

# scienta

Making IT your winning asset.

# Gjennomsnittlig overskridelse av estimater i IT-prosjekter har ligget på ca. 30% siden 70-tallet

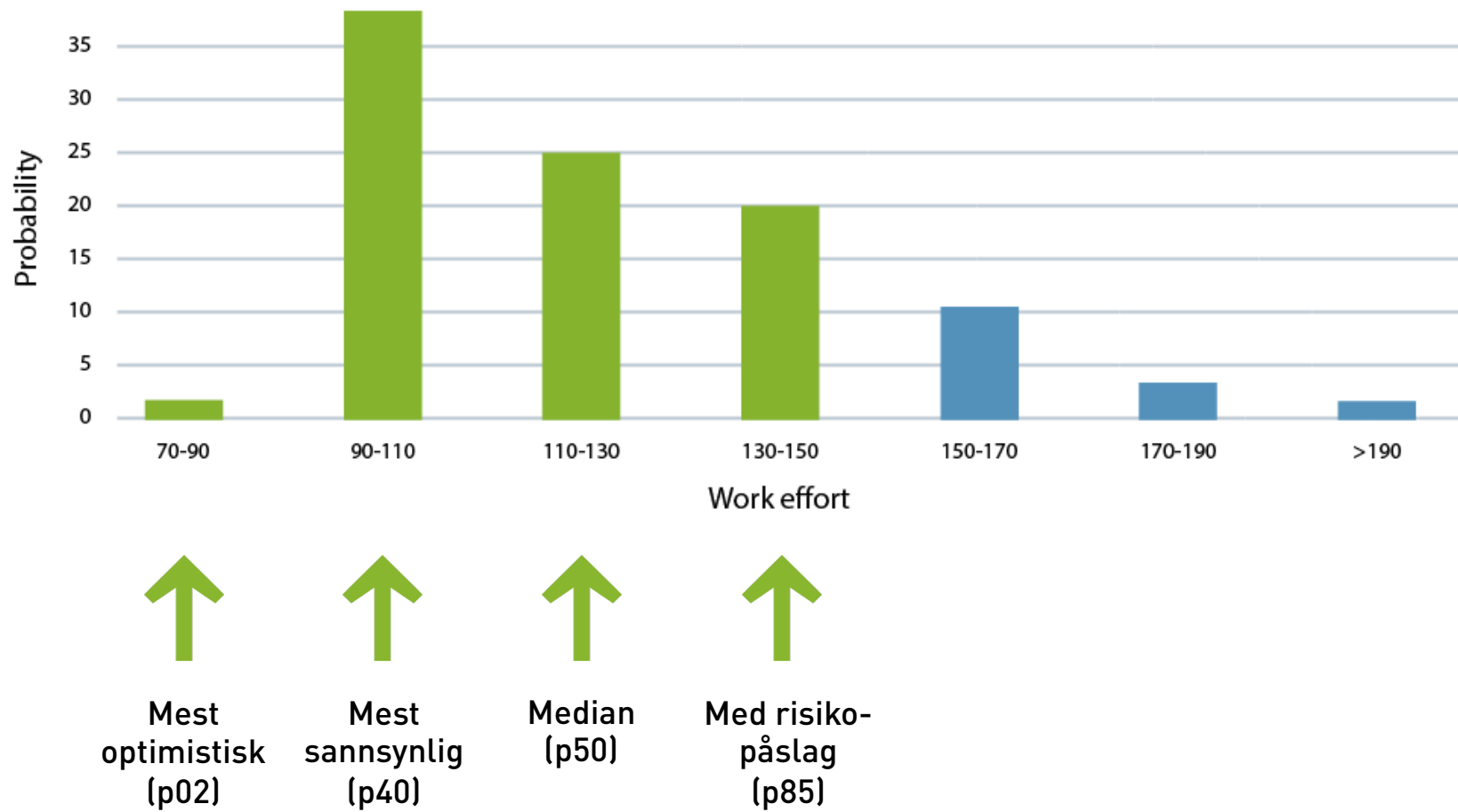
- **Overoptimistiske estimater kan ha alvorlige konsekvenser:**
  - ◆ Software med lav kvalitet
  - ◆ Oppstart av ulønnsomme prosjekter
  - ◆ Dårlige løsninger og lav kundetilfredshet
- **Men også for pessimistiske estimater kan skape problemer**
  - ◆ Gode prosjekter blir nedprioritert pga dårlig BC
  - ◆ Feil løsningskonsept velges
  - ◆ Tapte anbudskonkurranser

