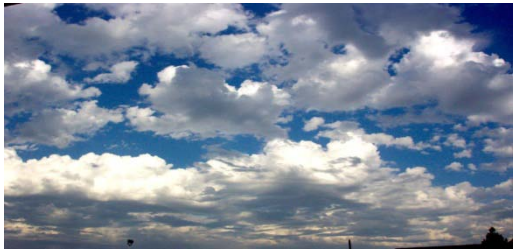
Hvordan forbedre estimering av tid og kostnader i IT-prosjekter

Magne Jørgensen
Simula Research Laboratory

Først: Litt om hvordan hjernen vår virker …

The Cocktail Party Effect ….

“Clouds Make Nerds Look Better”



- Studie av søkere på universitetsstudier (USA) viste at:
  - Det var 12% høyere sjanse å bli akseptert når det var solskinn, mot når det var mest overskyet.
  - “Nerder” hadde høyere sjanse sammenlignet med “ikke-nerder” når det var overskyet.
    - Nerde-faktor målt som “akademisk prestasjon” delt på “sosial score” (blant annet målt i antall verv).

HJELP! Hjernen min lar seg ikke kontrollere…

- Det meste av det som skjer i hjernen har vi ingen kontroll over!
- Derfor er det ofte galt å hevde ting som:
  - *"Jeg vet hvorfor jeg liker det jeg liker"*
  - *"Mitt ekspertestimat er basert på informasjon X"*
  - *"Jeg vil ikke bli påvirket i min vurdering av en middag med en mulig leverandør"*
- Dette er likevel det folk flest gjør.
- Vi liker å tro at vi har kontroll – som de rasjonelle personer vi ønsker å være.
  - Den rasjonelle reaksjonen på vår kunnskap om hjernen er imidlertid å innrømme irrasjonalitet og mangel på kontroll, og å kontrollere konteksten og mottiltakene bedre.

Mennesker har to ”hovedsystemer” for tenkning

- ”The dual-process theory of cognition”:
  - ”*Both theory and a substantial body of evidence, some of it derived from neuro-imagining studies of the brain employing fMRI technology, support the view that humans employ at least two distinct systems to process information, a rational system and an intuitively-oriented experiential system*” (Goel & Dolan, 2003)
- Når magefølelsen (hjertet, intuisjonen, din indre stemme, ...) sier en ting, mens ”hodet” (analysen) sier noe annet har vi en konflikt mellom disse to systemene.
- De intuitive systemet er høyst sannsynlig det evolusjonsmessige eldste, og det som oftest føles(!) mest ”naturlig” å følge.
- Uten kunnskap om ubevisste prosesser, blir estimatene ofte urealistiske – selv ved bruk av estimeringsmodeller, historiske data, sjekklister og andre tiltak.

Men, hvorfor så ofte overoptimistisk?

Overvurdering av egne evner er normalt hos mentalt friske mennesker!

Overvurderingen skjer i hovedsak på oppgaver som er enkle, generelle og/eller betyr noe for oss.

Undervurdering av risiko er normalt

(Risiko for egen skilsmisse er for eksempel sterkt undervurdert selv om de fleste vet at nær 50% av ekteskap sprekker)

Optimisme
er ofte
en fordel!
Mannen i grønt
må kanskje være
stor optimist for
å gidde å hoppe.

Av og til
vinner man med
høye odds -
stort sett ikke!

Over-optimisme innen IT-prosjekter

- Systematisk overoptimisme i IT-prosjekter
  - Ca. 30% overskridelse
    - En del definisjons og måleproblemer gjør at det er uklart hva dette tallet egentlig sier oss.
  - Ingen registrert forbedring over tid.
  - Ulike kulturer synes å ha samme nivå av overoptimisme i tidsestimater.
  - Overoptimisme ofte immun mot historiske data (”the planning fallacy”), særlig uten bruk av ”reference class” (analogi-basert) estimering.
  - Manglende læring skyldes delvis at intuisjonsbaserte estimater ikke kontrolleres av bevisste prosesser.
- IT-prosjekter trolig ikke dårligere enn andre (lignende) prosjekter
  - Også ca. 30% overskridelse, med tilsvarende måleproblemer
  - Stort sett ingen læring her heller

Vi trenger bedre kunnskap om årsaker til overoptimisme for å …

- identifisere situasjoner der overoptimistiske kostnadsestimater i IT-prosjekter kan forventes
- lage prosesser som fjerner (deler av) overoptimismen
- øke realismen i planene uten å ødelegge de (antatt) positive egenskapene som følger med optimisme

Biologi - 1

Evolusjonen synes å ha belønnet selvsikkerhet og risikoundervurdering (ref. bildene ovenfor).

Studier indikerer at selvsikkerhet og særlig ”narsissisme” henger sammen med overoptimisme.

En del indikasjoner på sammenheng mellom optimisme og biologi. F eks, så hjelper trolig optimisme på en del sykdomsforløp (styrking av immunforsvar), og håndtering av problematiske situasjoner (”coping”).

Simulering indikerer at overoptimisme i noen sammenhenger gir systematisk bedre utfall enn realisme. Dette gjelder særlig når man vet lite om sannsynligheten for ulike utfall, men mer om konsekvensene av utfallene.

Biologi - 2

- Det synes å være en overraskende lav korrelasjon mellom optimisme som personlighetsegenskap og optimistiske estimater!
  - Lave (overoptimistiske) estimater er altså mye mer enn optimistiske personer!
  - Kanskje ikke engang optimisme er hovedårsaken til for lave estimater (men f eks mhp vår ikke-lineære representasjon av tid, mer om dette senere).
- Menn synes å være mer overoptimistiske enn kvinner og (noe overraskende?) eldre mer enn yngre.
  - Kultur eller biologi?
- At noen har sterk overoptimisme lønner seg trolig for et samfunn, og av og til også for den enkelte.
  - Gründere er f eks mye mer overoptimistiske enn resten av befolkningen.

Gruppedynamikk

- Det synes å være et mønster at grupper fører til større sikkerhet (confidence) mhp hvor riktige vurderinger som gjøres
  - Grupper fører ofte til at vurderingens kvalitet går opp, men at troen på økt kvalitet går enda mer opp.
- Vår studie på IT-prosjekter indikerer imidlertid mindre overoptimisme ved bruk av grupper til å estimere.
  - Flere hoder husker på mer (union av aktiviteter)
  - Krav om begrunnelse kan gi økt realisme
  - Men, se opp for ”group think” og dominerende individer
  - Finner av og til eksempler på **alle** i gruppen går ut av en estimeringsdiskusjon med opplevelsen av at estimatet ble for lavt.
- Effekten av grupper på overoptimisme trolig svært kontekstavhengig, dvs vanskelig å finne generelle mønstre.
- Bruk av Delphi-metode (strukturerte grupper med uavhengige vurderinger) gir stort sett svært bra innvirkning på realisme.
  - Som for eksempel ”Planning Poker”

Motivasjon - 1

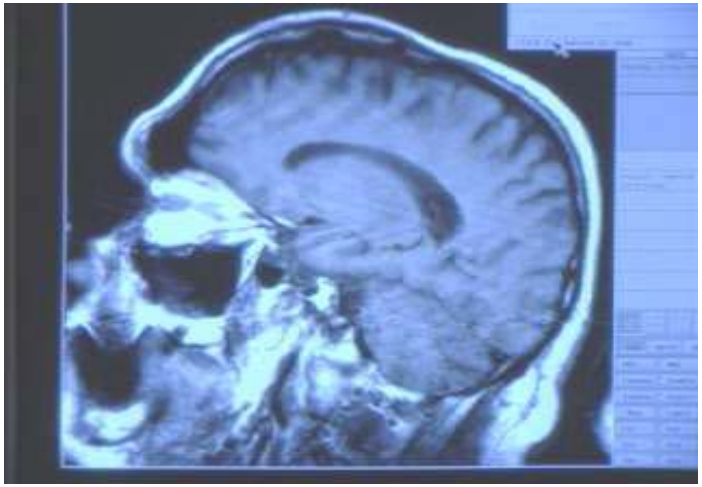
- Mange resultater viser sterk sammenheng mellom motivasjon for høy ytelse (ønsketenkning) og overoptimisme.
- Optimisme kan påvirke ytelsen positivt, MEN
  - Optimistiske estimater synes typisk å ha kun en kortvarig positiv effekt på gjennomføringen.
  - Hvor stor og hvor positiv effekten av optimisme er på ytelse overvurderes oftest sterkt.

Motivasjon – 2
(Unngå kognitiv dissonans)

- Det lønner seg med et godt selvbilde
  - For egen del
  - Selvbilde brukes som indikator på ytelse
- Derfor bør ikke et prosjekt ta lang tid, det bør ikke oppstå problemer man ikke kan løse, …
- Ubevisst, får vi oss selv derfor mer og mer til å tro på at vi skal klare det på urealistisk kort tid.
  - Hvis ikke, så får vi kognitiv dissonans (dvs sprik mellom hvem vi ønsker å være og hva vi estimerer at vi vil komme til å være)

Kognitive prosesser

- Planlegging (visualisering av fremtidsscenarier) gir ofte mindre grad av realisme enn historierefleksjoner (hvordan det har gått)!
- Overvurdering av hvor mye vi kan kontrollere/håndtere er en viktig kilde til overoptimisme (the illusion of control)
  - Dyktighet vs uflaks er ofte et spørsmål om man lykkes eller mislykkes



Estimering i tilbudsfasen/tidlig planlegging

Tilbudsfase: Viktige elementer

- Unngå ”winner’s curse”
  - ”*Vi vinner nesten bare når vi har vært over-optimistiske*”. Hva gjør vi med estimeringen og prisingen når konkurransen er stor?
  - M. Jørgensen. How to Avoid Selecting Providers with Bids Based on Over-Optimistic Cost Estimates, IEEE Software (May/June) 26(3):79-84, 2009.
- Vær nøye på å skille mellom prising og estimering.
  - Bør være adskilte prosesser og helst ulike personer.
  - Unngå alle ”anker” i estimeringsarbeidet (men disse kan godt være med i prisingsarbeidet).
- Fjern villedende og irrelevant informasjon **før** de som estimerer får den.
- Husk at dere oftest er mer over-optimistisk i prisingen, jo mindre relevant erfaring/kunnskap dere har.

