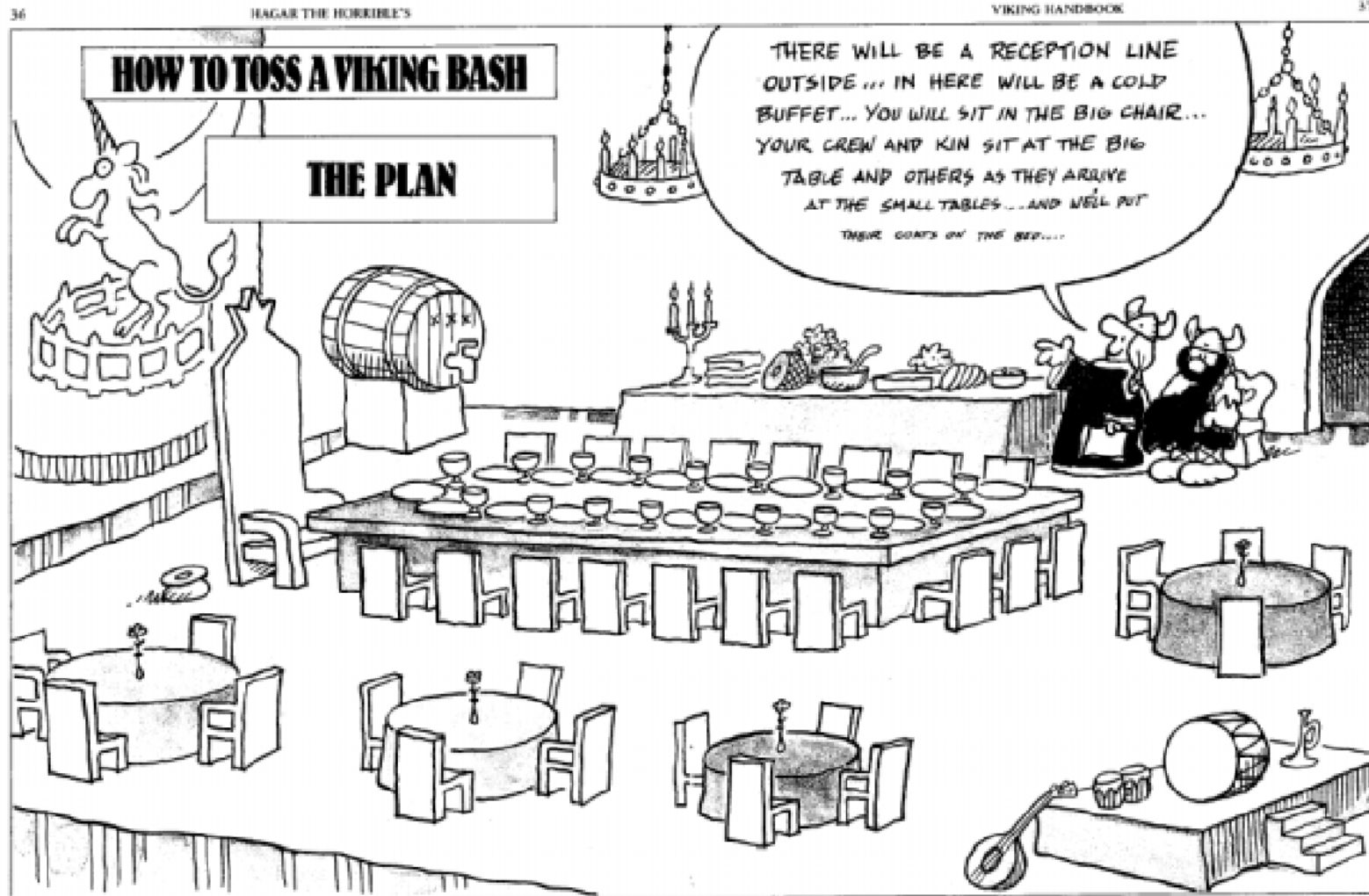


[simula . research laboratory]

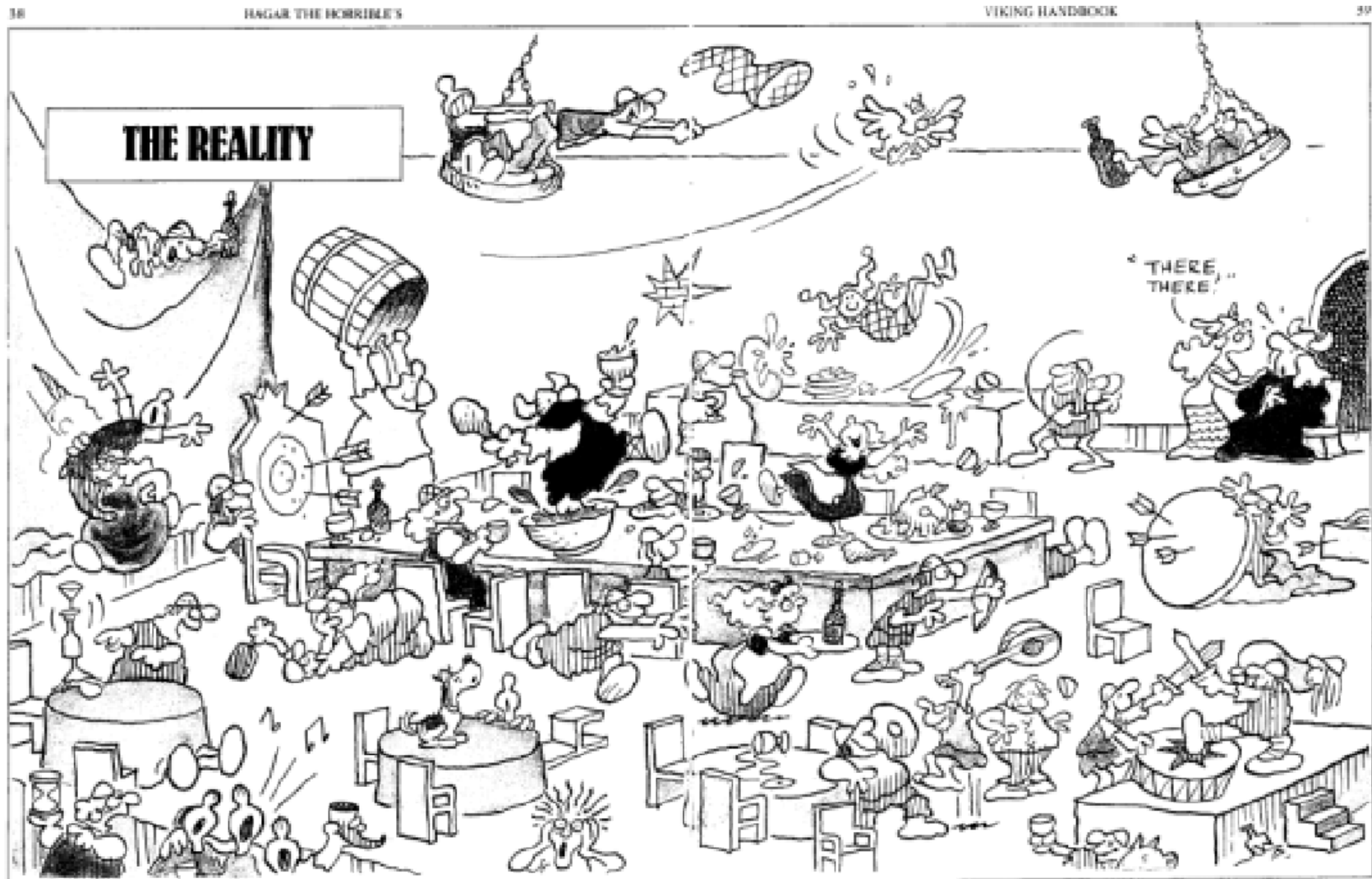
Estimering av kostnader i IT-prosjekter

**Nils Christian Haugen
Wasteless AS**

Planleggingsfasen



Gjennomføringen



Overskridelser

- Betydelig underestimering av kostnader.
 - 70-80% av prosjekter har kostnadsoverskridelser.Gjennomsnittlig overskridelser er 30-40%.

Hva er et estimat?

- En (kvalifisert) gjetning/beregning på hva et IT-prosjekt vil komme til å koste å gjennomføre (til første release). "Prosjektet vil koste kr. 50 mill."
- Brukes i anbudskonkurranser, i kontraktsinngåelser og for at diverse aktører (leverandør, kunde, prosjektledere,...) skal kunne beregne utgifter.
- Overoptimisme: For lavt estimat, altså underestimering.

- Urealistisk syn på usikkerheten i estimater.
 - Folk er eplekjekke mht egen vurderingsevne.

Hva er en usikkerhetsvurdering?

- En (kvalifisert) gjetning/beregning på hvor usikkert et estimat er. "Prosjektet vil koste kr. 50 mill. +/- kr. 10 mill., og det er jeg 75% sikker på."
- Ethvert kostnadsestimat bør være gitt med et usikkerhetsanslag.
- Eplekjeke (overconfidence): For snevert usikkerhetsintervall med for stor sikkerhet.
- 4 "Prosjektet vil koste kr. 50 mill. +/- kr. 5 mill., og det er jeg 95% sikker på."

Overskridelser

Hvorfor tror dere?

- Ingen vesentlig forbedring over tid.
 - Studier indikerer at vi er like dårlige til å estimere som for 30 år siden.
 - Ingen læring av feil eller erfaring.

- Konsekvenser:
 - **Prosjekt**: kursendring, omorganisering, avbrudd
 - **Kunder**: misfornøyde, økonomiske tap
 - **Leverandør**: dårlig lønnsomhet, tap (også av anseelse), konkurs
 - **Samfunn**: store verdier går tapt, infrastruktur kommer ikke på plass.
 - IKT er i følge SSB Norges nest største næring. Prosjekter i størrelsesorden > MNOK 100
 - Global satsning på IKT for humanitære, økologiske og økonomiske utfordringer.

- Spørreundersøkelse 2007 av mer enn 1000 IT-utviklere: (<http://certification.comptia.org/project>)
To av de tre mest kritiske faktorene ved IT-prosjektkatastrofer er relatert til kostnadsestimering.