# I dette foredraget skal vi se nærmere på fire alvorlige feil som ofte gjøres i estimering

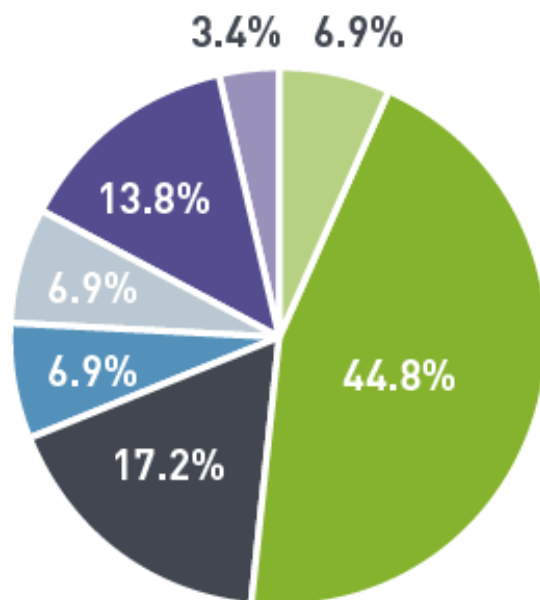
- Uklarhet (hva er egentlig et estimat)
- Overdreven to på egne evner (overoptimisme)
- Manglende selvinnsikt
- Ignoranse (manglende erfaringsdata)



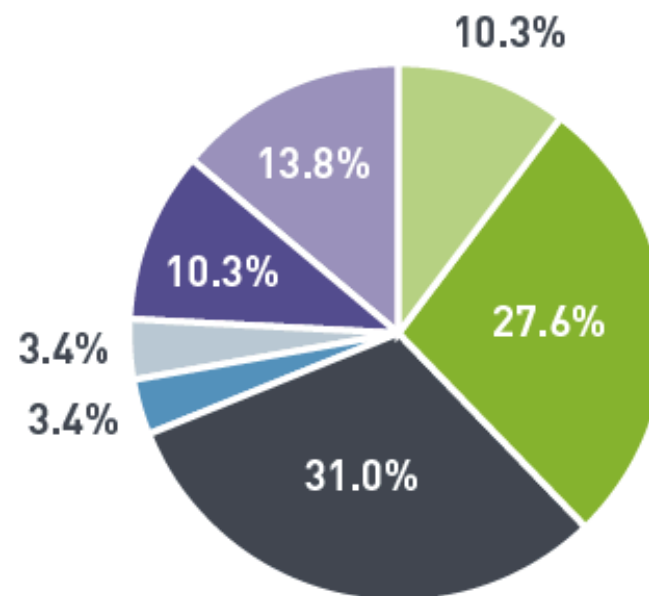
# Tenk probabilistisk



# En studie av 177 utviklere og 53 prosjektledere



Det utviklere mente



Det prosjektlederen oppfattet

- Ingen problemer
- Kun mindre problemer
- Mest sannsynlig
- Median (50%)
- Med risikobuffer
- Ekspertvurd. (intuisjon)
- Annet

# Anbefalinger

- Vær presis i hva du mener (og spør om hva som menes) med estimatet
- **Bruk forskjellig terminologi for forskjellige formål**
  - ◆ Estimert arbeidsmengde (formål: realisme)
  - ◆ Planlagt bruk av arbeidsmengde (formål: prosjektkontroll)
  - ◆ Budsjett (formål: Finansiell kontroll)
  - ◆ Pris (formål: Fortjeneste på kort eller lang sikt)
- Dokumenter hva som menes med estimatene, samt antagelsene de er bygget på

# Overoptimisme

## Overdreven tro på egne evner





# Overvurdering av egne evner er normalt hos mentalt friske mennesker!

- (94% av professorer tror de er bedre enn gjennomsnittet)



