Jeg vet ikke hva et ”story point” er, men det virker bra …

Magne Jørgensen
Simula Research Laboratory

Innhold

- Hva ligger bak (og bør ligge bak) konseptet ”story point”?
- Noen resultater fra studier på relativ estimering, story points, ideelle timer og timeverk
- Anbefalinger

Vet du hva et story point er?

- Dersom User Story A har dobbelt så mange story points som User Story B, så har User Story A:
  1. Dobbelt så stor størrelse
  2. Dobbelt så stor kombinert størrelse og kompleksitet
  3. Dobbelt så stor kombinert størrelse, kompleksitet og risiko
  4. Dobbelt så stor kombinert arbeidsmengde, størrelse, kompleksitet, risiko osv.
  5. Dobbelt så stor arbeidsmengde (timeverk)
  6. Dobbelt så lang varighet (kalendertid)
  7. Ingen av alternativene ovenfor

”Oppklaringer” om 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".

”Oppklaringer” om story points

- 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.”

- **Er det helt opp til teamet å bestemm om story points er:**
  - Relativ størrelse, kompleksitet og/eller risiko
  - Relativ arbeidsmengde
  - Relativ varighet

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.”*

Historien til ”Story Points”

- **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 (effektiv arbeidstid) enn varighet.
  - De innfører ”perfect days” (som i sin opprinnelse ligner svært på effektiv arbeidstid).
  - De ser imidlertid at skille mellom effektiv arbeidsmengde (”perfect day”) og varighet er vanskelig å kommunisere.
  - Derfor ”fjerner” de koblingen til arbeidsmengde ved at de kaller det ”points”. Som senere har blitt til ”story points” (norsk: Kravpoeng?). 1 : 1 - forholdet mellom ”perfect day” og ”story point” har etter hvert blitt oppløst.
  - I etterkant har også ideell tid fjernet seg fra opprinnelig ideell tid (effektiv arbeidsmengde) ved at man har lagt til andre ideelle forutsetninger (”flyt”, ingen problemer, …).

Mitt forsøk på klargjøring av Story Point

- Dersom nøyaktighetshensyn i estimeringen [f eks som respons til h*vilke og hvor mange "user stories" rekker vi i neste release?*] teller mest, bør %-vis forskjell i "story points" være identisk med %-vis forskjell i arbeidsmengde.
  - To ganger som mange story points bør dermed tilsvare to ganger så stor arbeidsmengde.
  - **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 (f eks story points per ukesverk), ikke varighet. Unntak for situasjoner med svært stabilt teaminnsats, der denne forskjellen ikke spiller noen rolle.
  - Estimering relatert til varighet bør ha en tilleggsprosess der man justere for antall arbeidsdager, variasjon i tilgjengelige ressurser, etc..

Mitt forsøk på klargjøring av Story Point

- Klargjøringen gir en relativt uproblematisk forståelse innad i en release, men kan (som de andre ”klargjøringene”) kreve mer kompliserte betraktninger mellom releasene:
  - Det viktige er å holde klart at målet med estimeringen er at arbeidsmengde-relasjonene også skal gjelde mellom releaser (ellers bør man re-estimere).
  - Det er også viktig å ikke la seg forvirre av at konverteringsfaktoren mellom story point og arbeidsmengde (noe forvirrende kalt ”velocity”) vil kunne variere mellom releaser.
- Dersom vi hadde kalt Story Points for ”Work Points” hadde vi kanskje ikke hatt disse forståelsesproblemene. Story points gir lett en følelse av at det er størrelse man måler (ref. function points – som er et størrelsesmål).
- Ikke veldig viktig hva vi kaller det, men viktig at vi er enige hva som er målet (estimeringsnøyaktighet) og hvordan vi oppnår det (samvariasjon med arbeidsmengde).

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. Story points 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 relativt og arbeidsmengde et absolutt estimat**: Arbeidsmengde (f eks timeverk) og story points har de samme “relative” egenskapene (begge er på en ratio-skala). Forskjellen ligger i hvor universell og veldefinert enheten er. Arbeidsmengde-estimering gjøres typisk relativt, i likhet med story points.
  - Det er i det hele tatt vanskelig å forestille seg hvordan man **ikke** skal estimere relativt.
  - En fare med story points er at referanse-rommet ved relativ estimering reduseres i forhold til timeverk-estimering!!!! Dvs, at story-point basert estimering er en redusert variant av relativ estimering.