SKARP
Au2Sys
Flexus
++

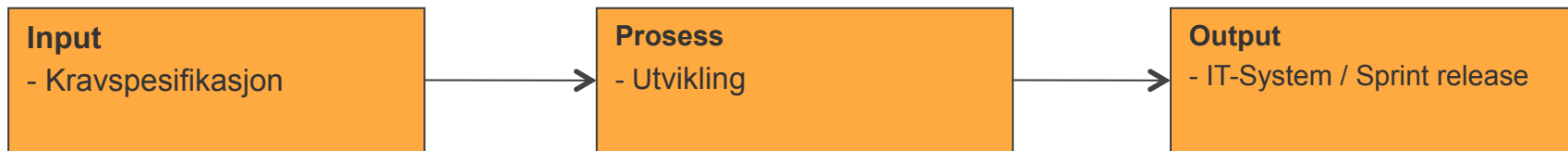
SKARP-prosjektet

- 1995: Dagens skatteregnskapssystem (Standardskatt) er over 20 år gammelt, Cobol-basert og vanskelig å vedlikeholde, og oppfyller ikke formelle krav til sikkerhet, kontroll og sporbarhet i slike systemer. Det koster også svært mye å drifte (50-60 mill kroner pr år).
- 1996: Prosjektet initiert. Dette er det største it-prosjektet direktoratet noensinne har igangsatt, med en kostnadsramme på nærmere **1 milliard kroner**.
- 2000: WM-Data får fastpriskontrakt på levering av det nye skatteregnskapssystemet (SOFIE) for Skattedirektoratet.
- 2002: Testingen av leveransene fra WM-data skulle startet tidlig våren 2002, og skulle etter planen settes i drift høsten 2002. Det er forsinkelser i prosjektet. Rykter om at WM-data allerede utvikler "gratis".
- 2003: Skattedirektoratet hever avtalen med WM-Data. Skatteetaten mener at årsaken til forsinkelsene i SKARP-prosjektet først og fremst skyldes det uføre kontrakten med VM-data medførte.
 - **VM-data taper prestisje, 250 millioner kroner, og 28 ansatte måtte gå.**
- 2003: Ny avtale inngås med Cap Gemini, basert på
 - Todelt kontrakt: SOFIE Basis (kjernen) og SOFIE Innføring (brukergrensesnittet))
 - PS2000 kontraktstandarden og
 - iterativ/inkrementell prosess.
- 2005: Pilotkommuner i drift (Stor bidragsyter for å bedre kvaliteten på systemet. Brukerstøtte sentralt. Stor utfordring som må løses: konvertering av data fra gammelt system).
- 2006: Cap Gemini inngår tre kontrakter om sluttleveranser (utvidet funksjonalitet og feilrettinger). Svært fleksibel kontraktsform i forhold til hvilke utvidelser og feilrettinger som skal med i hvilken release.
- 2007/2008: Alle skatteoppkreverne tar i bruk systemet i løpet av 2007, med unntak Oslo kemnerkontor som ikke vil ta systemet i bruk før 2008. SOFIE er basert på Oracle Applications og over 1000 egenutviklede programvaremoduler.

Hva er problemet?

- **Kostnadsestimering: Forsøk på å forutse fremtiden (forecasting).**
 - **Vellykket:** sjakk, forsikring, medisin
 - **Til dels vellykket:** vær, håndtverk, gambling (J. Shanteau)
 - **Katastrofalt:** børs og finans, kjærlighetslivet, systemutvikling
- **Komplekse systemer / Ill-structured task / Task complexity**
(H.A. Simon, K.A. Ericsson, D.J. Campbell, S. Bonner, R.E. Wood, M. Abdelmohammadi, A. Wright).
 - Systemutvikling er en kompleks ikke-strukturert oppgave
 - **Løst definerte krav som endrer seg underveis**
(behovsendringer, nye forskrifter, markedsendringer, må være sånn ellers kommer man aldri igang)
 - **Komplekse prosjekter**
(ny teknologi, vanskelig å bygge på tidligere erfaringer, tidspress, nyutvikling -- ikke produksjon)
 - **Personal-problemer**
(sykdom, avgang av nøkkelpersonale, rekruttering)
 - **Nytt IT-system medfører komplekse organisasjonsendringer**
(ikke-IT-aktige suksesskriterier, følelser og posisjonering)
 - Estimering av systemutvikling er 2. ordens kompleks og ikke-strukturert!

Task/Oppgave-definisjon

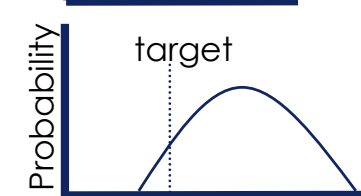
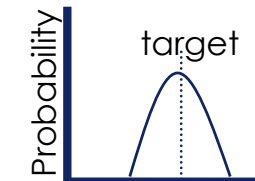
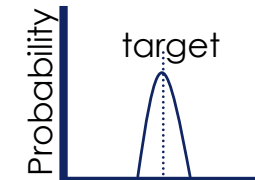
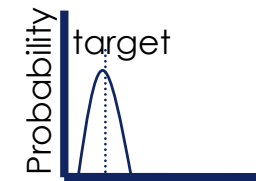
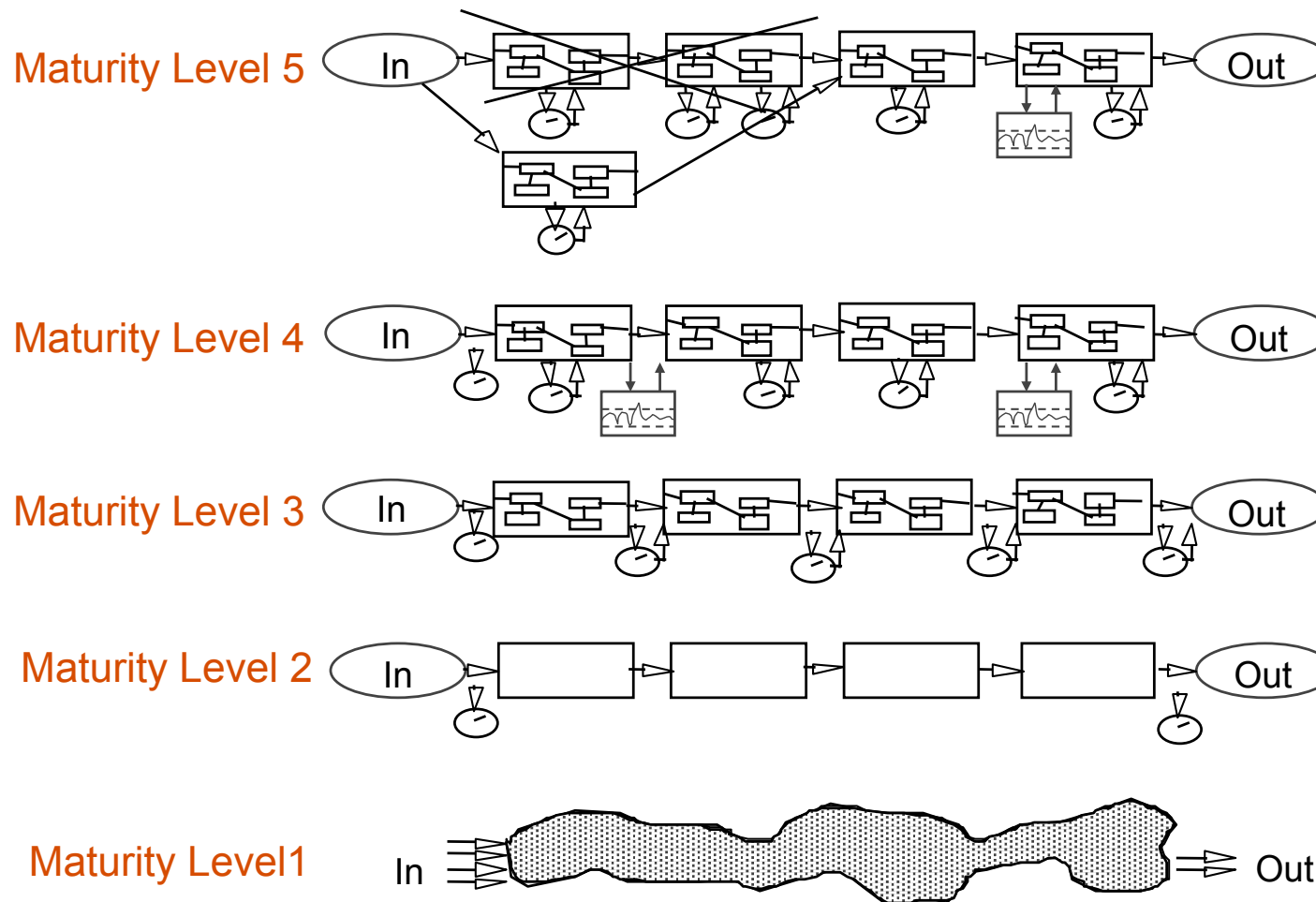