# Undervurdering av risiko er også normalt

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



# Forskning om overoptimisme

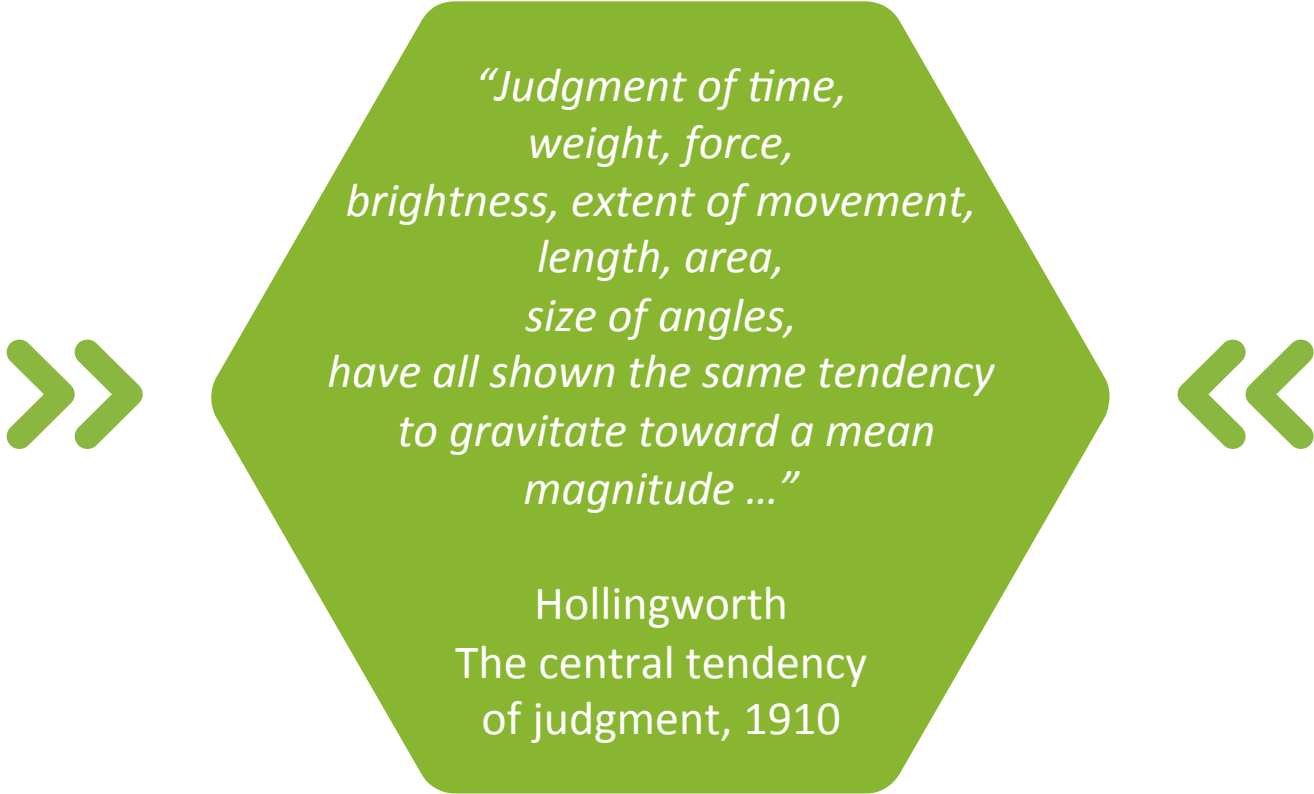
- Vi er mer realistiske når vi estimerer utvikling enn andre prosjektoppgaver (analyse, design, test, ...)
- Erfarne utviklere er dårlige til å estimere når mindre erfarne utviklere skal utføre oppgaven
- Vi er spesielt overoptimistiske når vi har et sterkt ønske om å starte prosjektet ("prosjektforelskelse")
- Mesteparten av tiden går ofte med til å analysere de områdene som er godt forstått fra før
  - ◆ Vi bruker tid på å justere "millimetrene", mens vi ignorerer "kilometerne"

# Anbefalinger

- La utviklere estimere mest sannsynlig arbeidsmengde for utvikling og enhetstest
- Gjør erfaringsbaserte påslag for øvrige prosjektaktiviteter (også for usikkerhetsintervaller)
- Bruk mest tid på å analysere de områdene som er dårligst forstått
- Unngå kompliserte estimeringsmodeller
- De som sterkt ønsker at prosjektet skal koste lite bør IKKE estimere ("prosjektforelskelse")
  - ◆ Det samme gjelder de som kjenner til kundens prisforventninger

