



Situasjonen i IT-bransjen i dag

- I gjennomsnitt ca. 30% underestimering av kostnader
- Ingen vesentlig forbedring over tid
- Konsekvenser:
 - Gjennomføringsproblemer
 - Misfornøyde kunder
 - Dårlig lønnsomhet eller tap for leverandør



Hvorfor har vi disse problemene?

- Iboende problemer:
 - Komplekse produkter (*stadig nye uprøvde muligheter, lite mulighet til å akkumulere erfaringer, kompleksiteten øker raskere enn størrelsen på prosjektet*)
 - Komplekse organisasjonsendringer ofte en del av leveransen (*mye følelser og posisjonering involvert*)
 - Komplekse prosjekter (*prosjekter er komplekse "ad hoc" - organisasjoner som dannes og skal yte fra første dag*)
 - Uklare krav (*skal prosjektet vente til alle krav er klare, kommer det aldri i gang*)
 - Menneskelige svakheter (*"bounded rationality", tabber, "små" feil med store konsekvenser, misforståelser, iboende optimisme*)



Hvorfor har vi disse problemene?

● Oppdragsgiverproblemer:

- Uklare krav
- Lite kunnskap / feilaktige forventninger til IT / ”politikk”
- Kommunikasjonsproblemer med leverandør (mangler felles språk)
- Avsetter for lite tid/ressurser til involvering/oppfølging
- Mangel på forankring i ledelse / forretningsstrategi
- Apati overfor IT-leverandører (det er slik de er-holdningen)?
- Anbudsprosesser som fører til valg av overoptimistiske leverandører (”the Winner’s curse”)



Winner’s Curse: Desto mer fokus på lav pris, desto høyere forventet overskridelse (ROSING-formelen)



$$RO = S \cdot IN + G$$

RO = Relative Overrun

S = Selection bias (= focus on low price)

IN = INaccuracy

G = General estimation over-optimism



Eksempel:

- Kunde velger et tilbud som er 40% lavere enn gjennomsnittstilbudet ($S = 0.4$)
- Korrelasjon mellom estimert og faktisk kostnad er 70% ($IN = 100\% - 70\% = 0.3$)
- Leverandørene er i gjennomsnitt 20% over-optimistiske i sine estimater ($G = 0.2$)
- Forventet kostnadsoverskridelse for valgt leverandør er 32% ($0.4 \cdot 0.3 + 0.2$)
- Sterkere prisfokus fører til høyere S og høyere forventet overskridelse!



Winner's curse formelen ...

$$(6) \text{rel} = \frac{\text{act} - \text{est}}{\mu} = \frac{\mu \left(1 - \rho_{\text{est}, \text{act}} \frac{\sigma_{\text{act}}}{\sigma_{\text{est}}} \right) (1-w)}{\mu} = \left(1 - \rho_{\text{est}, \text{act}} \frac{\sigma_{\text{act}}}{\sigma_{\text{est}}} \right) (1-w).$$

Mer om dette på: <http://simula.no/publications/Simula.simula.1478>



Hvorfor har vi disse problemene?

● Leverandørproblemer:

- Lite erfaringer mhp planlegging og gjennomføring av nye typer (f eks svært store) IT-prosjekter
 - » De virkelig store prosjektene får man stort sett oppleve kun 1-2 ganger i sin karriere.
 - » Fare for at man overfører erfaringer fra mindre og mellomstore prosjekter til store prosjekter, mao de vesentlige forskjellene (bla mhp produktivitet og risiko-eksplosjon) tar man ikke nok hensyn til.
 - » Undervurdering av viktigheten ved kommunikasjon med oppdragsgiver. Dessuten, kommunikasjon med brukere er lite lystbetont arbeid for mange "hackere".
- Kundens ansvar for prosjektfiaskoer (og suksesser) er ofte sterkt undervurdert



Hvorfor har vi disse problemene?