Systemutviklingsprosess

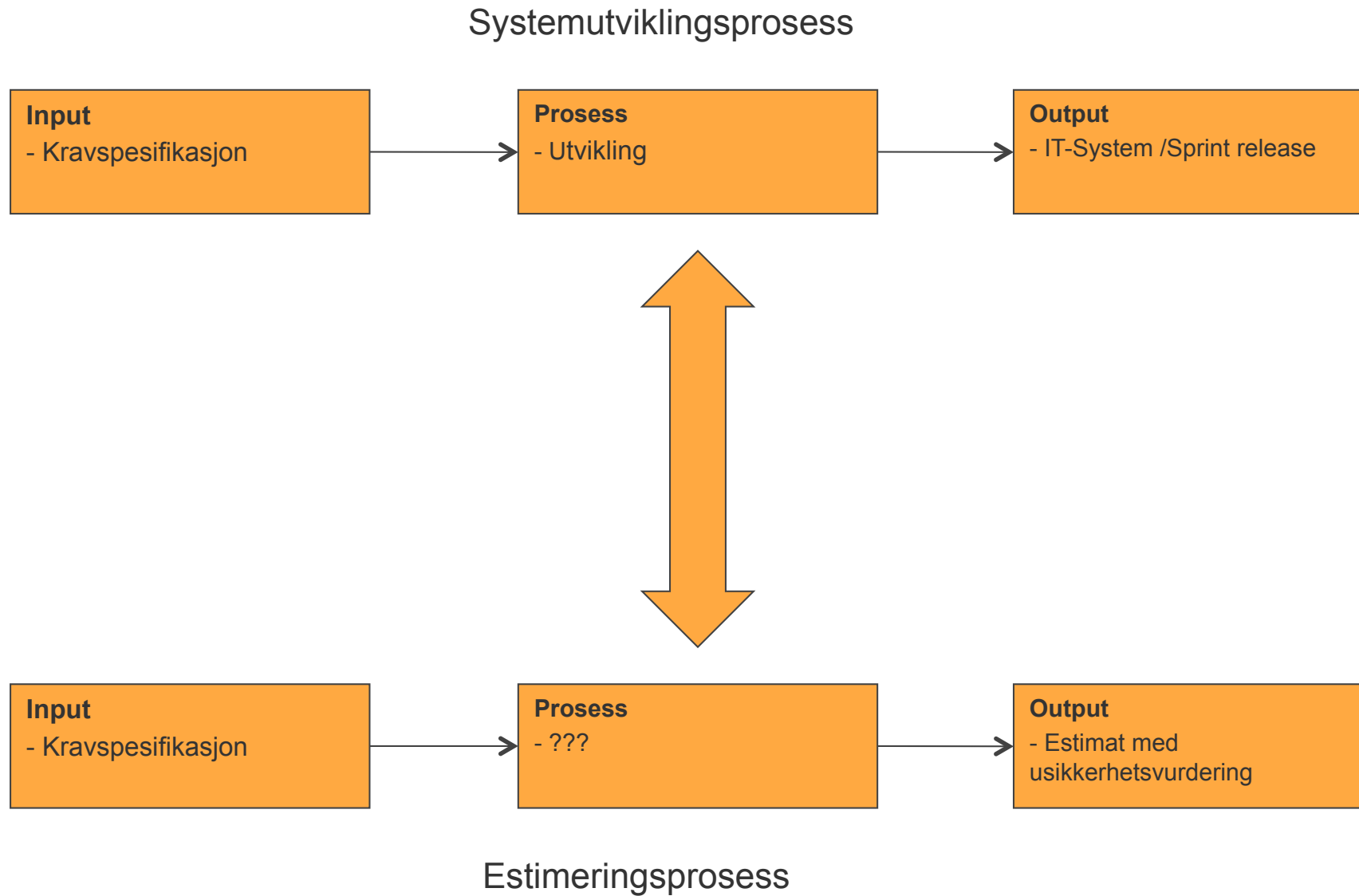
Capability Maturity Model Integration (CMMI): (hva som burde skje)

Task-definisjon

Kostnadsestimering



Task/Oppgave-definisjon

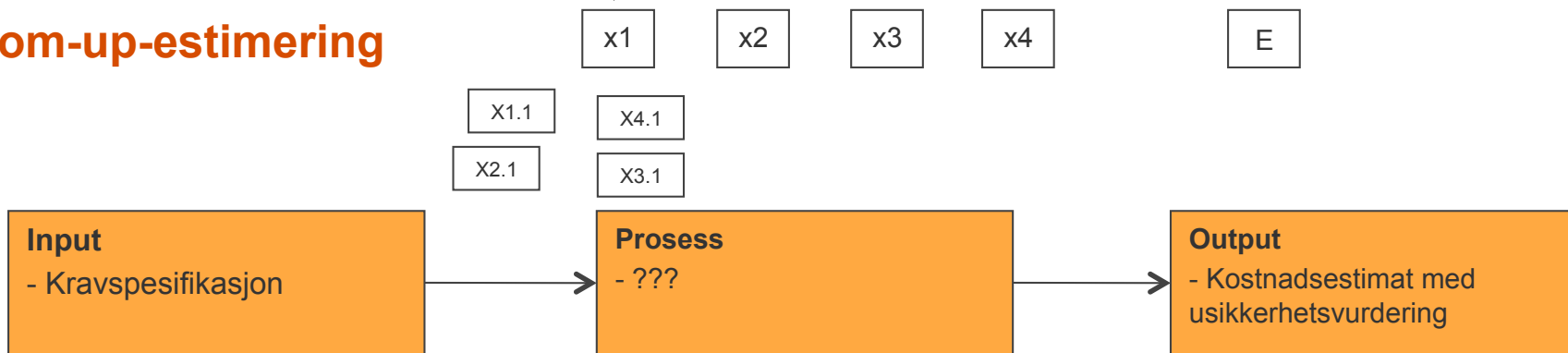


Task/Oppgave-definisjon

Systemutviklingsprosess



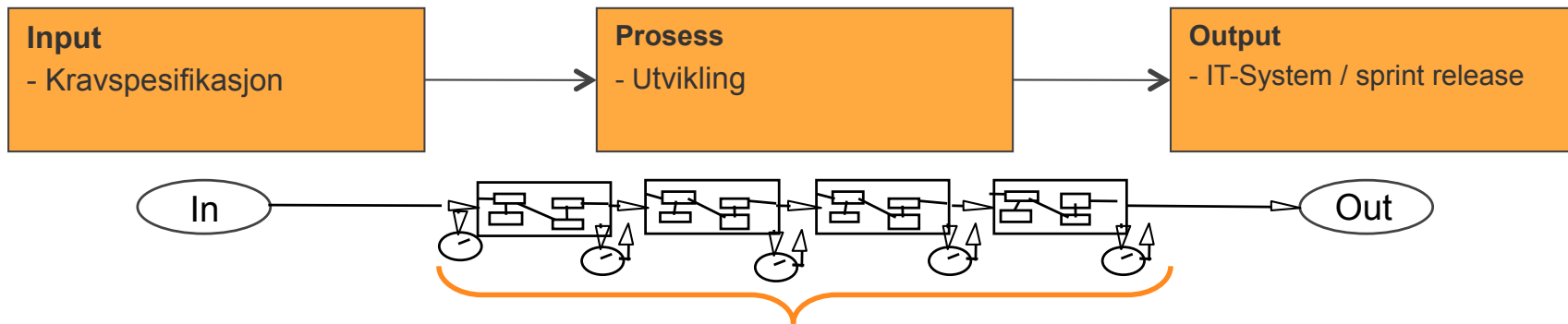
Bottom-up-estimering



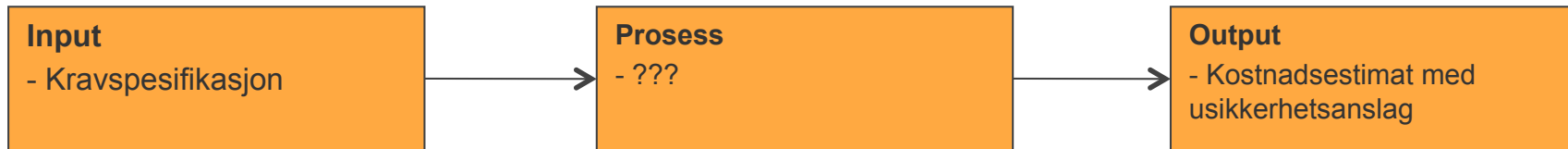
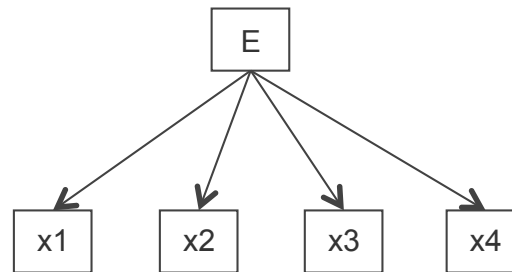
Estimeringsprosess

Task/Oppgave-definisjon

Systemutviklingsprosess



Top-down-estimering



Estimeringsprosess

Top-down eller bottom-up?

Resultat:

Magne Jørgensen, *Top-Down and Bottom-Up Expert Estimation of Software Development Effort*,
Journal of Information and Software Technology 46(1):3--16, 2004.

Donald MacGregor, *Decomposition for judgmental forecasting and estimation*,
in *Principles of forecasting: A handbook for researchers and practitioners*, 2001, p. 107-123.

- Top-down-estimering foretrukket av managers, bottom-up-estimering foretrukket av teknisk personell (programmerere)
- Top-down-estimering avhengig av at det finnes gode analogier
- Bottom-up avhengig av dybdekunnskap
- Top-down kan gi bedre estimater
- Top-down er raskere og billigere

Gjelder å finne rett nivå: Kombinasjon av top-down og bottom-up.
(det er ofte en forventning om at bottom-up er "fasiten")

Ekspert-estimering vs. Estimerings-modeller

Ekspert-estimering:

Kvantefiseringssteget er gjort på en ikke-eksplisitt, ikke-analytisk måte, muligens basert på intuisjon. Merk: Intuisjon er tillært integrert kunnskap (Hogarth: *Educating Intuition*, 2001.)

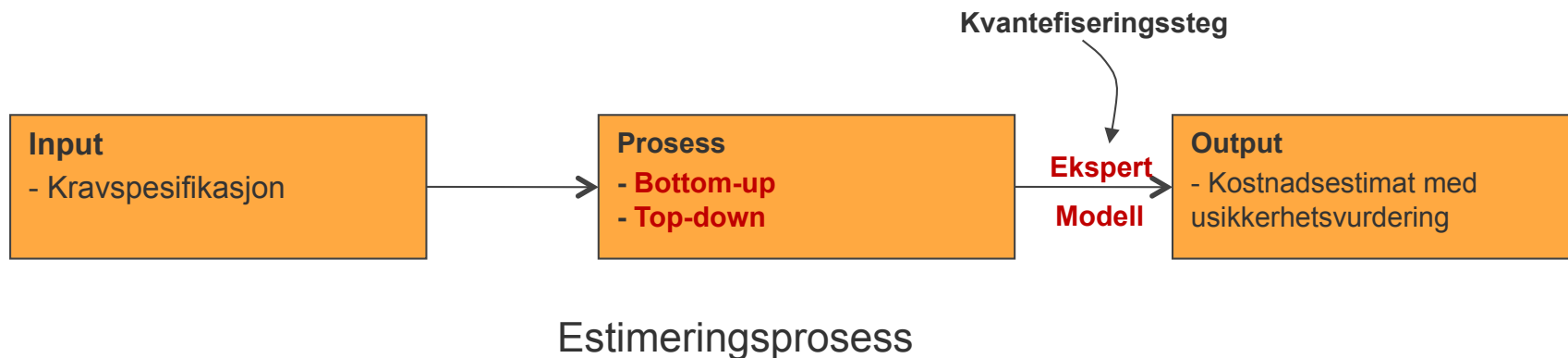
Estimeringsmodeller: COCOMO, SLIM, PRICE-S, Estimacs..., MkII Function Point, IFPUG Function Point, Feature Points, ...