Manglende selvinnsikt

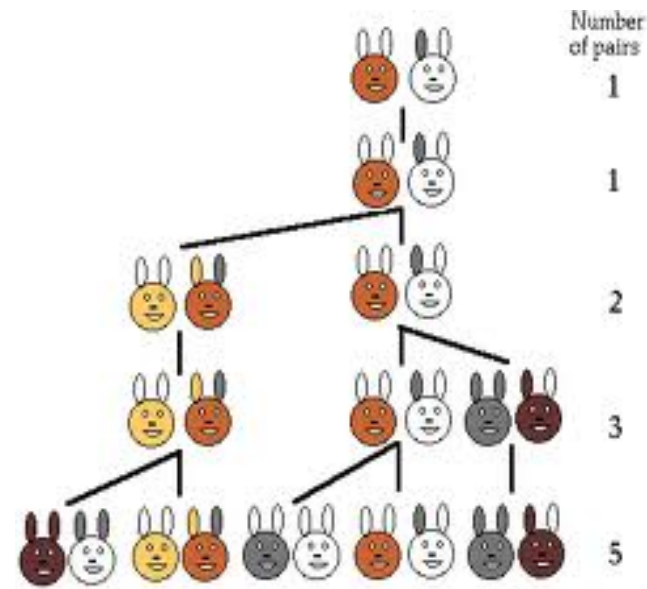
# JO MER VI IKKE VET, JO MER TREKKES VI MOT MIDTEN




*“Judgment of time,  
weight, force,  
brightness, extent of movement,  
length, area,  
size of angles,  
have all shown the same tendency  
to gravitate toward a mean  
magnitude ...”*

Hollingworth  
The central tendency  
of judgment, 1910

# Fibonacci er ikke helt ufarlig for Planning Poker



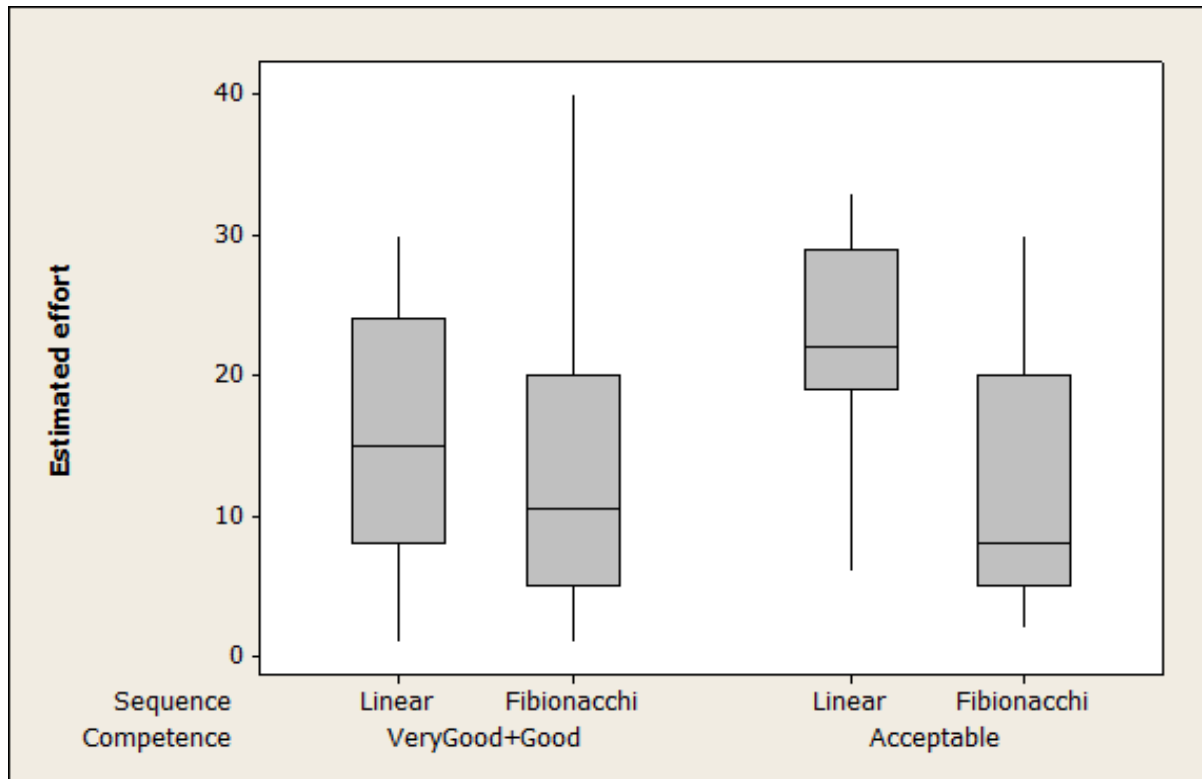
# Eksempel på Fibonacci-type planning poker kort

1	2	3	5	8
13	20	BIG split	?	