Ankereffekt: Kundens forventninger

Eksperiment:

- HIGH (LOW) group: *"The customer has indicated that he believes that **1000 (50)** work-hours is a reasonable effort estimate for the specified system. However, the customer knows very little about the implications of his specification on the development effort and you shall not let the customer's expectations impact your estimate. Your task is to provide a realistic effort estimate of a system that meets the requirements specification and has a sufficient quality."*
- Deltakere: Erfarne systemutviklere.
- Alle (HIGH, LOW, CONTROL) fikk samme kravspesifikasjon.
- M. Jørgensen, and S. Grimstad. **Avoiding Irrelevant and Misleading Information When Estimating Development Effort**, IEEE Software(May/June):78-83, 2008.

Resultat: Effekten av kundens forventninger

- HIGH group gjennomsnitt: 555 timeverk
- CONTROL group (uten forventinger) gjennomsnitt: 456 timeverk
- LOW group gjennomsnitt: 99 timeverk!!!
- **NB**: Ingen av utviklerne opplevde at de hadde blitt mye påvirket. De fleste mente at de ikke var påvirket i det hele tatt av kundens forventninger.

Feltstudie av det samme ….

"The preliminary budget of the new system is $10 000 [corresponding to about 100 work-hours with typical pricing in the country in which it will be built]. The preliminary budget is not built on any knowledge about the actual cost of developing the new system, and will, if needed, be extended to cover the expenses necessary to build a quality system with the desired functionality."

100 timeverk var en svært lav verdi for dette prosjektet og alle firmaene ble instruert om å IKKE bruke dette som input til estimatet.

Firmaene som deltok visste ikke at de var med i en studie, men skulle gi uavhengige estimater på et arbeid som skulle utføres **av andre (mao ingen direkte grunn til å være optimistiske)**. Firmaene ble betalt for estimeringsarbeidet.

Resultater fra feltstudien

Numerical Anchor

Group	**Median estimate**
Manipulated (client's expectation, 100 work-hours)	724 work-hours (n=23)
Ordinary	956 work-hours (n=23)

Mindre effekt en i "laboratoriet", men fortsatt en effekt av betydning.

Trolig større effekt dersom man er i tilbuds-modus.

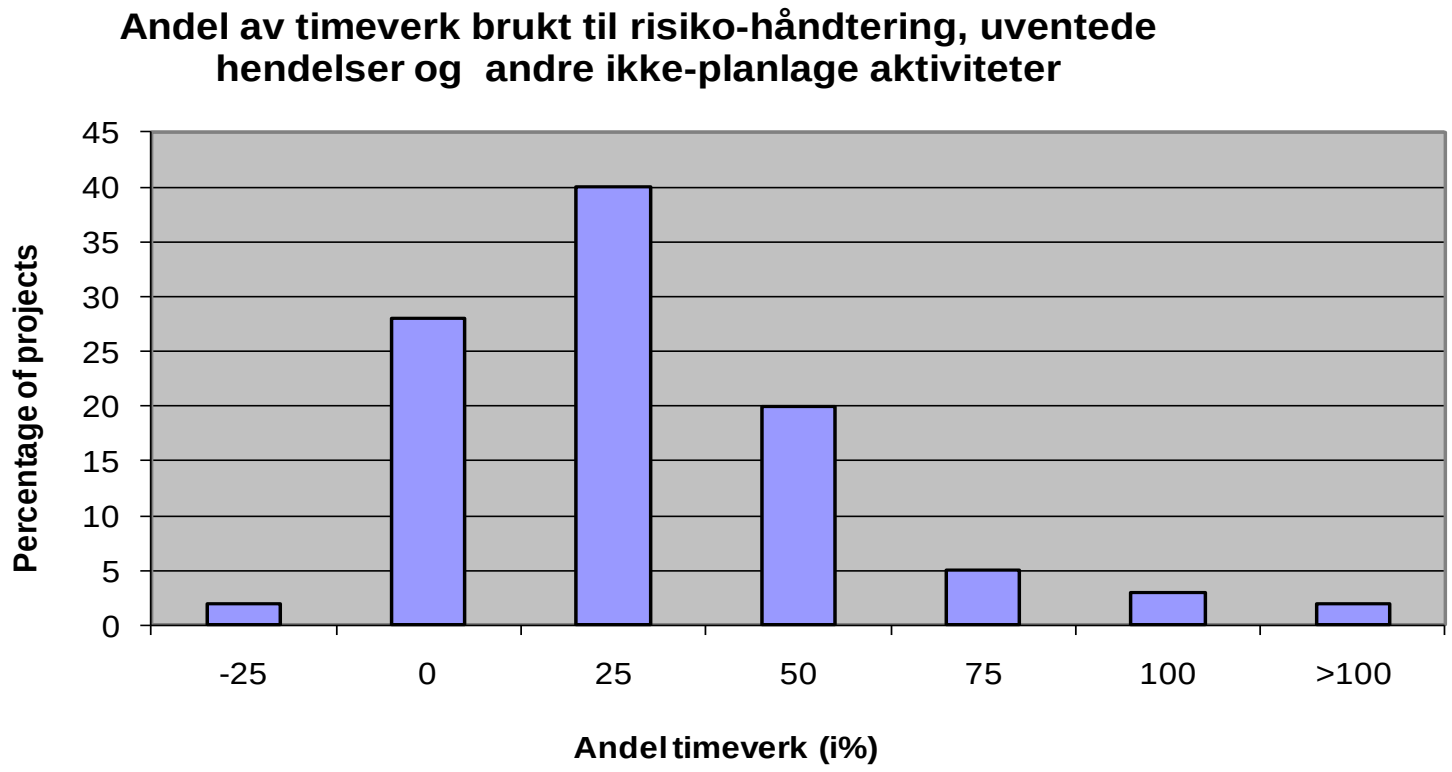
Tidligestimater av prosjekter

- Ingenting slår eksperters erfaring med svært like prosjekter.
  - Dersom slik ekspertise ikke er tilgjengelig, bør man enten skaffe denne kompetansen eksternt eller skaffe relevante historiske data.
- Work-break down (bottom-up estimering, ”looking forward”) er bra, men medfører ofte at man setter av for lite til ”ukjente” og ”uventede” aktiviteter.
  - Dersom mulig, gjør en ”spike” (læringsleveranse, forprosjekt) på de mest usikre delene av prosjektet. Bruk historiske data for å se hvor mye arbeid som typisk går med til ikke-planlagte aktiviteter.
- Mulig prosess:
  1. Estimer kjente aktiviteter i ”ideelle timeverk” (bruk erfarne personer og historikk så mye som mulig).
  2. Estimer deretter ”mest sannsynlig” timeforbruk for kjente aktiviteter, der forskjellene mellom ideelt og mest sannsynlig forbruk er i fokus
  3. Gjør en grundig risikoanalyse
  4. Gi et risikopåslag ut fra historiske data om hvor mye som må påregnes ut fra risiko og uventede hendelser. [NB: Dette fører til estimert, men ikke planlagt arbeidsmengde]

Hvorfor starte med ”ideelle timer” og vente med risikoanalyse?

- JavaZone 2009:
  - Ser ut som om utviklere estimerer i retning av ideelle timer uansett om man spør om ”mest sannsynlig” eller ”ideelle timer”.
    - Hvorfor da ikke endre spørsmålet, i stedet for å forsøke å få dem til å svare på det man spør om?
    - **JavaZone-eksp**: 33% økning fra ”ideelle” til ”mest sannsynlig” timer, mens kun 18% reduksjon fra ”mest sannsynlig” til ”ideelle” timer.
    - Støttes av to studier fra andre domener.
- Bruk av ”ideelle timer” (om enn noe annerledes bruk enn i smidig utvikling) ser ut til å gjøre oss mer bevisste om antagelsene for estimatene. Påslaget i etterkant for å få et realistisk estimat, blir dermed trolig bedre.
- Mye risikoanalyse **rett før** estimering gjør oss mer optimistiske! (Har flere studier på dette. Studier vil bli publisert i tidsskriftet ”Information and Software Technology”).

Hvordan bestemme risikopåslag …
(NB: Min-max intervaller fungerer svært dårlig!!!!)



Viser f eks at 30% av lignende prosjekter har brukt 50% eller mer på ikke-planlagte aktiviteter

Story-point basert estimering (Estimering av Release)

Hva er story points?

- Et knippe av Mike Cohn's forklaringer:
  - ”pure measure of size”; ”units of relative size”; “measurement of complexity and/or size of a requirement”; “an amalgamation of the amount of effort involved in developing the feature, the complexity of developing it, the risk inherent in it, and so on.”
- frank.vanpuffelen.net/2008/02/scrum-story-points-ideal-man-days-real.html
  - “What is a story point? The correct Scrum answer would be that it doesn't matter what unit it is. But of course there is a different unit behind the story points. …. So a story point is a so-called "ideal man day".
- www.renaissancesoftware.net/blog/archives/19 (James Grenning)
  - “When we give one story a four and another a two, we are saying the four is twice as difficult, will take twice the effort, as the two.”
- Norsk IT-utvikler: “*Gitt konkrete eksempler forstår folk dette veldig fort. Men å forklare det akademisk strever jeg fortsatt med selv etter mange år*.”
- **Relativ størrelse, kompleksitet, risiko, “ideal days”, relativ arbeidsmengde, relativ varighet, kombinasjon eller udefinert????**

Historien til ”Story Points”

- **Ron Jeffries**: *”As far as I know, story points started at Chrysler, on the C3 project, the first Extreme Programming project. Kent Beck originally suggested that the team estimate stories in days. We observed that we could estimate "perfect days" pretty well, that is, how long a story would take if we didn't get distracted by all the other things going on. However, perfect days caused trouble in communication, because a "three day" story always took longer than three days, leading to questions from management and others. So we decided to use an abstract number instead, which we called a "point". We actually used one point = one perfect day ourselves, though of course they can be scaled any way one wants.”*
- **Dette ser for meg ut som:**
  - C3-prosjektet innser at dagsverk (arbeidsmengde) og dager (varighet) ikke er sammenfallende og at det er lettere å estimere ideell arbeidsmengde enn varighet.
  - De innfører ”perfect days” (dvs, tiden man jobber på en arbeidsoppgave uten forstyrrelser), men ser at skille mellom arbeidsmengde (perfect day) og varighet fortsatt er vanskelig å kommunisere.
  - Defor ”fjerner” man koblingen til arbeidsmengde ved at man kaller det ”points”.