Kvantefiseringssteget er gjort på en analytisk, mekanisk, statistisk måte.

Hva tror dere er best?

Resultat: I mange disipliner (økonomi, medisin, meteorologi, management) viser forskning at modeller gir bedre estimater enn eksperter gir (J. Armstrong, *Principles of Forecasting. A Handbook for Researchers and Practitioners*, 2001).

Men i systemutvikling viser forskning at ekspertestimering er bedre enn modellene! (M. Jørgensen, *Estimation of Software Development Work Effort: Evidence on Expert Judgment and Formal Models. International Journal of Forecasting*, 23(3), 2007.)



Analogi-basert estimering

Eksempel på ekspert-estimering

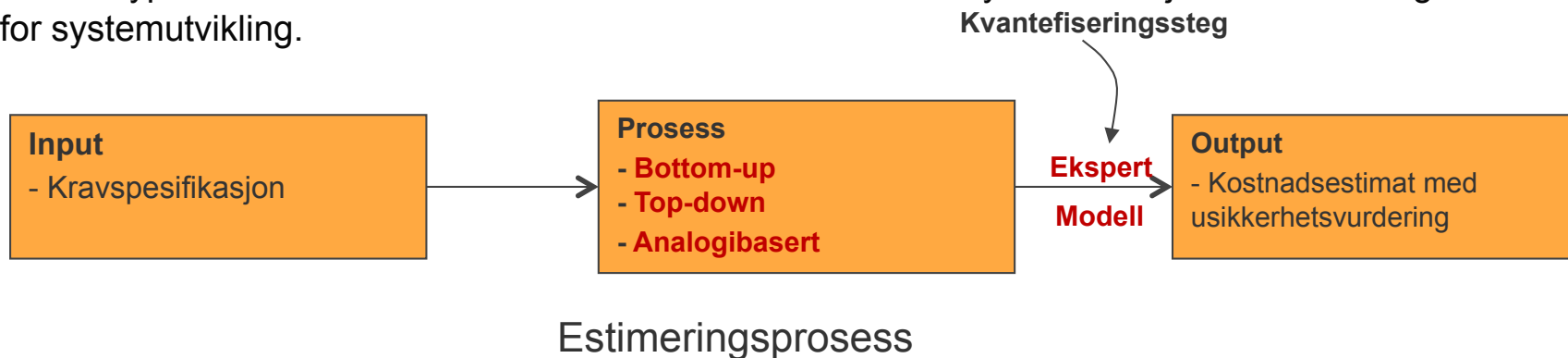
Basert på å finne analoger: liknende prosjekter, proto-/erketypiske prosjekter.

Historiske data -> Systematiserte historiske data -> Essensielle karakteristikk -> Erketyper

Resultat: Vanskelig. Det ser ut til at prosjektledere må ha analoger som er svært like til det prosjektet som skal estimeres. Man baserer seg på overflate-likheter. Det finnes ennå ingen god forståelse av dype karakteristikk av IT-prosjekter som ville gjøre det mulig å lage erketyper. Man må ha i stedet ha en stor base med historiske prosjekter.

Men det er trolig mye å hente på å gjøre forbedringer. Strebe mot å systematisere historiske data og lage verktøy som kan hjelpe prosjektledere å finne analogiske prosjekter.

På vei mot erketyper: Hva er de essensielle karakteristikkene? Hvor mye informasjon er nødvendig? Fordrer en teori for systemutvikling.



Input (cues)

Hvilke og hvor mange?

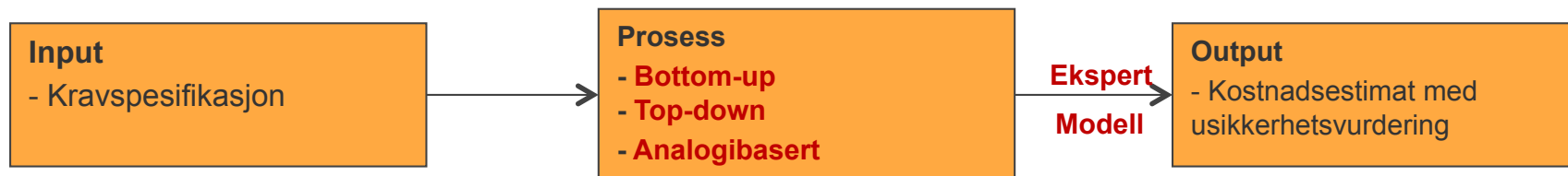
Vi er lært opp til å sanke all tilgjengelig informasjon og deretter ta en beslutning. (Rasjonelt og analytisk.)

Men det er trolig ikke slik vi har overlevd som art! Vi er ikke gode til å behandle mye informasjon.

Beslutningsstrategier som bruker få men viktige cues: Take the best (G.Gigerenzer, E. Todd, *ABC Group, Heuristics that Make Us Smart*, Oxford, 1999. G. Gigerenzer, *Gut Feelings, the Intelligence of the Unconscious*, Penguin, 2007).

Avhenger av estimeringsprosess? Ja.

Styrer estimeringsprosess? Ja.



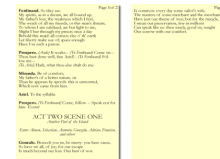
Estimeringsprosess

Input (cues)

Styrer prosjektleders inntrykk av prosjektet? Ja, mer enn man skulle tro.

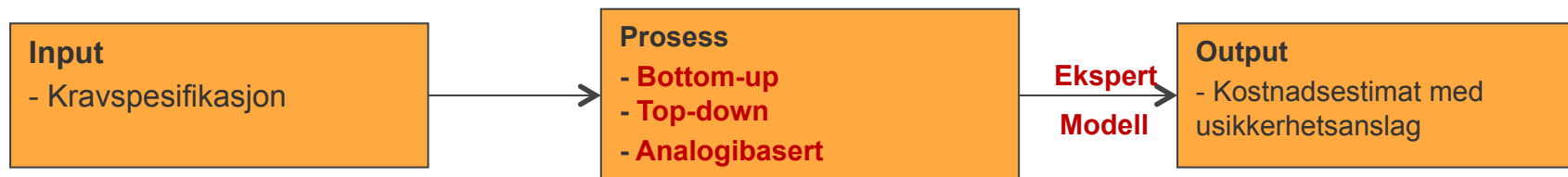
- IFI-studenter estimerte arbeidsmengde til den samme programmeringsoppgaven
 - Gruppe A: Fikk den originale spesifikasjonen, som var en side lang
 - Gruppe B: Fikk en versjon av spesifikasjonen som hadde identisk tekst, men var på syv sider. Økningen i lengde skyldes dobbel linjeavstand, vide marger, større font-størrelse og mer avstand mellom avsnittene

	Long	Normal	Difference
Mean	170	117	45%
StDev	173	98	77%



Resultat: Folk lar seg påvirke av fysisk størrelse på kravspesifikasjon, av ledende ord, av irrelevant informasjon; selv om det gjøres oppmerksom på hva som er irrelevant! (M. Jørgensen, S. Grimstad. *How to Avoid Impact from Irrelevant and Misleading Information When Estimating Software Development Effort*, IEEE Software(May/June), 2008).

Mye av estimeringsprosessen er ubevisst!



Estimeringsprosess

Mye av estimeringsprosessen er ubevisst!

Irrelevant og ledende informasjon.

Ankereffekt.

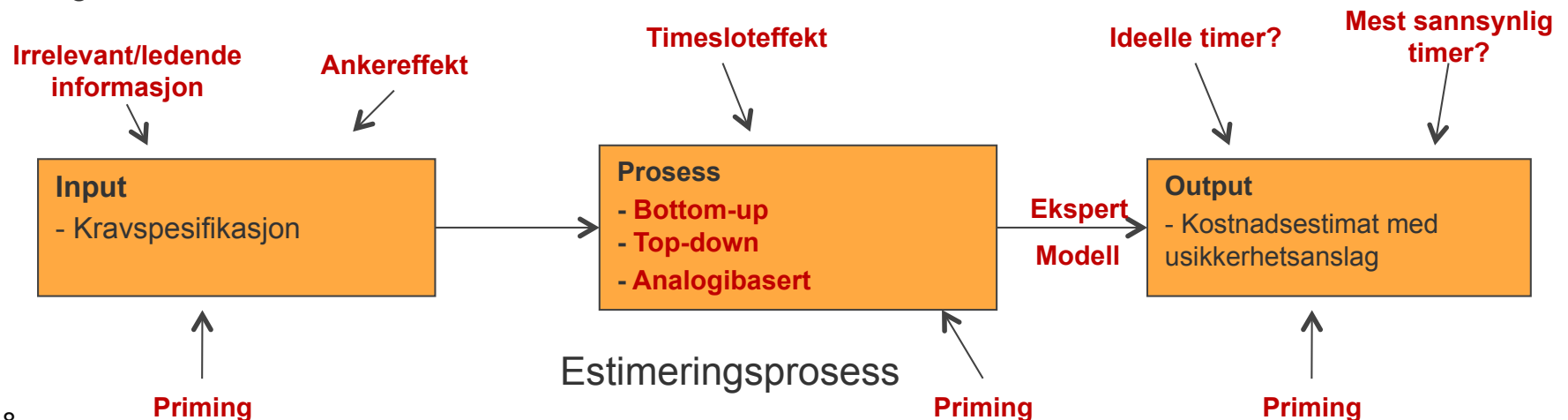
Gruppe A : Tidlig anslag 10 mill->Endelig anslag 30 mill

Gruppe B: Tidlig anslag 20 mill->Endelig anslag 50 mill

Timeslot-effekt. Det tar meg 3 dager programmere 2 user stories, men gi meg 3 dager så får jeg ferdig 4. Overoptimisme ved timeslot-spørsmålstilling. (T. Jørgensen 2009)

Hva er estimatet egentlig? Det gir mye høyere estimerer å starte med "idelle timer" (antatt ingen forstyrrelser, full konsentrasjon og topp produktivitet) for deretter å estimere "mest sannsynlig", enn å gå rett på mest sannsynlig". Dette trolig i all hovedsak fordi man i det siste tilfelle egentlig estimerer "Idelle timer". (M. Jørgensen 2009)

Priming. Eksponering til urelatert analogi-basert prosess før estimering gir analogi-basert prosess i estimeringen.