● Leverandørproblemer:

- Mangelfull læring av tidligere prosjekter
 - » *Forskningsresultater viser at vi er svært dårlige i å lære av tidligere erfaringer.*
 - » *"Feedback" er svært mangelfull. F eks, det er ingen felles forståelse av hva et estimat er.*
- For lite fokus på risiko
 - » *Sterk undervurdering av størrelse på "det uventede"*
- Uheldig valg av systemutviklingsprosess
 - » *Uklare krav, mange aktører, høy risiko, sammen med en rendyrket "fossefalls-modell" er en typiske feilen som gjøres.*



Ti faktorer som signaliserer fare for underestimering

1. Høy usikkerhet
2. Lite relevant erfaring
3. Sterkt ønske om å få lov til å starte opp prosjektet hos de som estimerer
4. Vanskelig å evaluere estimatene (f eks ved at antagelsene for estimatene er uklare eller at hva som er ment med estimatet ikke er klart)
5. Lang varighet på prosjektet
6. Prosjekt mye større enn tidligere prosjekter
7. Prosjektleveranser har liten grad av fleksibilitet (dette øker risikoen for at man overvurderer hvor mye man kan "justere seg inn")
8. Manglende skille mellom prosesser for estimat, plan og tilbud
9. Kundeforventning og/eller budsjett er kjent av de som estimerer
10. Estimerer utarbeidet som del av planlegging



Estimeringsprosessen



Forberedelser

- **1. Forstå estimeringsproblemet**
 - Identifiser mål og krav til nøyaktighet
 - Identifiser interessenter og politiske posisjoner
 - Spesifiser forutsetninger
 - Bestem nedbryting av problemet
- **2. Enighet om beslutninger og forutsetninger**
 - Identifiser relevante beslutninger og forutsetninger som kan påvirke
 - Avgjør om det er meningsfullt å estimere på nåværende tidspunkt
 - Avklar fleksibilitet og prosjektprioritet



Forberedelser

- **3. Innhent relevant informasjon**
 - Identifiser selskapsspesifikke kostnadsdrivere
 - Pass på at kildene er uhildet
 - Innhent informasjon fra flere kilder
 - Unngå irrelevant informasjon
- **4. Velg estimeringsprosess**
 - Baser prosessen på tilgjengelig informasjon
 - Benytt organisasjon og personspesifikk informasjon



Estimeringsfasen

- **5. Estimer mest sannsynlig arbeidsmengde**
 - Struktur estimeringsprosessen
 - Separer mest sannsynlig arbeidsmengde fra tilbud, plan etc.
 - Beskriv forutsetninger
 - Beskriv underliggende informasjon for etterprøvbarehet
- **6. Anslå usikkerhet**
 - Konfidensintervall



Estimeringsfasen

- **7. Gjennomgang av estimeringsprosessen og estimat**

- Benytt uavhengige eksperter til gjennomgang
- Sørg for at gjennomgangen kan føre til forandringer
- Benytt en sjekkliste



Anvendelsesfasen

- **8. Benytt estimatene i tilbudsskriving**

- Ta utgangspunkt i mest sannsynlig arbeidsmengde og estimat-usikkerheten

- **9. Benytt estimatene i planleggingen**

- Bestem buffer for uforutsette hendelser
- Planlegg aktiviteter som reduserer usikkerhet, som utvikling av delfunksjonalitet
- Planlegg re-estimering



Anvendelsesfasen

- **10. Kommuniser estimer, tilbud, plan og usikkerhet**
 - En god estimeringsprosess er et godt salgsargument!
 - Tilpass informasjon etter modenhet
 - Spesifiser risiko, og hvordan denne skal håndteres
 - Tilgjengeliggjør oversiktlige estimer og antakelser
 - Erkjenn og forhold dere til mottakers mål, uten å redusere realismen
- **11. Kontroller kostnadene**
 - Monitorer utviklingen og re-estimer
 - Sørg for å holde alle deltakere informert
 - Favoriser enkelhet