Antatte fordeler med”Story Points”

Fra Wikipedia (trolig skrevet av Mike Cohn)

- They are cheaper to arrive at. ***[sammenlignet med hva?]***
- They foster collaborative estimation. ***[bygger trolig på antagelsen om at story point er lettere å kommunisere enn alternativene]***
- They form a **consistent** measurement - the *relative* size/complexity of one requirement to the other does not shift as much as the effort/duration necessary to complete it. ***[gjelder ikke generelt, men kanskje vanligvis]***
- They allow planning with a level of ambiguity in the requirements. ***[Ikke akkurat unikt for story points.]***
- Points need not change if the relative sizes of the requirements (compared to each other) are not generally changing (cheaper re-estimation). **[Trolig riktig – i forhold til direkte estimering av timeverk]**

Klargjøring av Story Points

- **Antagelse**: Viktigste kriterium for riktig forståelse av story points bør være at det fremmer å kunne estimere arbeidsmengde og/eller varighet på en effektiv og rimelig nøyaktig måte. Best nøyaktighet i estimatene oppnås dersom forholdet mellom ”story point” og arbeidsmengde/varighet (dvs, forholdet beskrevet som ”velocity”) varierer minst mulig innen samme team/prosjekt.
- **Implikasjon**:
  - **Prosentvis forskjell i ”story points” være identisk med %-vis forskjell i arbeidsmengde.**
    - To ganger som mange story points bør f eks tilsvare to ganger så mye arbeid.
    - Story points bør da beskrives som et estimat av ”relativ arbeidsmengde” (som tar størrelse, kompleksitet, risiko og ”and so on” som inndata!).
    - Velocity bør referere til arbeidsmengde, ikke varighet.
    - Estimering av varighet bør ha en tilleggsprosess der man justere for antall arbeidsdager, variasjon i tilgjengelige ressurser, etc..
  - Det kan være tilfeller der en annen forståelse av ”story points” gir bedre effektivitet – f eks der ”størrelse” er svært lett å måle og korrelerer tilstrekkelig med arbeidsmengde til å brukes, eller at teamets innsats er svært konstant over tid.

Noen myter om story points

- **Story Points er “unit less”**: Det at enhetene til story points ikke er vel-definert, universelle eller refererer til synbare eller fysiske enheter er ikke det samme som at det er enhetsløst. Det har enheten “story points”. Vi bruker ofte samme navn både som navn på måling og enhet, for eksempel er timeverk ofte brukt både som målestørrelse og enhet.

- **Story points gir en relativ og arbeidsmengde en absolutt måling og estimering**: Arbeidsmengde (f eks timeverk) og story points har de samme “relative” egenskapene. Forskjellen ligger i hvor universell og veldefinert enheten er. Arbeidsmengde-estimering gjøres typisk relativt, i likhet med story points.

- **Systemutviklere er bedre i å estimere relativt enn absolutt**: Dette er enten et meningsløst utsagn, fordi estimering av IT-prosjekter som ikke er basert på sammenligninger (er relativ) neppe finnes. Dersom det menes at forholdet mellom “estimerte story points” og ”faktisk timeverk” samvarierer bedre enn “estimerte timeverk” og “faktiske timeverk”, så mangler dokumentasjon. (kommer nærmere inn på det)

Noen myter om story points

- **Velocity (SP/work-hours) måler “hastighet” (working speed)**: Velocity er en konverteringsfaktor mellom to målinger av arbeidsmengde, med forskjellig enhet. Som en konvertering mellom mitt lengdemål “skritt” og meter.
- **Story point gjør at vi ikke trenger å estimere dagsverk/dager**: Kalendere og klokker styrer fortsatt hverdagen …

Sammenligning av egenskaper

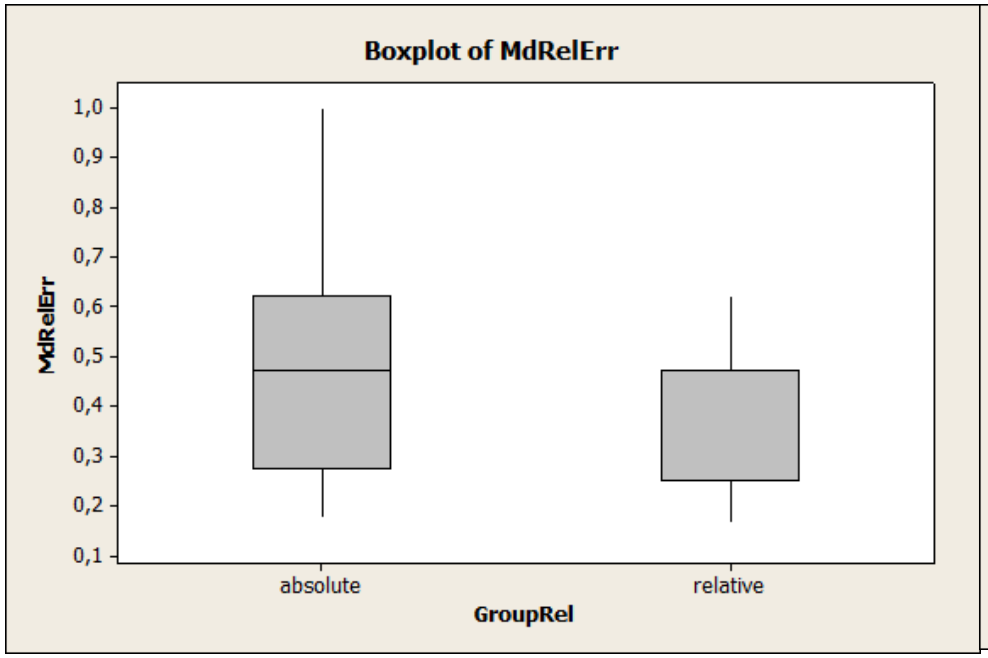
	Timeverk	Ideelle timer	Story points
Klarhet i konseptet	OK	OK-	Nei
Lett å bruke	OK	OK	OK+
Lett å forstå for kunde	OK-	Nei	Nei
Egnet input til estimering av varighet/innhold	OK-	OK-	OK (gitt historikk)
Læring underveis	OK-	OK-	OK+
Lett å re-estimere	OK-	OK-	OK+
Nøyaktighet	?	?	?

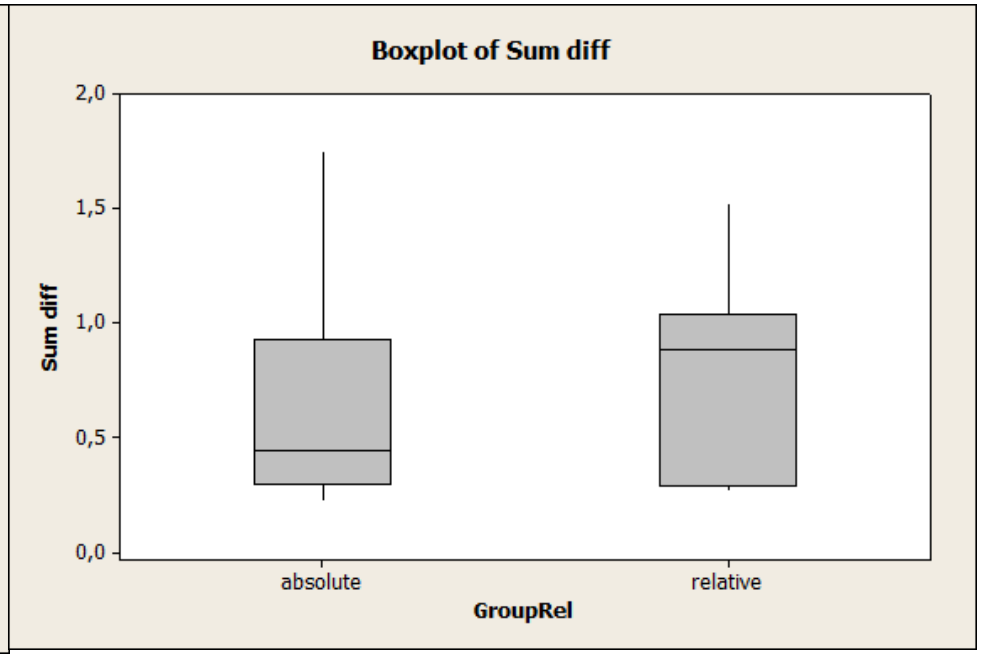
Er vi bedre til å estimere relativt?

Eksperiment 1:

1. Eksempel på relativ estimering (med lengden på elver)
2. Rangering av antall innbyggere i fem store byer (Berlin, Warsawa, Napoli, London, og Praha)
3. Velg byen som er rangert som 3. størst og gi den 10 “by-poeng”. Gi de andre byene by-poeng relatert til antall innbyggere.
4. Anslå antall innbyggere for den byen du tror du kan gi det mest nøyaktige estimatet.