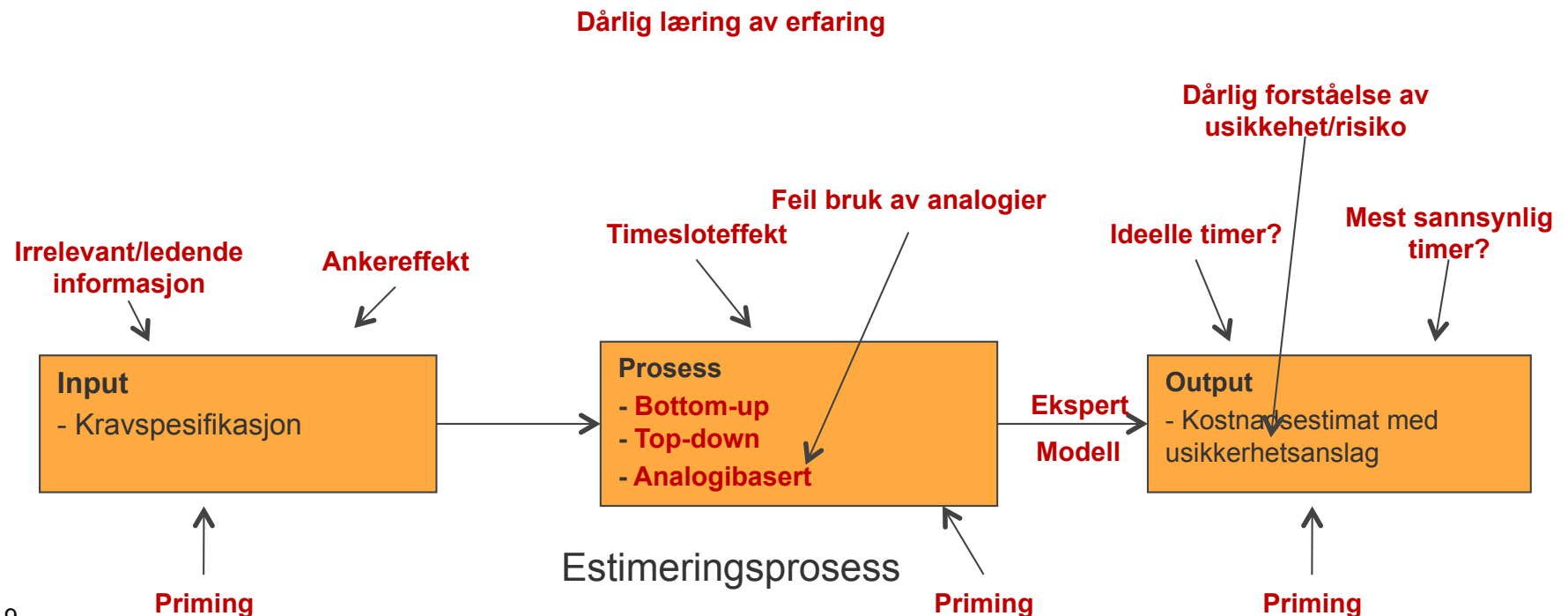


Ute av kontroll? Hva skal man gjøre?

Man kan kompensere for de ubevisste effektene dersom man forstår når og i hvilken grad de oppstår.

Man kan øke bevisstheten om fallgruber.

- Vet at det er vanlig å bruke analogier feil: Dette prosjektet er dobbelt så stort som det forrige, så da tar det dobbelt så lang tid. Feil!
- Vet at det er dårlig læring. Derfor: Ikke anta at du kommer til å gjøre det bedre neste gang! (Det vil oppstå andre vanskeligheter!)
- Vet at forståelse av risiko er ufullstendig

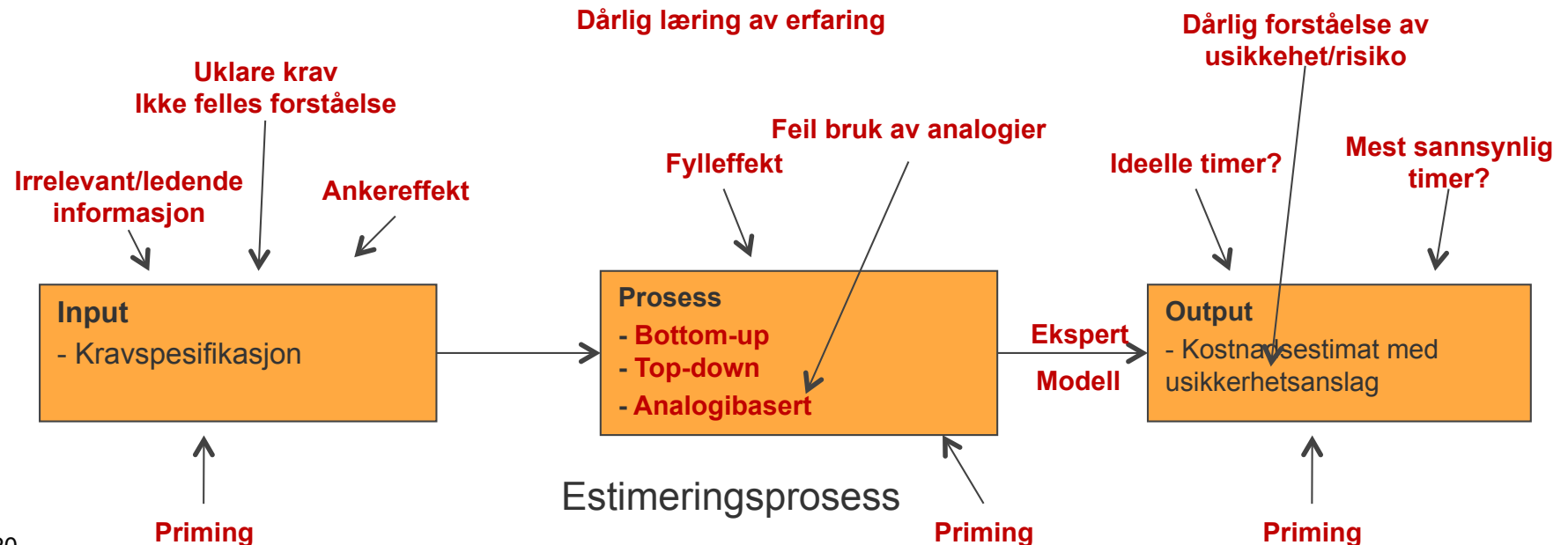


Ute av kontroll? Hva skal man gjøre?

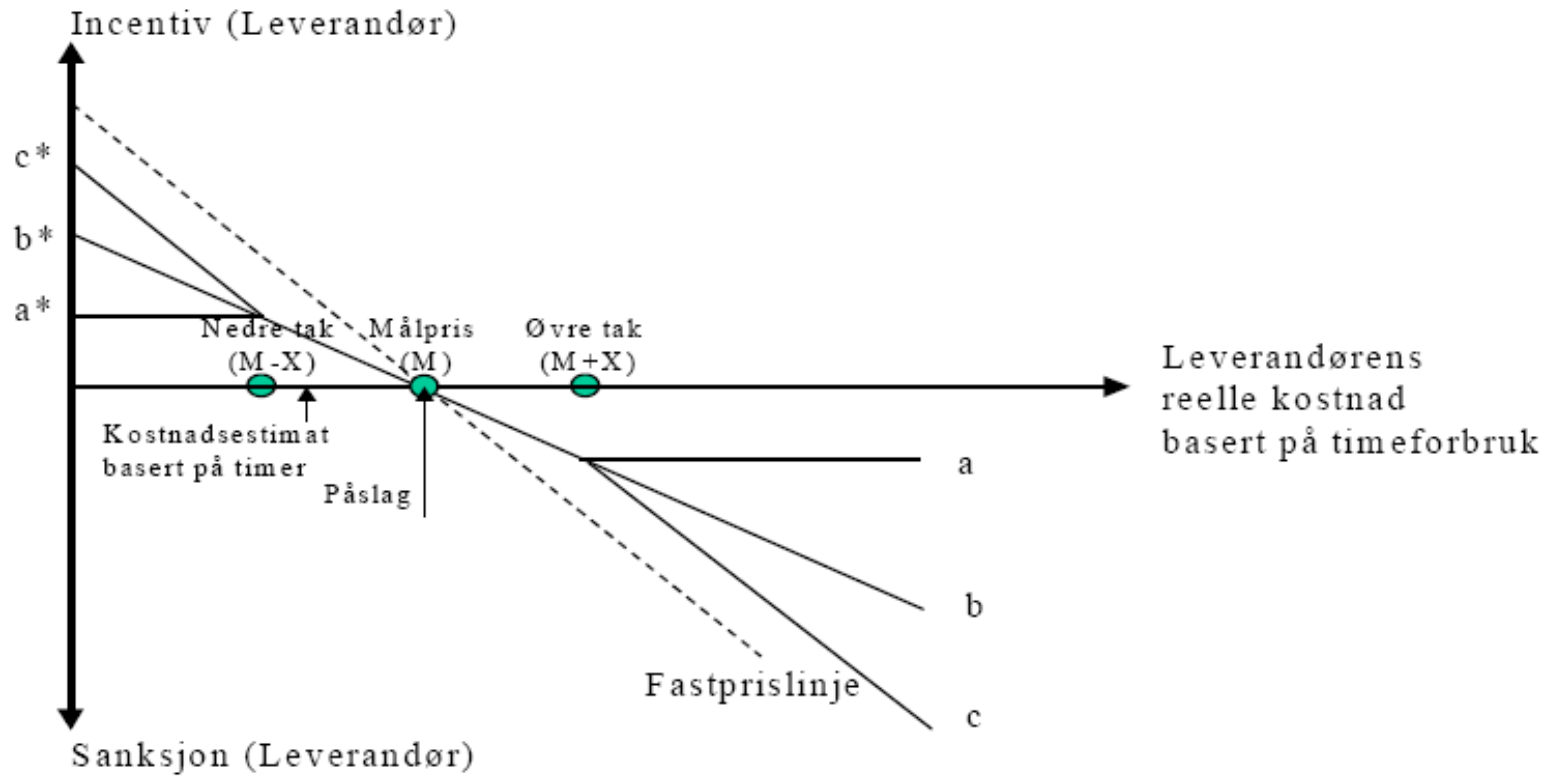
Man kan forbedre kommunikasjonen mellom kunde og leverandør:

Dette krever

- at man setter av mer ressurser til forberedelser/involvering/oppfølging (agile er vanskelig for kunden)
- at kunden blir IT-kyndig (mer aktuelt under agile/scrum)
- at man tar problemstillingene seriøst også på toppledelsesnivå
- at man bruker gode kontraktsstandarder (PS2000) som eksplisitt tar stilling til risikodeling



Incentivmodell: Usikkerhetshåndtering

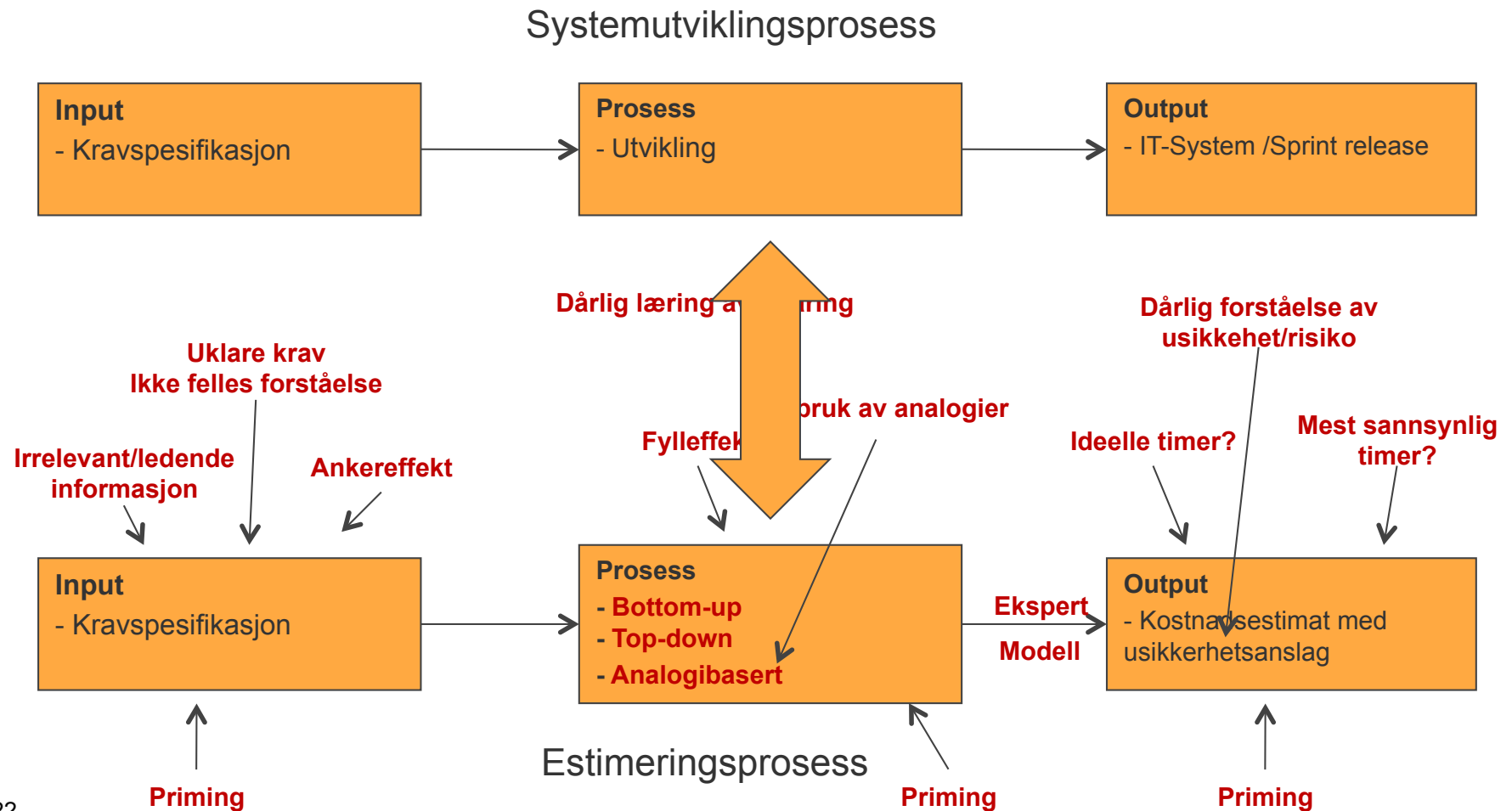


Ute av kontroll? Hva skal man gjøre?

Man kan bruke en hensiktsmessig utviklingsprosess:

Studier vi har gjennomført viste blant annet at (2003):

- Fossefall: 55% overskridelse
- Inkrementelle/iterative prosesser: 24% overskridelse



Tross alt ikke så ille

Standish Group CHAOS rapport 1994: Gjennomsnittlig overskridelse på 189%

M. Jørgensen, K. Moløkken-Østvold: CHAOS-rapporten ikke til å stole på. Biased utvalg.

(How Large Are Software Cost Overruns? Critical Comments on the Standish Group's CHAOS Reports. Information and Software Technology, 2006. 48(4): p. 297-301)

Ikke så mye verre enn andre næringer? Ikke lett å si. Mye politikk, anbudspill, osv.



Evidensbasert veiledning til estimeringsprosess

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 og nødvendig 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**
 - **Identifiser historisk data fra tidligere prosjekter**

- **4. Velg estimeringsprosess**
 - **Baser prosessen på tilgjengelig informasjon**
 - **Benytt organisasjon og personspeifikk informasjon**

Estimeringsfasen

- **5. Estimer mest sannsynlig arbeidsmengde**
 - **Strukturer estimeringsprosessen**
 - **Separer mest sannsynlig arbeidsmengde fra tilbud, plan etc.**
 - **Beskriv forutsetninger**
 - **Beskriv underliggende informasjon for etterprøvbarehet**

- **6. Anslå usikkerhet**

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 erfaringsgjennomganger**
 - **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 og eventuelle tiltak men kan gjøre
- 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

Litt om (formelle) estimeringsmodellene

- COCOMO, SLIM, PRICE-S, Estimacs, ...
- MkII Function Point, IFPUG Function Point, Feature 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 studiene som er gjennomført viser at enkle modeller 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.

Studier viser også at ekspert-estimerer, selv uten bruk av hjelpemidler, som oftest gir mer nøyaktige estimater enn bruk av modeller.

Kort oppsummering

- **Når du skal estimere arbeidsmengde for en utviklingsoppgave eller et prosjekt så bør du:**
 - Ha historiske data for lignende oppgaver tilgjengelig (eller ha tilgang på personer med svært relevant erfaring).
 - Unngå irrelevant informasjon (f eks hva personer som er mye mer erfarne enn deg ville brukt på oppgaven eller hva kunden forventer)
 - Frigjøre deg fra faktorer som fører til “ønsketenkning” (f eks unngå situasjoner der estimatet blir et middel til å signalisere effektivitet)
 - Strukturere prosessen vha sjekklister (lag din egen basert på tidligere erfaring og kombiner med andres!)
 - Kombiner estimer fra flere ulike kilder (helst uavhengige)
 - Ikke fokusere på detaljer, men på de mest usikre områdene (høy detaljering av aktiviteter fører ofte til dårligere nøyaktighet, men **større** tro på dem)



Gruppe-estimering

Eksperiment: individuell vs gruppe-estimering



- Tyve fagpersoner fra samme firma estimerte hver for seg arbeidsmengden for det samme systemutviklingsprosjektet [*]
 - Deltakerne hadde forskjellig bakgrunn
 - Prosjekt var et reelt prosjekt som var implementert
- De delte seg deretter opp i fem grupper. Hver gruppe ble enig om et felles estimat
 - Gjennom diskusjon og kombinasjon av kunnskap

[*] Moløkken-Østvold and Jørgensen (2003): Software Effort Estimation: Unstructured Group Discussion as a Method to Reduce Individual Biases. In The 15th Annual Workshop of the Psychology of Programming Interest Group

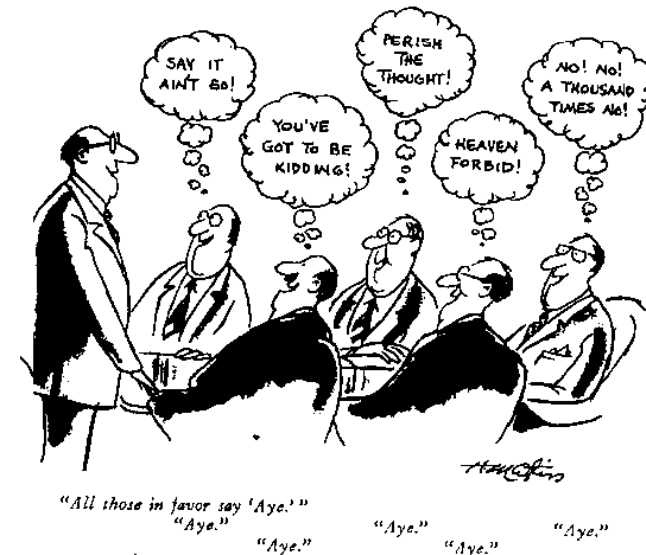
Resultater

- Estimatenes som var basert på gruppe-diskusjon var nærmere den faktiske arbeidsmengden enn gjennomsnittet av de individuelle estimatene
 - Mulig forklaring: Gruppens evne til å identifisere flere prosjekt-aktiviteter
 - Mulig forklaring: At de i gruppen måtte begrunne estimatene sine kan medføre at realisme øker
- Vi fant lignende resultater i et eksperiment hvor vi undersøkte usikkerhetsintervall [*]
 - Gruppediskusjoner medførte at man anga mer realistiske usikkerhetsintervaller