# 104 studenter ved Univ. i Oslo estimerte samme oppg.

Lineær-gruppe: 1, 2, 3, ..., 40, mer enn 40 timer

Fibonacci-gruppe: 1, 2, 3, 5, 8, 13, 20, 30, 40, mer enn 40 timer

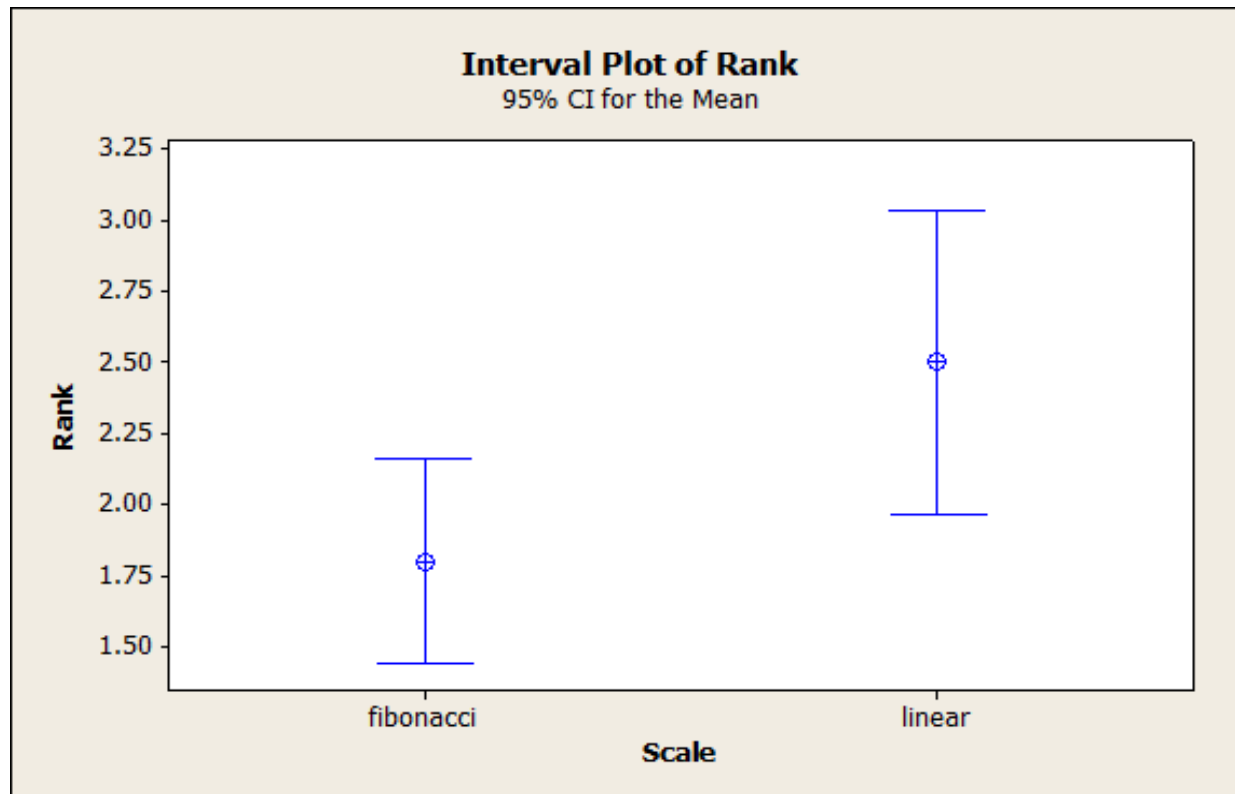




# 16 utviklere, planning poker, reelle oppgaver, 10 oppgaver

Lineær-gruppe: 1,2,3,4,5,... 100 timer

Fibonacci-gruppe: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100 timer



# Hva så?

- Vit at vurderingene dine trekkes mot det som oppfattes som “midten” med økt mangel på kunnskap.
- Når vi nesten ikke vet noe, vil midt-verdien egentlig gjenspeile “vet ikke”.
- Unngå at det som oppfattes som midten av en tall- eller kategoriskala avviker mye fra faktisk middels arbeidsmengde for f eks en user story.
- Dette er særlig viktig for ikke-lineære skalaer som Fibonacci-skalaen og planning poker.