Resultater





Indikerer at:

1) Nøyaktigheten ble bedre med relativ estimering, MEN

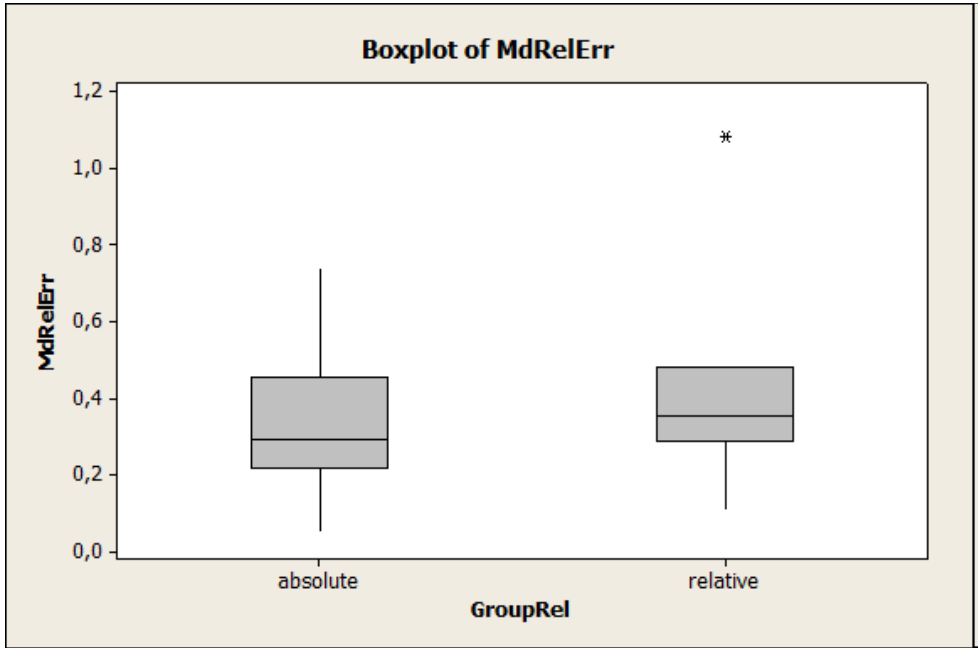
2) De relative forskjellene var bedre representert ved estimering med fysisk enhet (antall innbyggere)!!!

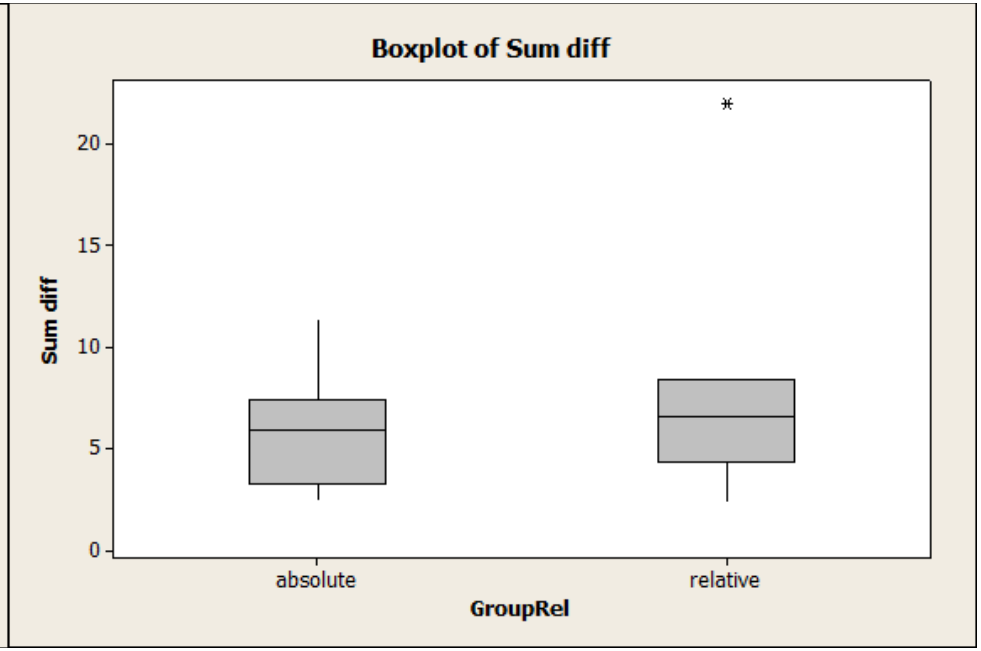
Er vi bedre til å estimere relativt?

Eksperiment 2:

1. Eksempel på relativ estimering
2. Rangering av vekten på fem dyr (løve, isbjørn, jaguar, hyene, tiger)
3. Velg dyret som er rangert som 3. størst og gi den 10 “dyre-poeng”. Gi de andre dyrene dyre-poeng relatert til vekt.
4. Anslå vekt for det dyret der du tror du kan gi mest nøyaktige estimater.

Resultater





Indikerer at:

1) Nøyaktigheten ble **ikke** bedre med relativ estimering,

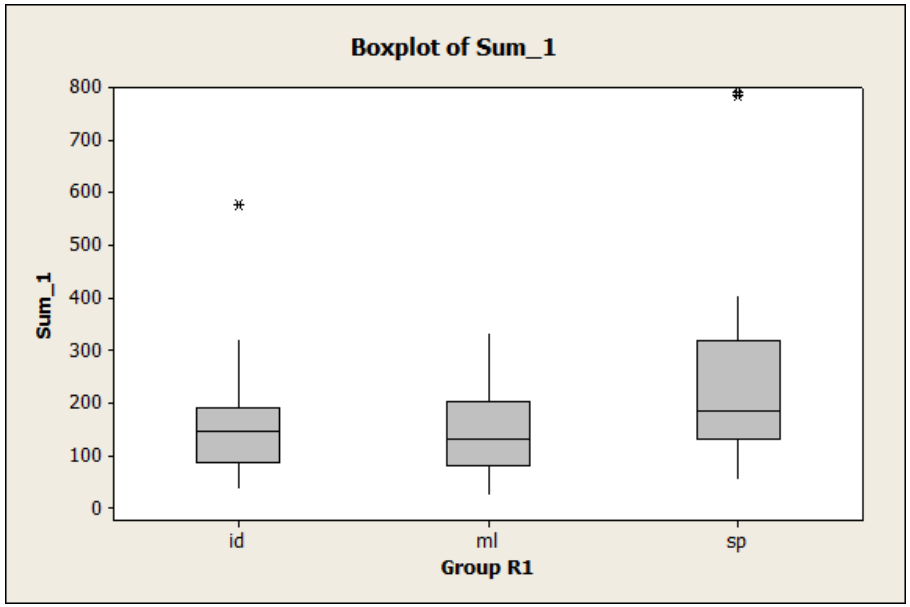
2) De relative forskjellene fortsatt bedre representert ved estimering med fysisk enhe (kilo)t!!!

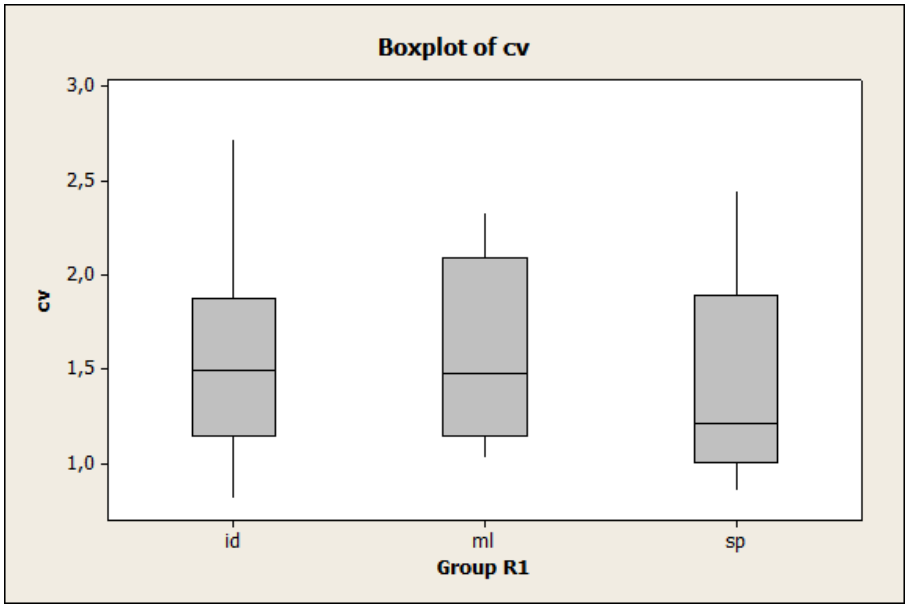
Det er mao så langt liten grunn til å si at vi generelt sett er bedre til å estimere relativt. Men hva med IT-prosjekter.