Læringsfasen

- **12. Lær av erfaringer**
 - Arranger erfaringsgjennomgang
 - Forstå underliggende årsaker for eventuelle avvik
 - Oppdater sjekklisten, erfaringsdatabasen, WBS etc. på bakgrunn av gjennomgangen
 - Ikke overgeneraliser



Typer usikkerhet i estimatene og hvordan disse håndteres

- Normalvariasjon i produktivitet
 - Angis f eks som minimum-maksimum intervaller per aktivitet
- Risiko som følge av kjente risikofaktorer
 - Angis f eks som sannsynlighet x utfall, samt innvirkning på totalt kostnadsforbruk
- Risiko som følge av uventede hendelser (“forvent det uventede”)
 - Angis som “risikobuffer” basert på andel kostnader til håndtering av uventede hendelser
- Kaos (f eks total endring i prosjektets mandat)
 - Krisehåndteringsrutiner



Eksempel på egnet prosess for usikkerhetsvurdering

1. Estimer mest sannsynlig arbeidsmengde
2. Finn fram til tidligere prosjekter med lignende “estimeringskompleksitet” (de trenger ikke være veldig like, det er mer viktig at det er minst 10-20 prosjekter i grunnlaget)
3. Lag en fordeling av estimeringsavvik for disse (se neste slide)
4. Bruk denne til å finne hvor sannsynlig, f eks, 50% overskridelse er
5. La risikobuffer og budsjetter gjenspeile denne usikkerheten



Eksempel ...

Table 2. Distribution of Estimation Error of Similar Projects

Teams (Group B only)										
Estimation Error Category	11	12	13	14	15	16	17	18	19	Mean value
>100% overrun	45	18	10	10	10	5	10	0	18	14
50-100% overrun	20	40	35	20	10	5	20	5	25	20
25-49% overrun	15	22	25	30	30	35	40	20	30	27
10-24% overrun	10	15	25	20	30	45	20	40	15	24
+/- 10% of error	7	4	0	5	10	10	10	20	12	10
10-25% too high estimates	3	1	0	10	5	0	0	10	0	3
24-50% too high estimates	0	0	0	0	5	0	0	5	0	1
>50% too high estimates	0	0	0	0	0	0	0	0	0	0



Hva med estimeringsmodellene?

- COCOMO, SLIM, PRICE-S, Estimacs, ...
 - Eks: Arbeidsmengde = $2,4 * \text{Størrelse}^{1,05} * \text{Kompleksitet}$
- Størrelsesmål: MkII Function Point, Linjer kode, Use Case Points, ...

Viktig prinsipp: Bruk enkle metoder dersom det ikke er påvist at de mer kompliserte modeller er bedre – og det er det ikke for de som er nevnt ovenfor!

De få studiene som er gjennomført viser at enkle modeller og ekspertestimater som oftest er minst like gode som de mer avanserte modellene. En grunn til dette er at enkle modeller er mer “robuste”, dvs de gjør ikke så mange antagelser mhp fordelinger og sammenhenger.

Dessuten, enkle modeller muliggjør at brukeren skjønner antagelser og utregninger, kan forholde seg til estimatene. Det er en grunn til at svært få bruker estimeringsmodeller.

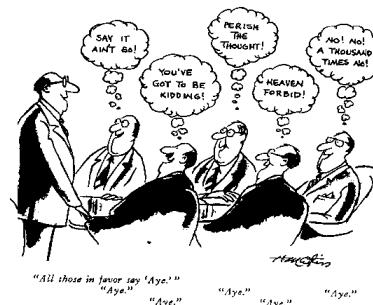


Planning Poker (eksempel på metode for gruppeestimering - ofte brukt i smidige prosjekter)