# Erfaringstall

- Ignoranse



# Forskning om erfaringstall

- Erfaringstall er helt essensielt for gode estimater
- Vi er mindre optimistiske når vi ser bakover (hva har skjedd) enn når vi ser fremover (planlegger)
- Vi overvurderer hvor mye vi har lært av tidligere prosjekter
- Systematisk innsamling og bruk av erfaringstall er sjeldent i IT-organisasjoner (dette i motsetning til andre bransjer som olje og bygg)

# Eksperiment: Hvor gode er vi til å estimere når vi ikke har historiske data tilgjengelig?

- Syv erfarne systemutviklere estimerte hver 60 utviklingsoppgaver (utvikling + enhetstest)
- Seks av oppgavene ble estimering to ganger (med minst en måneds mellomrom)
- Erfaringstall var ikke tilgjengelige, men alle hadde god kjennskap til systemet

Session 1 (April, 2006)	Session 2 (May, 2006)	Session 3 (June, 2006)
T1	T21	T41
T2 (same as T44)	T22	T42
T3	T23 (same as T59)	T43
T4 (same as T46)	T24 (same as T60)	T44 (same as T2)
T5	T25	T45
T6	T26	T46 (same as T4)
T7	T27	T47
T8	T28	T48
T9	T29	T49
T10	T30 (same as T14)	T50
T11	T31	T51
T12	T32	T52
T13	T33	T53
T14 (same as T30)	T34 (same as T16)	T54
T15	T35	T55
T16 (same as T34)	T36	T56
T17	T37	T57
T18	T38	T58
T19	T39	T59 (same as T23)
T20	T40	T60 (same as T24)

# Resultat

- Gjennomsnittlig (mean) forskjell mellom estimer av samme utvikler og av samme oppgave, men estimert på to forskjellige tidspunkt, var så høy som 71% (median = 50%)!
- Oppgavene var “estimeringsvennlige”
  - ◆ tydelige spesifikasjoner
  - ◆ kjent teknologi
  - ◆ små oppgaver
  - ◆ ikke påvist læring mellom estimeringssesjonene

# Anbefalinger

- Ikke stol på at du (eller andre) husker erfaringstall riktig - sjekk skriftlig dokumentasjon!
- Ikke anta at dere vil gjøre det mye bedre enn tidligere prosjekter
- Ikke vær redd for å innhente eksterne erfaringstall, men vurder nøye hvor relevante de er i din kontekst
- Utpek en “cost engineer” som har ansvar for innsamling, analyse og distribusjon av erfaringstall på tvers av prosjekter



# Oppsummering

- **Vær klar i din tale**
  - ◆ Klargjør hva du mener med et estimat.
  - ◆ Dokumenter antagelser og bruk av erfaringstall
  - ◆ Ikke motta estimer uten å sikre deg at du har forstått hva som er ment med dem og hvordan de er utarbeidet
- **Vær rasjonell med hensyn til din irrasjonalitet**
  - ◆ Ta hensyn til at middel-kategorier betyr mye når usikkerheten øker
  - ◆ Vær oppmerksom på at du er spesielt utsatt for overoptimisme når du sterkt ønsker at prosjektet skal startes
- **Bruk tid på å samle, analysere og distribuere erfaringstall**

# Flere dødssynder (Smidig 2014!)

- **"The client's curse"**
  - ◆ Ikke velg en leverandør som er mye billigere enn alle andre
  - ◆ ... med mindre det finnes en god begrunnelse for lavere pris
- **Ankereffekter og irrelevant informasjon**
  - ◆ Det er ekstremt lett å manipulere estimater med irrelevant og misvisende informasjon
  - ◆ Har man vært eksponert for irrelevant informasjon er skaden allerede skjedd
- **Dårlig prosjektstyring**
  - ◆ Unngå "big bang"-prosjekter (iterative prosesser virker!)
  - ◆ Ta høyde for risiko i prosjektplanleggingen (ikke la kundens prioritering av forretningsverdi være eneste input til planlegging)

# scienta

Magne Jørgensen (magnej@simula.no)  
Stein Grimstad (stein@scienta.no)