Timeverk vs Ideelle timer vs Story Points

- Tre releaser (R1, R2 og R3), ca. like vanskelige. Beskrevet som user stories. Det tok ca. 800 timeverk å utvikle hele systemet.
- Tre grupper (polske utviklere)
  - Dag 1: Alle estimerte R1. G1 vha ML, G2 vha Ideelle timer og G3 vha Story Points
  - Dag 2: Alle estimerte R2. G1 vha Ideelle timer, G2 vha Story Points og G3 vha ML
  - Dag 3: Alle estimerte R3. G1 vha Story Points, G2 vha ML og G3 vha Ideelle timer
- Alle som estimerte vha Ideelle timer ble etterpå bedt om å estimere ML. Alle som estimerte vha Story Points ble bedt om å estimere referanseoppgaven i timeverk.
  - Ideelle timer: "*Assume that you are able to work without interruptions, that there are no unexpected events and you are fully productive all the time.*"
- Forskningsspørsmål:
  - Forskjeller i variasjon?
  - Forskjeller i høye lave estimater (når omregnet til timeverk)

Resultater





Indikerer at:
- Estimatene ble høyere med Story Points
- Minst diskriminering mellom User Stories (mean/std) med Story Points! (arbeidsmengden for de ulike User Stories varierer altså mindre ved bruk av Story Points)

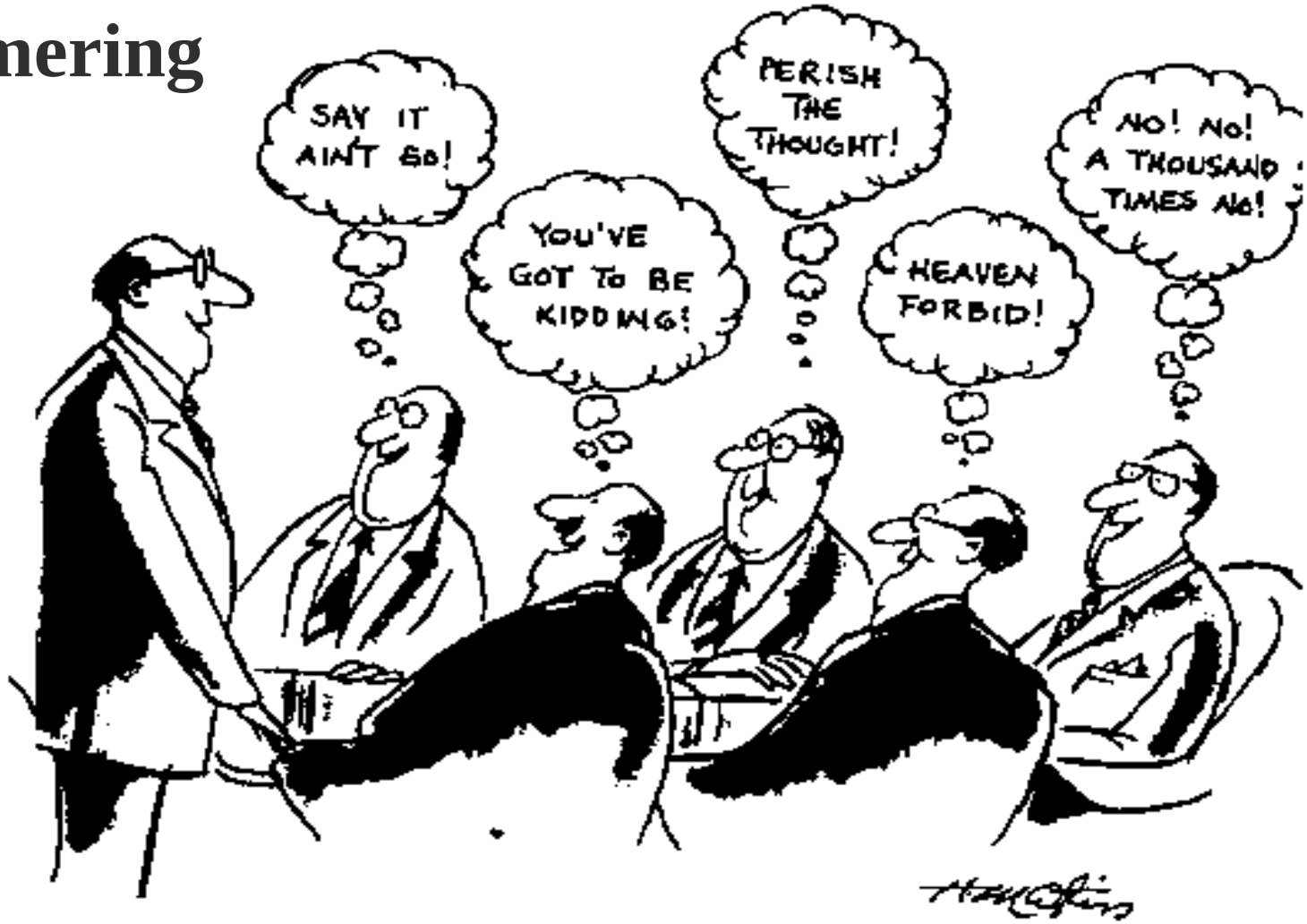
Masteroppgave (Ivar Fredriksen)

- Studie av iterasjoner i et norsk IT-firma.
  - De som gjorde jobben (”interns” under opplæring)
  - ”Eksperter” som kun estimerte
- Noen sprints ble estimert i ideelle timer andre i story points
- Funn:
  - ”Story points” mer ”konsistent” enn ideelle timer (bedre korrelert med faktisk arbeidsmengde)
  - Story points ville trolig gi mer realistiske estimater enn ideelle timer (analyse basert på ulike antagelser, som var vanskelig å verifisere)

Anbefalinger

- Innfør en presisering av hva ”story points” betyr, og baser denne på relative forskjeller i arbeidsmengde (ikke varighet, ikke størrelse, ikke kompleksitet, ..).
- Story points har trolig en stor fordel ved at den lette bruk av historiske data. Dette kan i seg selv være god nok grunn til å bruke den.
- Husk å reestimere (eller bruke andre metoder) så snart antagelsene story point bygger på ikke gjelder, f eks dersom arbeidsmengden til en gruppe user stories oppdages å være mye større i forhold til andre user stories enn først antatt eller det meste av arbeidet består i å implementere kompliserte algoritmer, verifisere løsninger, refaktorere etc..
- Bruk strukturerte metoder for å kombinere uavhengige estimater (f eks Planning Poker)

Gruppe-estimering



"All those in favor say 'Aye.'"
"Aye."
"Aye."
"Aye."
"Aye."
"Aye."

Planning Poker – enkelt fortalt

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

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

Planning poker – noen forskningsfunn

- Skaper gode diskusjoner og informasjonsutveksling
- God måte å kombinere estimater på.
  - Mer selvstendig tenkning enn ved ustrukturert gruppe-estimering
  - Flere hoder husker mer enn ett
  - Kombinasjon av estimater virker
- Kan over tid skape en større grad av forsiktighet i estimatene og mer kvalitetsfokus, med den bi-effekt at estimatene (og kostnadene) går opp.
- Trenger tilleggsprosesser for å:
  - Estimere uventede (ikke-planlagte) hendelser
  - Unngå å bli påvirket av villedende informasjon

Estimering av aktiviteter
(f eks user stories innen en iterasjon)

Estimering på iterasjonsnivå (aktivitetsestimering)

- Over og under-estimering
  - Små oppgaver underestimeres
  - Middels og store oppgaver over-estimeres
- Ikke-lineær tidsskala
- Formatet kan være viktig
  - Mer optimisme av å spørre: ”*Hvor mye rekker du i løpet av en dag?*” enn å spørre ”Hvor lang tid trenger du på å lage ….?”
- Best å estimere eget arbeid
  - En svært erfaren utvikler bør sjelden estimere arbeidet til en uerfaren.

Estimering på iterasjonsnivå

- Se tilbake på tidligere oppgaver
  - ”Simulering” av hva du skal gjøre steg-for-steg fører typisk til over-optimisme, men kombiner gjerne metoder.
- Registrer timeforbruk og analyser hvor og hvorfor du bommet
  - Pomodoro-teknikken kan være nyttig som utgangspunkt.
- Bruk sjekklister og maler basert på historiske data (f eks hvor mye tid som går med til testing) for å sikre at ting huskes på

De aller viktigste poengene …

- Det er tre ting som må stemme for å få realistiske estimater:
  - Situasjonen må muliggjøre et fokus på realisme
    - **Viktigst**: Ingen estimeringsanker eller miks av tilbud og estimeringsarbeid må forekomme.
  - Estimeringsprosessen må være hensiktsmessig
    - **Viktigst**: Prosessen må muliggjøre riktig bruk av historiske data og god prosess for å estimere ”det ukjente”.
  - Estimeringskompetansen må være tilstrekkelig god.
    - **Viktigst**: Ingenting slår tidligere erfaring med lignende prosjekter eller relevante historiske data
- Så enkelt, men likevel så vanskelig!

Ti faktorer som signaliserer fare for overoptimisme

1. Høy usikkerhet
2. Lite relevant erfaring
3. Høy grad av prosjektengasjement (commitment) 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. Estimater utarbeidet som del av planlegging

Elleve råd for å redusere overoptimisme

1. Utdann en ”cost engineer” som evalueres mhp realisme i estimater og ikke selv er del av prosjektene han/hun estimerer.
2. Skill tydelig mellom prosesser for estimering, planlegging og prising.
3. Unngå irrelevant informasjon som leder til for lave ”ankere” (f eks kundeforventninger)
4. Bruk 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. Dersom historiske data ikke finnes, forsøk å finne eksperter med relevant bakgrunn (også utenfor organisasjonen) og historiske data.
9. Ikke la ekspertutviklere estimere arbeidsmengde til uerfarne (bruk middels dyktige til å estimere middels og under middels dyktige).
10. 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.
11. Kombiner estimater fra uavhengige kilder. Sørg for at disse har ulik bakgrunn og roller. Bruk en Delphi-prosess for å kombinere vurderingsbaserte estimater.

Utvikling av programvare i lavkostland

Magne Jørgensen
Simula Research Laboratory

Min bakgrunn for å mene noe …

Egen erfaring:

- i prosjekter både på kundesiden og utviklersiden.
- Ansvarlig (på kundesiden) for flere mindre prosjekter mot utviklingsfirma i India, Russland og Polen.

Egen forskning:

- Besøkt og intervjuet 14 utviklingsfirma i India, Nepal, Vietnam, Polen, Romania og Ukraina om forhold rundt offshoring.
- Gjennomført 3 omfattende studier på utviklere/firma i offshoring-land.

Kunnskap om andres forskning og erfaringer:

- Oppsummert relevante forskning innen området.
- Oppsummert forskning på kulturelle forskjeller med relevans for offshoring.