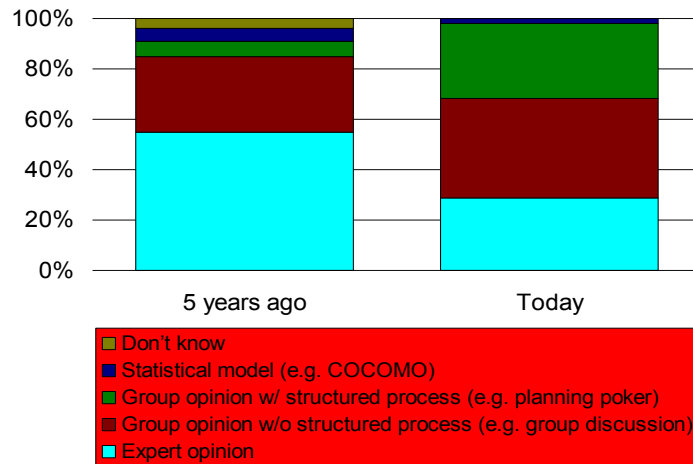
Forskning på gruppe-estimering

- Få studier innen Software Engineering
- ...men mange relevante studier innen andre forskningsfelt (psykologi, business forecasting, etc)
- Fordeler:
 - Kombinering av estimerer forbedrer estimeringen (spesielt når de som estimerer har forskjellig bakgrunn)
 - Struktur kan forbedre estimeringen (for eksempel: redusere påvirkningen fra irrelevant informasjon)
 - "Flere hoder husker mer enn ett"
- Ulemper :
 - Ressurskrevende (dyr) sammenlignet med individuell estimering
 - "Group think" kan forekomme (for eksempel: at alle er enige med sjefen)
 - "Group polarization" kan forekomme (for eksempel: gruppen er mer optimistisk enn gjennomsnittet av individene)





Gruppe-estimering vinner frem i norsk IT-industri (undersøkelse på JavaZone 2007)



© Institutt for informatikk Magne Jørgensen

29



Planning Poker

1. Kunden forklarer “user story”
2. Teamet diskuterer hvilken jobb som må gjøres
3. Alle velger et kort som representerer estimatet
4. Alle viser estimatet sitt samtidig
5. De med lavest og høyest estimat begrunner
6. Teamet diskuterer estimatene
7. Gjenta fra steg 3. frem til estimatene konvergerer
8. Teamet blir enige om et estimat

Kort: Spillkort, f eks med tallene 1, 2, 3, 5, 8, 13, 21

© Institutt for informatikk Magne Jørgensen

30



Når kan vi bruke Planning Poker?

- Release-planlegging
 - kunden velger funksjonalitet for neste release
 - estimatene er basis for å prioritere kravene og prosjektbemanningen
 - Planning Poker kommer raskt opp med realistiske estimater og avslører uklare krav
- Iterasjonsplanlegging og design
 - Bryter ned kravene i konkrete oppgaver og tildeler ansvar for oppgavene
 - Estimering med Planning Poker avslører uklare krav
 - Planning Poker kan fasilitere design-diskusjoner

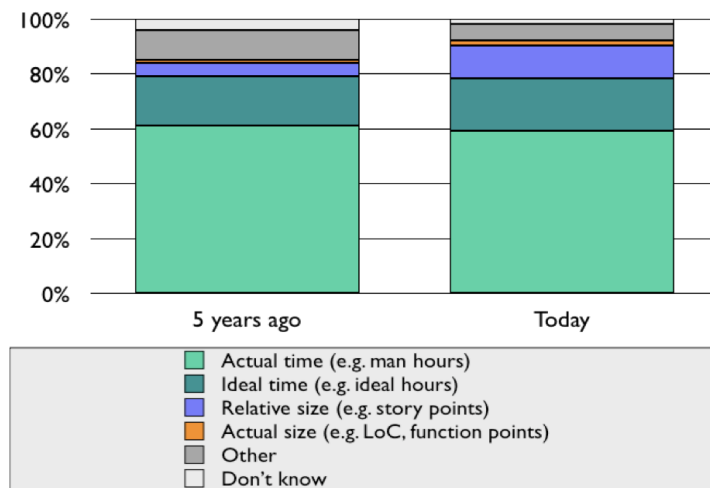


Hva bør vi estimere?