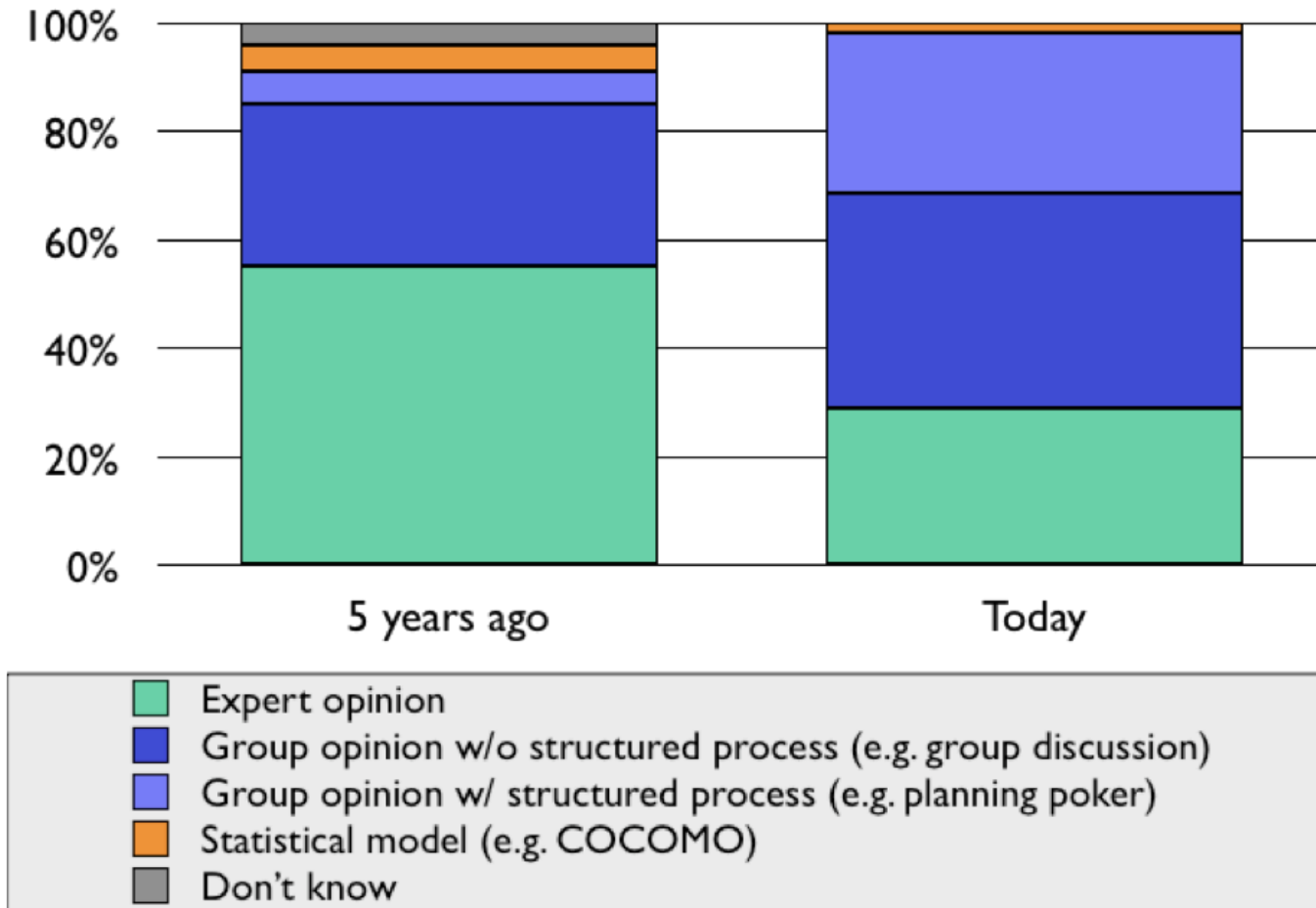
[*] Combination of software development effort prediction intervals: Why, when and how? Jørgensen and Moløkken, SEKE 2002

Forskning på gruppe-estimering

- Få studier innen Software Engineering
- ...men mange relevante studier innen andre forskningsfelt (psykologi, business forecasting, etc)
- Resultater
 - Kombinering av estimater forbedrer estimeringen (spesielt når de som estimerer har forskjellig bakgrunn). **Markets**.
 - Struktur kan forbedre estimeringen (for eksempel: redusere påvirkningen fra irrelevant informasjon)
 - ”Flere hoder husker mer”
- Ulemper
 - Ressurs-krevende (dyr) sammenlignet med individuell estimering
 - ”Group think” kan forekomme (for eksempel: 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)



Strukturert gruppe-estimerings påvirkning på opplevd estimeringsnøyaktighet (JavaZone 2007)

- 50% opplever at estimeringsnøyaktigheten var forbedret
- 30% opplever at estimeringsnøyaktigheten var uendret
- 10% opplever at estimeringsnøyaktigheten var forverret
- 10% visste ikke

Metoder for strukturert gruppe-estimering

- Planning Poker
- Wide-band Delphi



Planning Poker

- Smidig ("Agile") estimeringsteknikk
- Beskrevet av Grenning [1] og Cohn [2]

Team på 4-5 personer. En er kunde (produkt eier), resten er utviklere (leverandører) (eller andre interessenter).

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

[1] J. W. Grenning, *Planning Poker*, 2002

[2] M. Cohn, *Agile Estimating and Planning*, 2005

Estimering av relativ størrelse

- Estimér relativ størrelse, ikke varighet
 - Vi er flinkere å vurdere størrelse enn tid
 - Uavhengig av hvem som utfører oppgaven
- Alternative enheter for størrelse
 - “Story points”
 - Ideelle dager
- Bli enige om en referanse
 - Finn en en oppgave som dere vurderer til å være litt større enn de aller minste, og gi den størrelsen “2”.
 - Estimer størrelsen av resterende oppgaver relativt til referanseoppgaven
- Utled varighet under planleggingen
 - Mål prosjekthastigheten og bruk “gårsdagens vær”
 - Prosjekthastighet = summen av “story points” levert i iterasjon

Planning Poker

Team på 4-5 personer. En av dere er kunde (produkteier), resten er utviklere (leverandører).

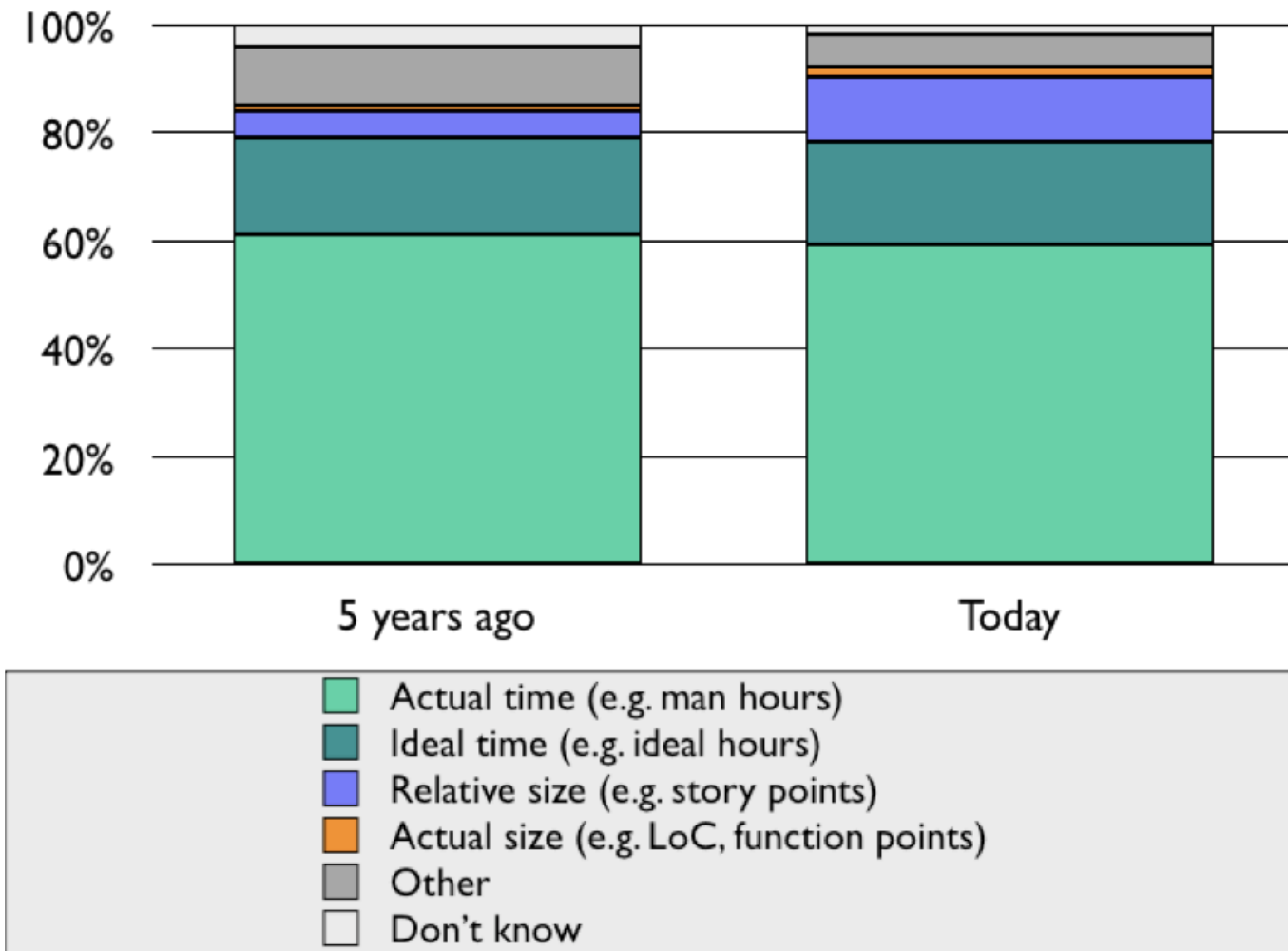
Velg en user story som referanse—en som ser liten ut men ikke minst. Sett den til 2.

1. Kunden forklarer “user story”
2. Teamet diskuterer hvilken jobb som må gjøres
3. Alle velger et kort som representerer deres estimat
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

Bør vi bruke faste eller fleksible størrelser?

- Faste størrelser er enklere og mer effektivt
 - Eksperimenter med fleksible størrelser indikerer at teamet ofte standardiserer uansett
 - Færre valg øker tempo
 - Fibonacci-sekvensen er effektiv: 1, 2, 3, 5, 8, splitt
- Husk: dette er estimer
 - Vi trenger ikke den ekstra presisjonen som fleksible estimer gir
 - Pluss/minus et par timer er ofte ikke veldig viktig

“Hva estimerer du?”