Offshoring …

- Offshoring oppfattes her som outsourcing til leverandører fra annet land enn ens eget.
- Grunnene til offshoring er stort sett de samme som for outsourcing generelt, trolig med en ekstra stor vektlegging av kostnadskomponenten.
  - Det finnes land med offshoringsfirma som tilbyr timepriser på 10 USD og lavere for programmerere med solid utdannelse i ”computer science”.
- Utfordringene ved offshoring er også mye de samme som for outsourcing generelt, men med ekstra utfordringer relatert til:
  - Kulturelle forskjeller
  - Tidsforskjeller
  - Språk
  - Geografisk avstand
  - Kunnskap om land, utdannelse, politisk system, etc.
  - Politisk stabilitet, korrupsjon, ….

Noen resultater fra forskningen

- ”Outsourcing: A Game for Losers” (Paul Strassmann, 1995, 2004).
  - ”*Strategy isn’t driving outsourcing. Statistics show the real reason companies outsource is simple: They’re in financial trouble.*”
  - Flere studier tyder på at firma med finansielle problemer (lavere fortjeneste, høyere kostnader) er de som hyppigst tyr til outsourcing. Dette gjelder også (i enda større grad?) for offshoring.
  - Egne problemer ikke alltid et optimalt utgangspunkt for offshoring og gir en høyere grad av fiaskoer i samarbeidet, mao offshoring får i mange tilfelle uforskyldt skylden for fiasko.

Noen resultater fra forskningen

- Det synes imidlertid å være nokså høy risiko forbundet ved å outsource, særlig dersom man outsourcer ”for mye”.
  - De som outsourcet **mindre** enn 80% av IT-budsjettet beskrev outsourcingen som suksessfylt i 85% av tilfellene. De som outsourcet **mer** enn 80%, kun i 29% av tilfellene! (Lacity and Willcocks, 1998). Flere andre studier som finner lignende forhold.
  - En viktig årsak til dette synes å være at for stor andel outsourcing fører til dårligere evne til å være ”den gode kunde”. Dette gjelder i enda større grad for offshoring, der det er enda vanskeligere å være ”den gode kunde”.

Noen resultater fra forskningen

- Det finnes svært mange studier på ”suksessfaktorer”. Suksessfaktorer som hyppigst trekkes fram er (ikke overraskende):
  - Forankring i toppledelsen
  - Evne til å velge/evaluere leverandør
  - Detaljeringsgrad i kontrakter (jo mer detaljert, jo bedre)
  - Kontrakttype (flere studier er – overraskende for meg - i favør av fixed-price, selv om denne modellen også har sine klare svakheter for IT-prosjekter)
  - Varighet til kontrakten (kortere gir høyere grad av suksess, men det gir også større kontrakter, mao det som er mest risiko er små kontrakter med lang varighet).
  - Relasjonshåndtering (tillit, normer, åpenhet i kommunikasjon, samarbeid, felles prosesser, kunnskap, …)
  - Suksess i tidligere samarbeid (”suksess avler suksess”)

Noen resultater fra forskningen

- Forskningsresultater spesifikt for offshoring:
  - Studier dokumenter en betraktelig økning i risiko i forhold til annen outsourcing som følge av følgende faktorer (basert på gjennomgang av 20 studier beskrevet i Lacity et al, 2009):
    - Tidssoner
    - Mindre kontroll (behov for økt kontroll)
    - Økte problemer med kunnskapsoverføring
    - Større kulturelle forskjeller
    - Større behov for å definere leveransene med større detaljeringsgrad
    - Mer geografisk spredte team
  - Erfaringer (se Johann Rost, *Insider's guide to outsourcing risks and rewards*) tilsier at en kostnadsreduksjon på 80% ved offshoring sjelden gir mer enn 15-30% lavere kostnader pga økte "transaksjonskostnader". Ved lite volum, vil transaksjonskostnadene lett bli høyere enn innsparingene i lønnskostnader.
    - **Kommentar**: Man har ofte valget mellom lavere kvalitet og økte transaksjonskostnader.

Egne studier: Anbudsprosessen + valg av leverandør

- Se: M. Jørgensen. How to Avoid Selecting Providers with Bids Based on Over-Optimistic Cost Estimates, IEEE Software (May/June) 26(3):79-84, 2009.
- Noen funn:
  - Invitasjon av mange tilbydere øker sterkt sannsynligheten for å velge en leverandør som har et svært overoptimistisk anbud. Winner's curse blir lett "client's curse".
  - Mindre kompetanse hos leverandør gir lavere anbud og dermed større sannsynlighet for å bli valgt dersom pris er viktig kriterium og kunde ikke er god til å evaluere kompetanse. (Adverse selection-effekten).
  - Kunde bør optimalt sett kunne ha gjort oppgavene selv (dvs bør ikke outsource alt) for å være i stand til å følge opp hensiktsmessig.
  - Kunde bør være mye mer kravstor mhp HVEM hos en leverandør som settes på oppgaven. Mer bruk av kompetansetester. Pareto-prinsippet gjelder for utviklere (20% av utviklere gjør 80% av jobben). Dårlige utviklere i et kvalitetsfirma kan være katastrofale for ditt prosjekt.
  - Ofte dårlig insitamentsordninger ("moral hazard"-effekten). Disse belønner lav kvalitet i utviklingsfasen, som da gir innsparinger SAMT økte inntekter i vedlikeholdsfasen.

Egne studier: Er det kulturelle forskjeller i estimeringen?

- Gjennomsnittlig overskridelser i IT-prosjekter synes å være ca. 30% over hele verden (og har vært på det nivået over lang tid). Studiene måler ikke kvalitet på leveransene, dvs hva som er levert med i gjennomsnitt 30% overskridelser. Her kan det være forskjeller.
- Noen studier (ikke på IT-prosjekter) antyder at det er mer ”over-confidence” i kollektive kulturer mens andre studier ikke finner slike sammenhenger.
- IT-utviklerne har i stor grad den samme typen utdannelse og utfører de samme oppgavene, til dels også for de samme typen kundene. Det er mao en på mange måter homogen gruppe mennesker, med unntak av i hvilken kultur de er oppvokst og lever i.
  - Arbeidsprosesser er dermed nokså like, men det kan likevel være (og er!) store forskjeller på kommunikasjon, maktforhold og forventinger til deg som kunde.

Eget studium (deltagerne)

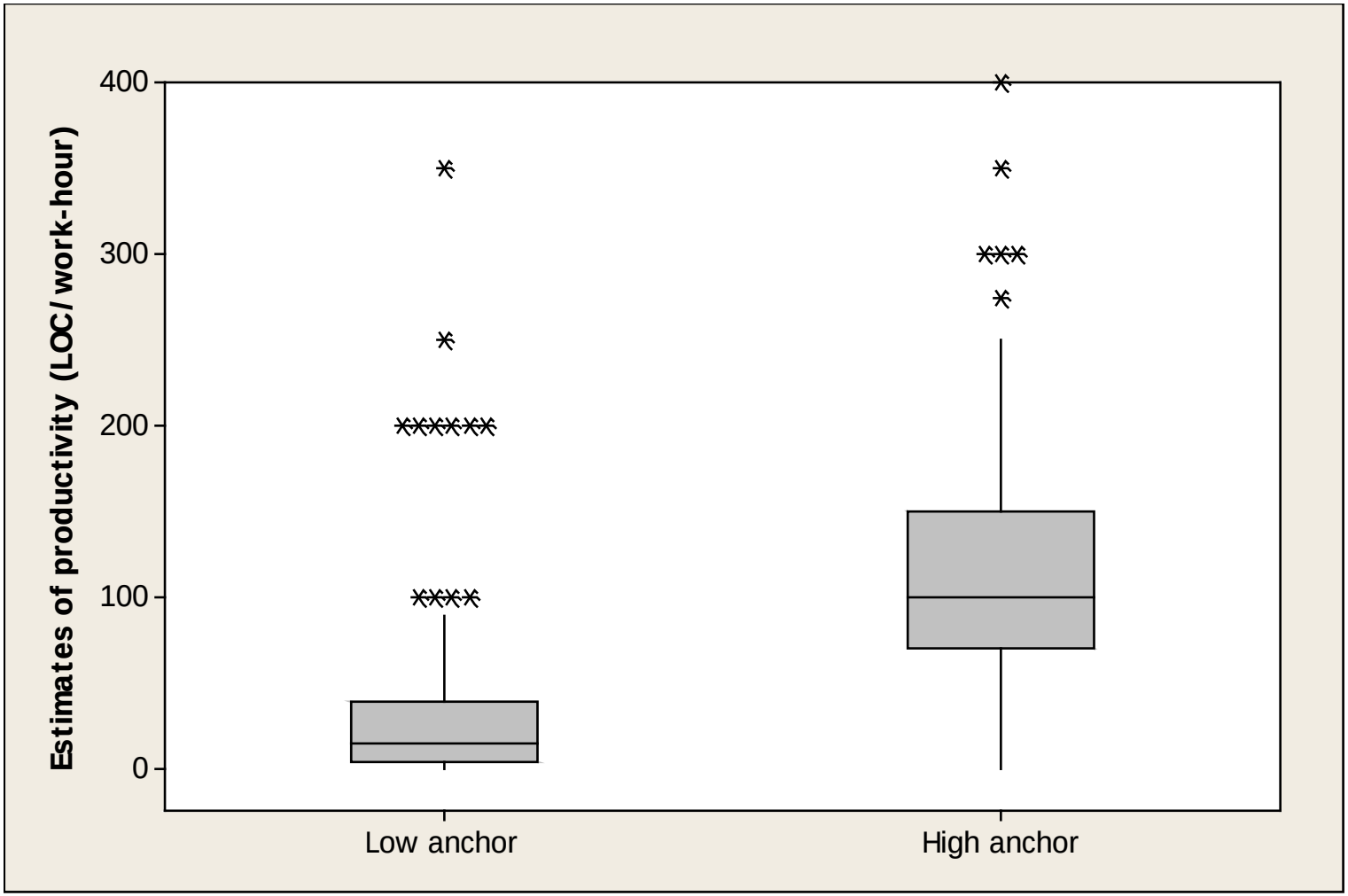
Country	N	Age	Exp. (months)	Dev. Skill (1..5)	Est. skill (1..5)	Engl. Skill(1..5)	Fem-ale	MSc	Manager
India	70	24	20	1.8	2.2	2.4	13%	42%	9%
Nepal	62	27	38	1.9	2.4	1.9	19%	13%	17%
Poland	50	29	56	1.9	3.0	2.4	10%	88%	12%
Romania	49	26	37	2.3	2.8	2.1	14%	45%	10%
Ukraine	78	27	52	2.1	2.9	2.7	10%	61%	21%
Vietnam	65	26	34	2.2	2.8	2.7	15%	11%	18%
Total	373	26	39	2.0	2.7	2.4	14%	41%	15%

