristen er gitt? (estimering når man har en inkrementell utviklingsmodell)



PERT-estimering

- $\text{Estimat} = (\text{MP} + 4 \cdot \text{MS} + \text{MO}) / 6$
 - Forutsetninger er bla "beta-fordeling" av mulig arbeidsmengde.
 - MP = Mest pessimistisk, MS = Mest sannsynlig, MO = Mest optimistisk
 - Med "mest pessimistisk" hhv "mest optimistisk" menes noe i retning av en verdi det er usannsynlig (f eks at det kun vil skje i 1 av 20 tilfelle) at vil bli overskredet.
- Hvordan fastlegge MP og MO?
 - Ekspertvurdering
 - "Empiriske fordelinger" (anbefales dersom mulig)



Noen svakheter ved ekspertestimer

- **Svakhet 1:** Vi undervurderer effekten av kombinerte OG-sannsynligheter. F eks.: Selv om sannsynligheten for at aktivitetene A, B, C og D skal skje uten større problemer er høy, hhv 80%, 90%, 70% og 80%, er sannsynlighet for at A OG B OG C ikke skal ha større problemer relativt lav ($0.8 * 0.9 * 0.7 * 0.8 = 0.4$), dvs 40%.
- **Svakhet 2:** Vår initielle vurdering ("anker") har typiske en urimelig stor innflytelse på senere vurderinger. Vi lar oss lett påvirke av villedende informasjon, selv om vi vet at den er villedende.
- **Svakhet 3:** Vi klarer ikke å ta nok i mhp MIN og MAX-intervaller.
- **Viktig:** Studier viser at disse svakhetene ikke blir særlig mindre med lang erfaring.



Svært usannsynlige hendelser ...

- For prosjekter som varer en stund er det sannsynlig at det vil skje en eller flere usannsynlige alvorlige hendelser. Det er ofte ikke hensiktsmessig å ta med slike betraktninger på detaljert nivå, f eks "hvor sannsynlig det er at oppdragsgiver går konkurs underveis".
- Estimeringsmodeller håndterer dette problemet ikke særlig godt.
 - Legg på litt "ekstra" (f eks 10%)
- Forsikringselskaper løser et lignende problem på denne måten:
 - En del av forsikringspremien for skadeforsikring er ikke "historiebasert", men skal ta høyde for katastrofer (svært usannsynlige hendelser) som de vet vil skje av og til.



Estimeringsmetoder - oversikt

- ekspertestimater (“educated guess”)
 - uformell
 - delphi-metode
- mer strukturerte metoder
 - top-down
 - bottom-up (som vist i tidligere eksempel)
 - inside-out (f eks starter med bottom-up av programmeringsfasen, deretter top-down for resten av aktivitetene)
 - regresjonsanalyse (ofte i form av “debate and trial”, dvs pseudo-regresjon)
 - analogi (f eks bruk av mønstergjenkjennings-algoritmer)
- De aller fleste estimeringer er enten rene ekspertestimater eller kombinerer tilnærmingene ovenfor. Dvs, det er nesten ingen som “tror på” å kun bruke en estimeringsmodell som ikke krever ekspertkunnskap.
- Eksempler på estimeringsmodeller som bruker metodene ovenfor:
 - COCOMO I og II (formel + linje kode eller funksjonspoeng-basert, top-down)
 - Angel (finder lignende prosjekt, justerer for ulikheter, top-down)



Ekspertestimater

- Ofte de mest nøyaktige, men muliggjør ikke (i samme grad som modeller):
 - akkumulering av ekspertuavhengig kunnskap
 - læring fra historien
 - kvantitative analyser
- Ofte anbefalt at ekspertestimater brukes på “aktivitets-nivå”, mens modell-estimater brukes for å korrigere/validere aktivitets-estimatene og å gi et perspektiv på total-estimatene.
- Undersøkelser viser at ekspertestimatene stort sett er like nøyaktige som modell-estimater, men at ekspertestimater har en større tendens til alt for optimistiske estimater.
- Delphi-metode går ut på at flere eksperter “konvergerer” i estimater:
 - 1. runde gir de uavhengig av hverandre estimater
 - 2. runde (hvis ikke konsesus) ser de på hverandres estimater (og begrunnelser) og endrer eventuelt sitt eget i lys av denne informasjonen
 - osv. helt til konsensus er nådd.



“Formelle” estimeringsmetoder (1)

Typisk metode for top-down estimering av arbeidsmengde vha estimeringsmodell:

1. Estimer/mål størrelsen på det som skal utvikles (f eks i “linjer kode”, “object points”, “feature points” eller i “funksjonspoeng”).
2. Bestem andre faktorer som har innvirkning på produktiviteten (f eks kompleksitet, nye verktøy, ustabile krav,)
3. Finn historisk produktivitet (basert på samme størrelse-mål som under 1) og juster for “unormale tilstander” (høy kompleksitet osv, dvs alle relevante faktorer foruten størrelse).
4. Arbeidsmengde [f eks i månedsverk] = Størrelse / Produktivitet



“Formelle” estimeringsmetoder (2)

Top-down: Start med et estimat av hele systemutviklingen, bryt ned i aktivitetsestimater/faser.

Bottom-up: Start med estimater av aktiviteter/faser, summer for å få estimatet for hele systemutviklingen. PERT-basert estimering er stor sett bottom up.

Inside-out: Start med bottom-up estimering av ”basisaktiviteter” (f eks programmeringsaktiviteter). Bestem arbeidsmengde på de andre aktivitetene i relasjon til dette estimatet, f eks at testaktiviteter er 50% av arbeidsmengden til programmering.

Regresjon: Gitt historiske data, trekkes en “linje” nærmest mulig til de historiske datapunktene.

Analogi: Bruk av algoritmer for å finne en mengde med tidligere systemutviklingsprosjekter som ligner mest på det aktuelle.