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

- 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
 - Husk at dette er estimer
- Ikke fange opp de forskjellige synspunktene
 - Mange spørsmål vil komme opp i diskusjonene
 - Viktig å ha representanter med forskjellige synspunkt tilstede

Andre vanlige spørsmål

- Hva gjør du når kunden ikke er tilstede?
 - Utnevner en av utviklerne til å presentere kravene
 - Skriver ned antakelser, og sjekker disse med kunden i etterkant
- Hva gjør du dersom du ikke har kortstokk?
 - Bruker fingre eller skriver estimatene på lapper
- Hva gjør du dersom enhetene du skal estimere i ikke passer med enhetene på kortene?
 - Tilpasser enhetene. For eksempel et kort med 1 på kan dere bli enige om at betyr 100, 2 betyr 200, osv.

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 inflytelse enn andre
- Flere spørsmål blir stilt, og mer informasjon blir delt
 - Flere hoder husker mer
 - De med forskjellige synspunkt har kompetanse innen forskjellige områder
- Flere estimerer
 - Kombinering av estimater reduserer over-optimisme
 - Estimeringsstrategiene varierer
- Estimaten reflekterer teamets gjennomsnittelige evne til å løse oppgaven
 - Ekspert-estimerer har en tendens til å basere seg på ekspertens evner
 - Dere vet ikke nødvendigvis hvem som vil ende opp med å gjøre oppgaven
- Det er gøy!

Når kan vi ellers 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

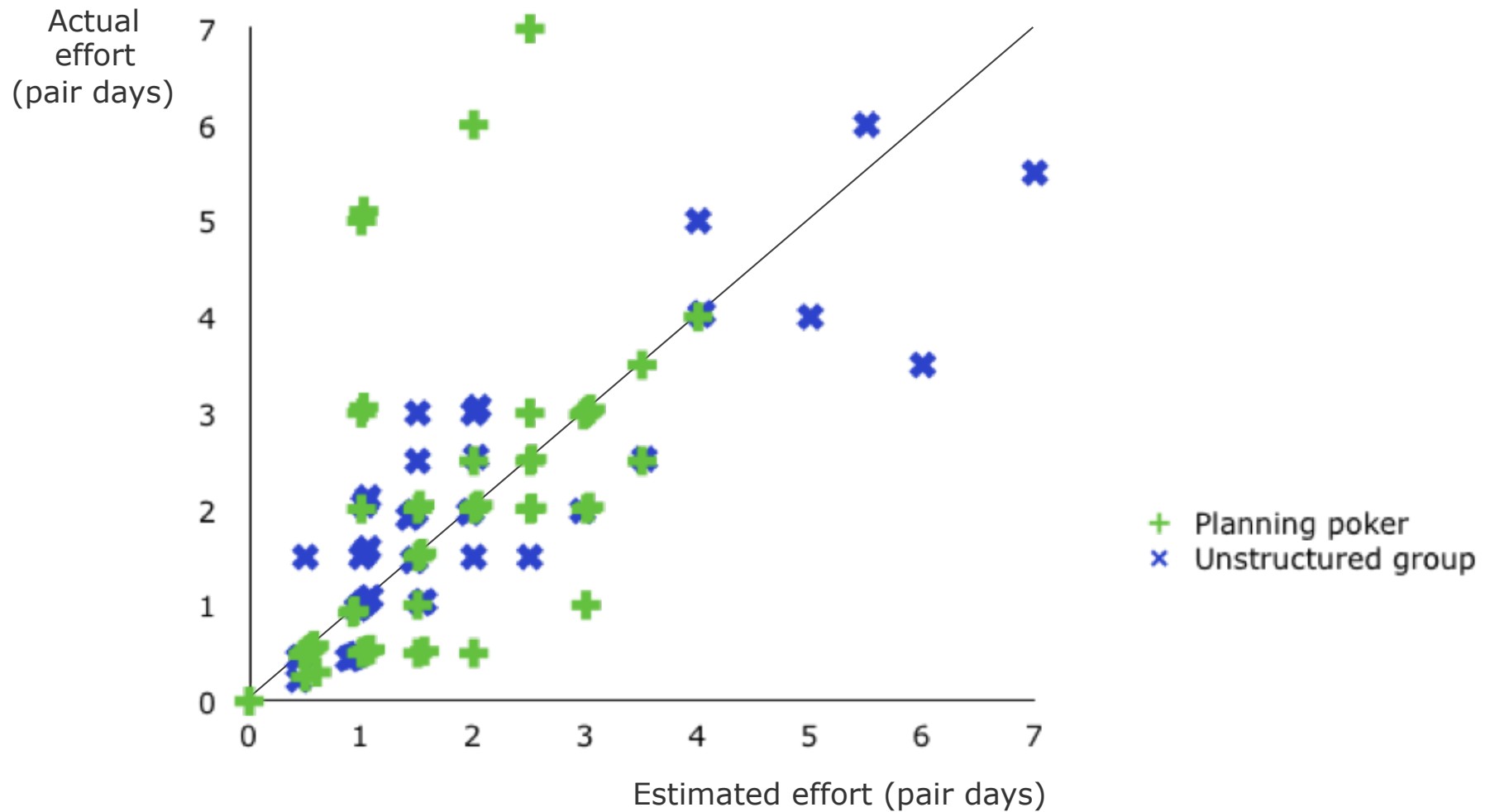
To industrielle studier

	Planning poker vs. unstructured group	Planning poker vs. individual expert
Planning scale	Release planning (2-3 months)	Sprint planning (2 weeks)
Team	8-12 developers	4-6 developers
Automated acceptance tests	Yes	No
Pair programming	Yes	No
Progress visibility	Story cards on wall	Jira
Customer view in session	Business analyst	Developers

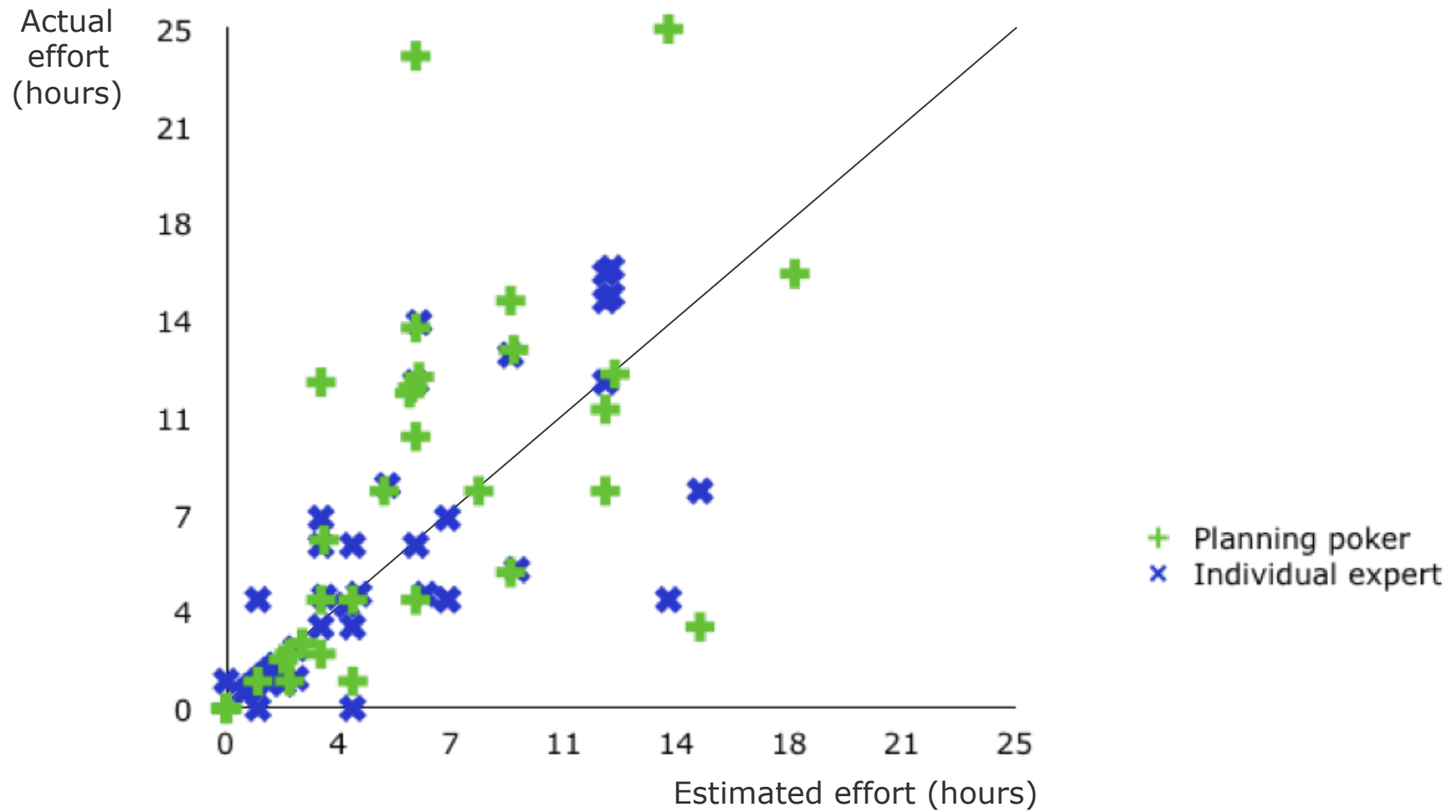
Felles for begge studiene

- Moro! Begge teamene fortsatte med dette!
- Mer effektiv estimeringsprosess
- Økt eierskap til estimatene
- Økt ansvar for projektets progresjon
- Men hvordan påvirket det estimeringsnøyaktighet?

Planning poker vs. ustrukturert gruppe-estimering



Planning poker vs. Individuell ekspert-estimering



Wideband Delphi (eksempel)

- Forbredelse av estimeringsprosessen
 - Utarbeid estimeringsmateriell
 - Velg estimeringspersonell inklusive en ordstyrer
- Kick-off-møte
 - Ordstyreren presenterer estimeringsoppgaven, estimeringsmaterialet, estimeringsprosessen, estimeringsstørrelsene, osv.
 - Gruppen diskuterer valg av eksperter, etc.
- Individuell estimering
 - Identifiser aktiviteter og estimer
 - Snakk med eksterne eksperter ved behov
- Estimeringsmøte
 - Ordstyrer oppsummerer estimatene og aktivitetslistene
 - Ekspertene diskuterer resultatene (fokuser på anonymitet)
- Oppsummering
 - Ofte gjort av ordstyrer og prosjektleder



Slutt