Estimeringsoppgave 1

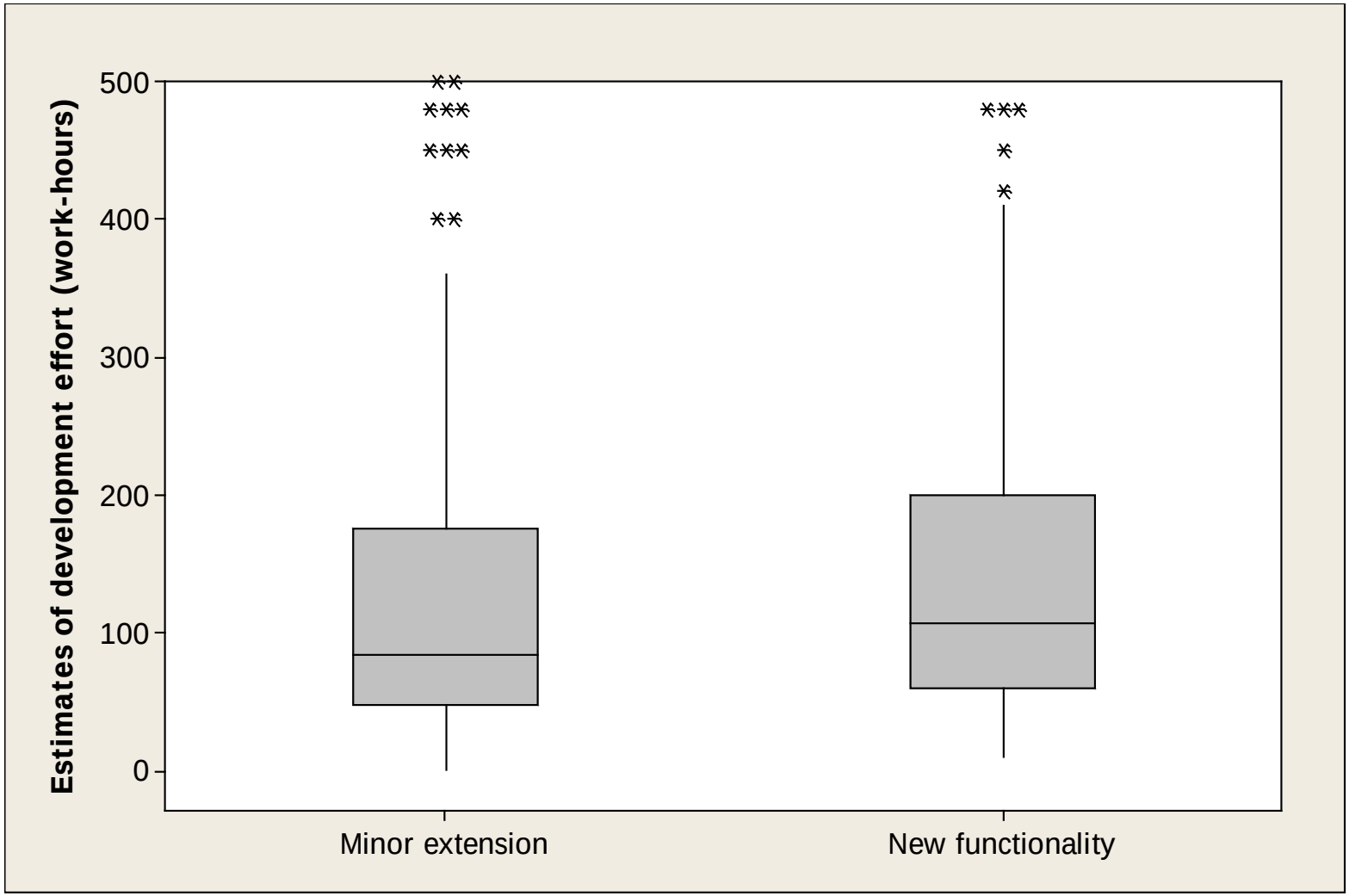
- Low Anchor: Did you on average write more or less than 1 Line of Code per work-hours in your last project?
- High Anchor: Did you on average write more or less than 200 Lines of Code per work-hours in your last project?

- **Estimer antall linjer kode per timeverk i ditt siste prosjekt.**

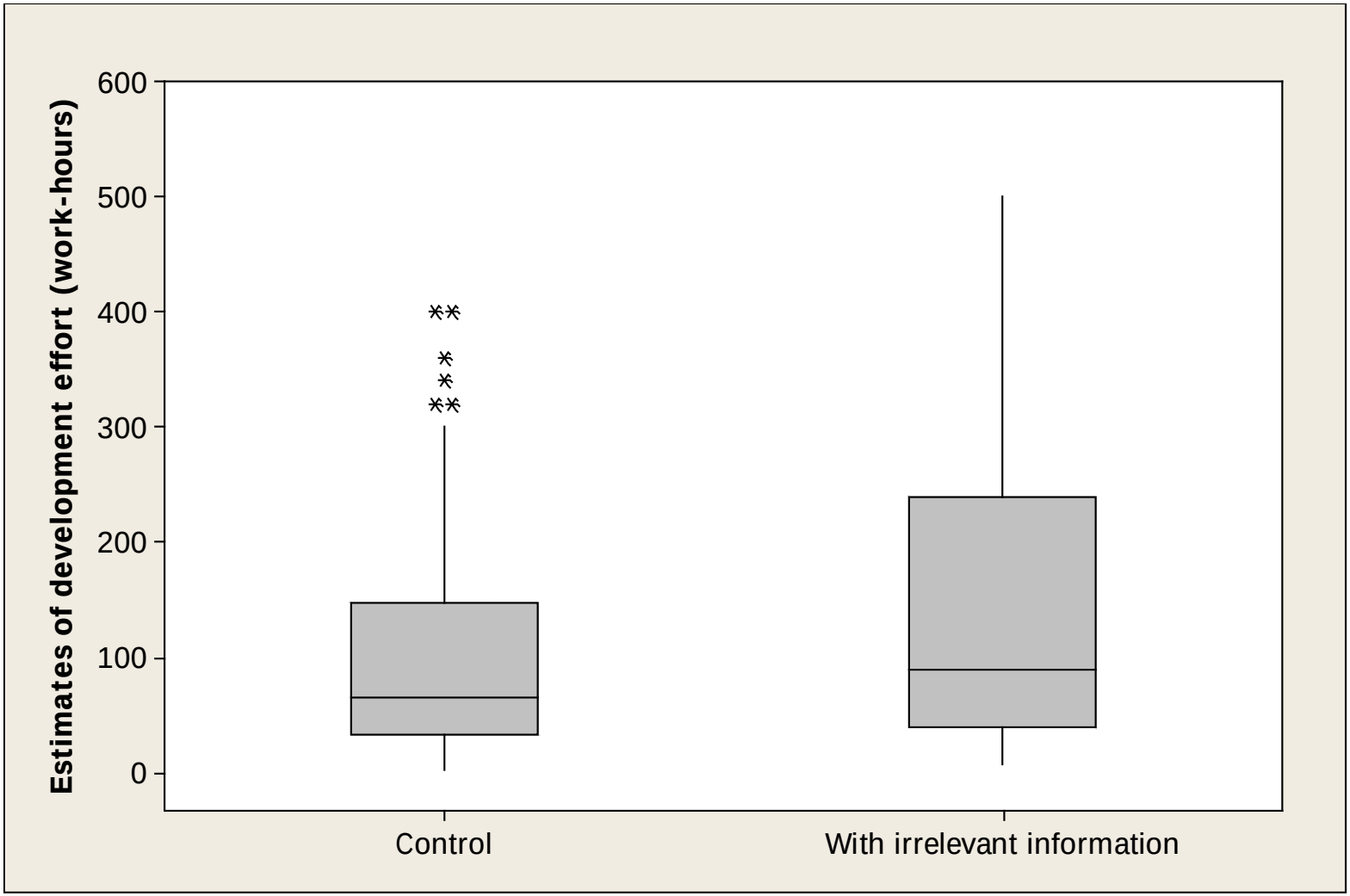
Estimeringsoppgave 1



Estimeringsoppgave 2



Estimeringsoppgave 3



Need-for-cognition, self-construal (independent/interdependent, thinking style (analytic/holistic)

Question (FRA INTERDEPENDENCE SPØRSMÅLENE)	Answer
1. I have respect for the authority figures with whom I interact	1 2 3 4 5 6 7
2. It is important for me to maintain harmony within my group	1 2 3 4 5 6 7
3. My happiness depends on the happiness of those around me	1 2 3 4 5 6 7
4. I would offer my seat in a bus to my boss	1 2 3 4 5 6 7
5. I respect people that are modest about themselves	1 2 3 4 5 6 7
6. I will sacrifice my self-interest for the benefit of the group I am in	1 2 3 4 5 6 7
7. I often have the feeling that my relationship with others are more important that my own accomplishments	1 2 3 4 5 6 7
8. I should take into consideration my parents' advice when making career plans	1 2 3 4 5 6 7
9. It is important to me to respect decisions made by the group	1 2 3 4 5 6 7
10. I will stay in a group if they need me, even when I'm not happy with the group	1 2 3 4 5 6 7
11. If my brother or sister fails, I feel responsible	1 2 3 4 5 6 7
12. Even when I strongly disagree with group members, I avoid an argument.	1 2 3 4 5 6 7