- Alternative enheter for estimeringen
 - “Story points” (relativ estimering)
 - Ideelle dager
 - Timeverk
- Utled hvor mye man rekker i neste release, iterasjon, ..
 - Mål prosjekthastigheten (produktivitet, “velocity”) på tidligere releaser. Prosjekthastighet = summen av de “story points” som ble levert i tidligere iterasjoner
 - Anta at man rekker like mye som i tidlere release, iterasjon, ... (IKKE anta at du har blitt mye mer produktiv)



“Hva estimerer du?” (JavaZone 2007)



© Institutt for informatikk Magne Jørgensen

33



Effektivisering av estimeringen

- Kategorier arbeidsmengde øker effektiviteten i estimeringen
- Bør gjenspeile nøyaktigheten i estimatene
 - 1256,5 timeverk gir f eks et galt bilde av nøyaktighet i de fleste situasjoner.
- Fibonacci-sekvensen brukes ofte: 1, 2, 3, 5, 8, splitt

© Institutt for informatikk Magne Jørgensen

34



Bør vi forsøke å bli (helt) enige eller skal vi bruke gjennomsnittet?

- **Begrunn estimatene etter den første runden med Planning Poker**
 - Avdekker hva man har tatt hensyn til i estimeringen
 - Viktig for å avdekke mest mulig detaljer
- **Anbefaling**
 - Gjør alltid minst to runder med Planning Poker
 - Fortsett så lenge forskjellene i estimerer er store
 - Bruk gjennomsnittet (eventuelt flertallet) når forskjellene er små



Forhold man bør ta hensyn til

- **Ikke bruke for mye tid / grave seg ned i for mange detaljer**
 - Ikke diskutert altfor lenge før den første runden med poker
 - Etter en stund vil diskusjonene gi mindre verdi
 - Bruk en stoppeklokke dersom lange diskusjoner er et problem
- **Fange opp de forskjellige synspunktene**
 - Mange spørsmål vil komme opp i diskusjonene
 - Viktig å ha representanter med forskjellige synspunkt tilstede



Hvorfor virker Planning Poker?

- **Samtidig visning av estimater kan redusere noen feilkilder**
 - Det første estimatet vil normalt danne et anker
 - Noen i teamet har mer innflytelse enn andre
- **Flere spørsmål blir stilt, og mer informasjon blir delt**
- **Estimatene reflekterer teamets gjennomsnittlige evne til å løse oppgaven**
 - Ekspert-estimer har en tendens til å basere seg på ekspertens evner
 - Vi vet ikke nødvendigvis hvem som vil ende opp med å gjøre oppgaven



Råd for å bedre estimeringsnøyaktighet

1. Skill tydelig mellom prosesser for estimering, planlegging og prising.
2. Unngå irrelevant informasjon som leder til for lave ”ankere” (f eks kundeforventninger)
3. Bruk historiske data.
4. Dersom historiske data ikke finnes, forsøk å finne eksperter med relevant bakgrunn (også utenfor organisasjonen) og historiske data.
5. Ikke stol på at prosjektforbruk huskes riktig. Dokumenter reelt tidsforbruk.
6. Krev at argumentasjon for estimater er etterprøvbar, dvs viser til lignende erfaringer. Krev en svært god argumentasjon dersom ytelsen estimeres å være bedre enn historiske data skulle tilsi.
7. Ikke regn med at dere har lært noe særlig fra forrige prosjekt.
8. Ikke la ekspertutviklere estimere arbeidsmengde til uerfarne (bruk middels dyktige til å estimere middels og under middels dyktige).
9. Dersom en person har motiver for lave estimater (svært engasjert i å få prosjektet startet, indirekte belønnes for å angi lave estimater for eget arbeid, etc), finn en annen person til å estimere.
10. Kombiner estimater fra uavhengige kilder. Sørg for at disse har ulik bakgrunn og roller. Bruk en Delphi-prosess (f eks planning poker) for å kombinere vurderingsbaserte estimater.