Mean Scores

Country	Interdependence	Holistic/Analytic	Need-for-cognition
India	5.1 (0.7)	4.7 (0.6)	0.7 (0,7)
Nepal	5.0 (0.9)	5.0 (0.6)	1.1 (0.8)
Vietnam	4.5 (0.8)	4.6 (0.6)	0.7 (0.6)
Poland	4.4 (0.7)	4.9 (0.5)	1.5 (0.7)
Romania	4.4 (0.6)	5.0 (0.7)	1.6 (0.9)
Ukraine	4.2 (0.7)	4.8 (0.6)	1.3 (0.9)
All countries	4.6 (0.8)	4.8 (0.6)	1.1 (0.9)
ANOVA	$p<0.001$	$p=0.004$	$p<0.001$

Ingen systematiske forskjeller i påvirkning (men legg merke til de lave estimatene fra India)

	Estimation Task 1			Estimation Task 2			Estimation Task 3		
Group	**Low anch.**	**High anch.**	**Diff**	**"Minor ext."**	**"New func."**	**Diff**	**Control**	**Irr. Inf.**	**Diff**
India	25	150	125***	63	80	17	30	58	28*
Nepal	11	120	109***	50	152	102*	80	90	10
Poland	12	100	88***	102	110	8	80	100	20
Romania	10	70	60***	95	100	5	50	70	20
Ukraine	10	100	90***	120	120	0	60	200	140*
Vietnam	25	100	75***	90	120	30	100	100	0

Interdependence-faktoren synes å være spennende!

Instrument		Estimation Task 1			Estimation Task 2			Estimation Task 3		
		Low anch.	High anch.	Diff.	Min. ext.	New func.	Diff.	Contr.	Irr. inf.	Diff.
Interdependence	High	15	100	85***	70	100	30**	50	100	50**
	Low	15	105	85***	120	120	0	80	80	0
Holistic/ Analytic	High	14	100	86***	80	100	20*	60	90	30*
	Low	20	100	80***	93	120	27	80	90	10
Need-for-cog	High	11	100	89***	80	107	27*	66	98	32*
	Low	20	100	80***	90	110	20	72	80	8*

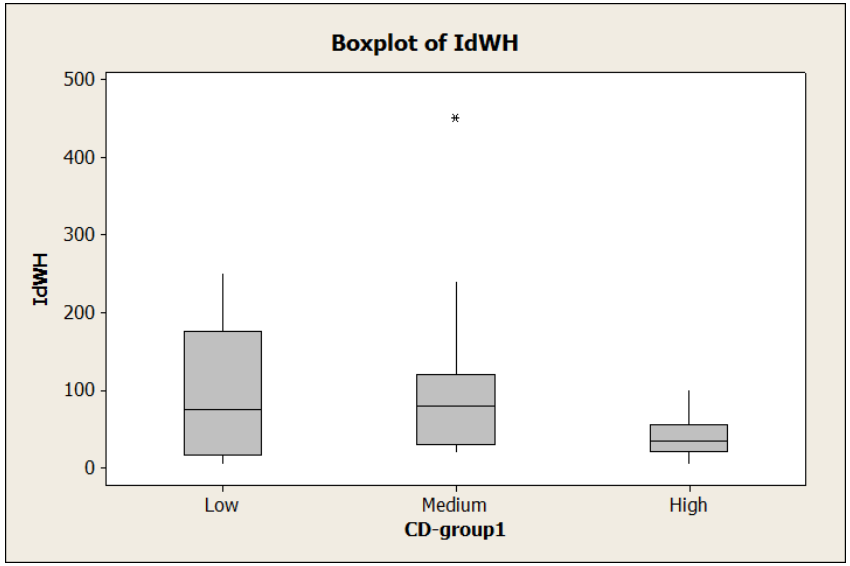
Høy grad av ”interdependence” synes å gi:

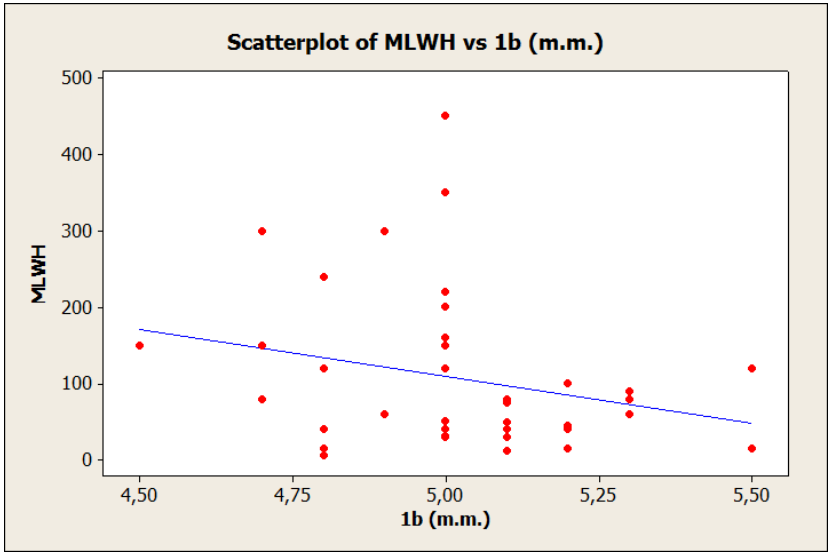
- Lavere (trolig mer optimistiske) estimater. Stemmer også godt med at land med høyere grad av ”interdependence” har lavere estimater.
  - Kan skyldes mer ”følsomhet” overfor konteksten (der lave estimater er ”bra”)
- Større påvirkning av villedende informasjon.
- Har sjekket dette i ett annet studie og finner også lavere estimater med høyere grad av ”interdependence” der, selv om effekten ikke er like tydelig.

Oppfølgingsstudie

Mer kontekstpåvirkning ➔ lavere estimater?

(MEN, hvor sterk er koplingen mellom optiske illusjoner og kulturfenomener?)





Kruskal-Wallis, p=0.04

Litt fra intervjuene med offshoringsfirmaene …

Øst-europa - suksessfaktorer:

- Kunden må ha tilstrekkelig ressurser på sin side, hå forståelse for hva IT-utvikling innebærer, klare krav og at de møtes (fysisk).
- Bør vektlegging laveste pris i mindre grad. Dette skaper et uheldig ”spill” mellom leverandørene, og mellom kunde og leverandør – basert på helt urealistiske anslag på tidsbruk.
- Vi som leverandører trenger tid for å lære om andre kulturer (f eks den franske).
- Store forskjeller mellom US (oppfattes å kunne være ufine og svært krevende), tyske (direkte og krevende), franske (vanskelig kommunikasjon) og japanske (omstendelige, detaljorienterte, men langsiktige) kunder.

Litt fra intervjuene …

Asia - suksessfaktorer:

- Mer direkte kommunikasjon.
- Ønsker seg mer fokus på tidligere prestasjoner til et firma, mindre på pris. Vektlegg å velge firma som behandler de ansatte bra!
- Sjekke bedre at business-modellen er forstått (eksempel med forsikringsordninger som skal implementeres, men ikke finnes f eks i Nepal).
- Oppfatter US som lettere å ha med å gjøre (raskere beslutninger, mer direkte tale) enn Europa. Liker bedre japanske kunder på grunn av svært detaljerte spesifikasjoner.
- Utviklere (særlig fra Vietnam) kan virke litt uprofesjonelle pga manglende engelsk-kunnskap. Mye at man sier ”ja”, uten egentlig å mene mer enn ”jeg hører etter”.

”Straddlers” (kulturelle mellommenn)

- “Embedded knowledge and offshore software development”, by Brian Nicholson and Sundeep Sahay.
  - En historie om et utviklingsfirma i England som dannet et søsterfirma i India, men etter 2 år la det ned igjen.
  - ”*The case emphasizes that outsourcing is not merely about managing the economics, but also developing cultural sensitivity and empathy.*”
    - Uformelt: Ingen god ide å lage et ”little England” i India.
- Foreslår bruk av personer som har ett ben i hver kultur og som kan forhindre misforståelser og stadig dårligere tillit mellom partene.
  - Se også: ”The capabilities of the offshore middlemen”, Mahnke og Wareham, 2007.
- Åpenbar fordel: Mindre misforståelser.
- Åpenbar mangel: Ikke lett å finne/utdanne slike?

Oppsummering

- Offshoring øker risikoen vesentlig i forhold til outsourcing til firma i eget land.
- Direkte kostnadsbesparelser kan være svært høye, men totalbilde er typisk mye mindre gunstig enn reduksjonen i lønnskostnader (f eks fra 80% reduksjon i lønnskostander til 20% besparelser totalt sett + høyere risiko for totalfiasko).
- Offshoring er svært krevende både mhp kompetanse og oppfølging. Det bør settes av MYE tid til personlige møter og ”lære hverandre kjenne” – særlig i starten.
- Ved å velge en leverandør med større kulturell likhet (Øst-Europa) vil trolig risikoen reduseres betydelig mot å velge leverandør i Asia.
- Personer med høyere grad av ”interdependence” gir lavere estimater. Dette var særlig tydelig med indiske utviklere.
- Mellommenn/brobyggere (straddlers) synes å være en god ide, men fortsatt mangler mye erfaringer med dette + at disse kan være vanskelig å få tak i.

Oppsummering

- Det aller viktigste (uavhengig av in-house, nearshore, offshore, …) er imidlertid evnen til å:
  - Velge kompetente leverandører
  - Insistere på kompetente utviklere på ditt prosjekt (og sjekke dette)
  - Være en god kunde
  - Lage incitamentordninger som stimulerer til effektivitet og lav Total Cost of Ownership
- Det finnes gode og dårlige leverandører i alle land!
- Min mistanke er at den største risikoen (og den du kan gjøre mest med!) er relatert til deg som kunde!