Noen ”myter” om story points

- **Systemutviklere er bedre i å estimere relativt enn absolutt**:
  - Estimering av IT-prosjekter som ikke er basert på sammenligninger (er relativ) neppe finnes.
  - Dersom det menes at forholdet mellom “estimerte story points” og ”faktiske timeverk” samvarierer bedre enn “estimerte timeverk” og “faktiske timeverk”, så mangler dokumentasjon.
  - Dersom det menes at vi er bedre til å vurdere om A er større enn B, enn hvor mye større A er enn B, så er utsagnet irrelevant (og helt opplagt).
- **Story point gjør at vi ikke trenger å estimere dagsverk/dager**: Kalendere og klokker styrer fortsatt hverdagen. Eneste forskjellen er at vi utsetter konverteringen …

Noen beskrivelser av ”ideelle timer/dager”

- www.targetprocess.com/blog/2004/12/iteration-velocitys-and-user-story.html
  - “I can imagine and estimate how much work I could accomplish within a single ideal day. No interruption, no coffee breaks, great mood and so on. So I estimate User Stories in ideal days.”
- www.extremeplanner.com/blog/2006/11/agile-estimating-how-long-is-ideal-day.html
  - “After doing some reading and thinking, I decided an ideal week was something like a "man-week", adjusting for daily distractions.”
- agilesoftwaredevelopment.com/2006/12/measures-of-size
  - “Ideal days are the imaginable amount of time that would be spent on the project if there were no interruptions (including email checking), everybody was working full time at full speed without any vacations and everything needed was present from the day one.”
- **Lett å forstå, men det er vanskelig å være konsistent mhp hvor “ideelt” man skal tenke. Også her uklart I hvilken grad et relateres til arbeidsmengde (man-week) og varighet (distraction).**

Sammenligning (noe subjektiv)

	Timeverk	Ideelle timer	Story points
Klarhet i konseptet	OK	OK-	Nei
Effektiv å bruke	OK	OK	OK+
Lett å forstå for kunde	OK-	Nei	Nei
Underveislæring	OK-	OK-	OK+
Lett å re-estimere	OK-	OK-	OK+
Nøyaktighet	?	?	?

Er vi bedre til å estimere ”relativt”?

Eksperiment 1:

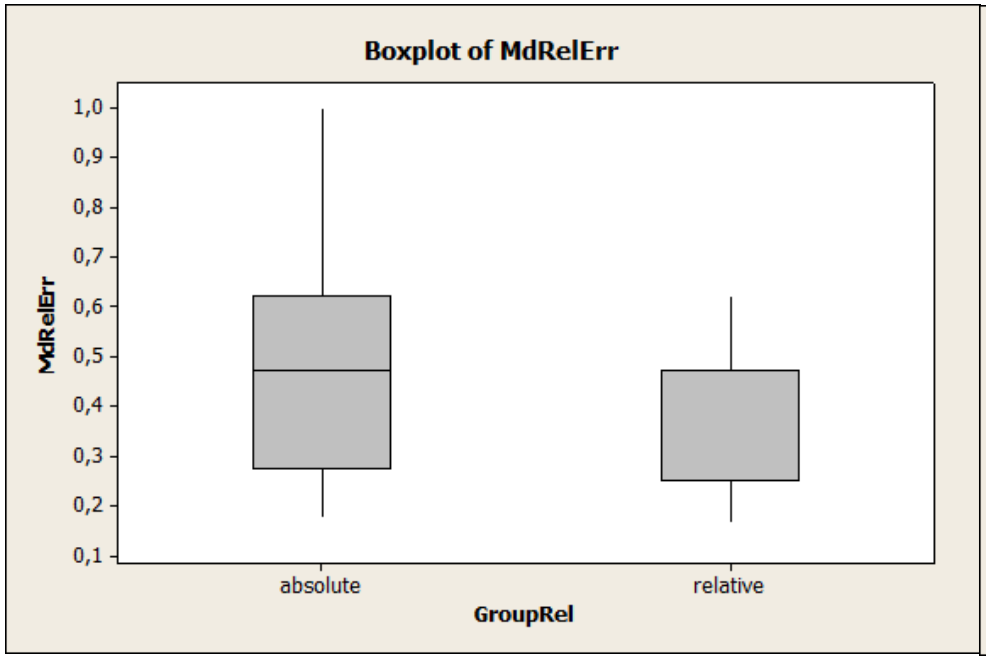
Gruppe 1:

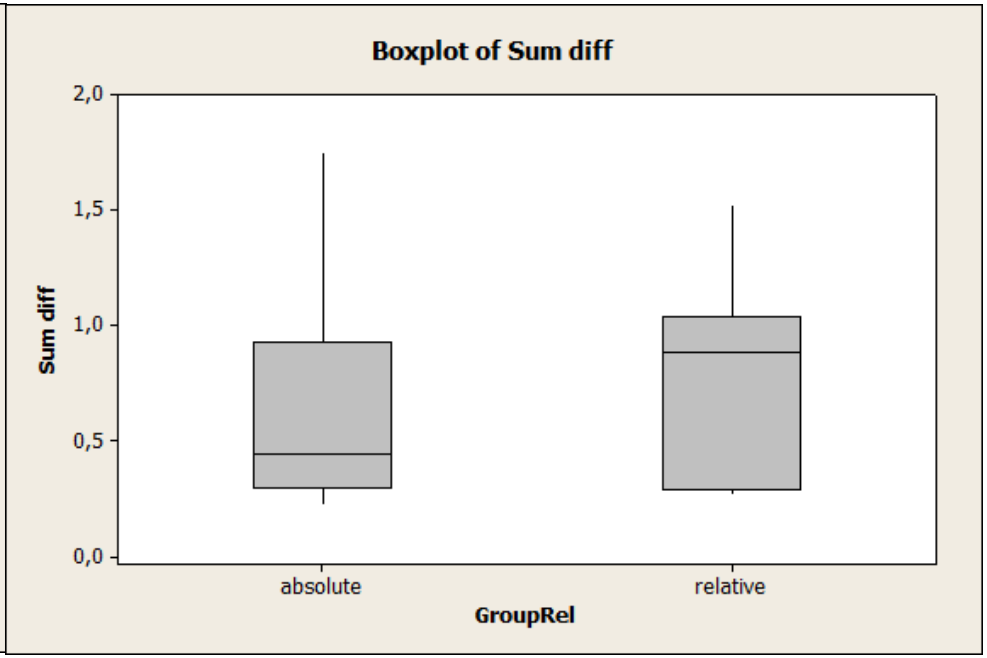
1. Opplæring/eksempler relativ estimering.
2. Rangering av antall innbyggere i fem store byer (Berlin, Warsawa, Napoli, London, og Praha)
3. Velg byen som er rangert som 3. størst og gi den 10 “by-poeng”. Gi de andre byene by-poeng relatert til antall innbyggere.
4. Anslå antall innbyggere for den byen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

Gruppe 2:

1. Direkte estimering av antall innbyggere.

Resultater





Indikerer at:

1) Nøyaktigheten (MdRelErr = median relativ error) ble bedre med relativ estimering.

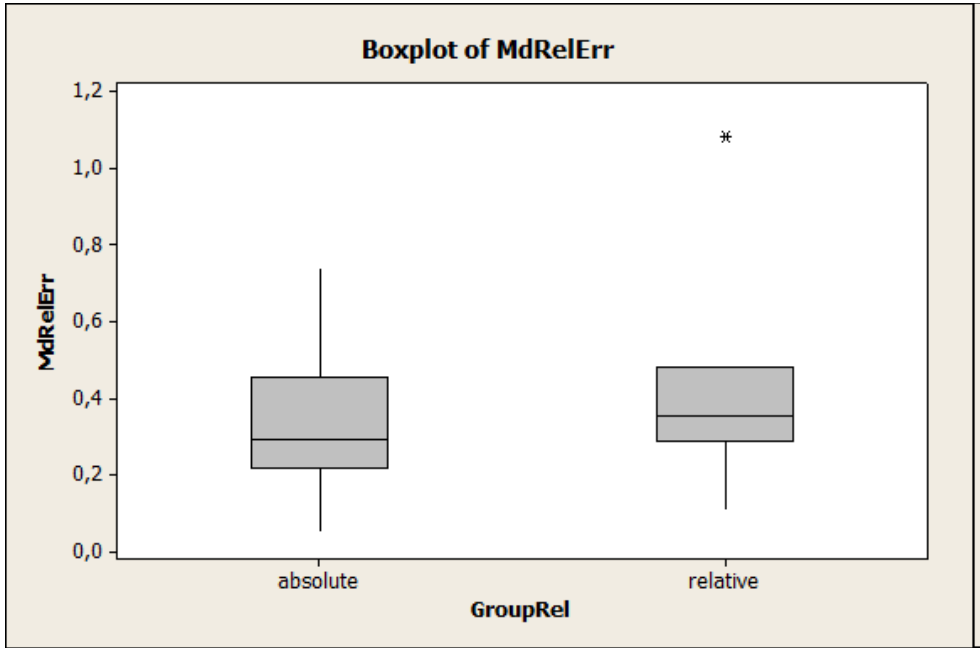
2) **De relative forskjellene var bedre representert ved estimering med fysisk enhet!!!** (Sum diff viser gjennomsnittlig sum av prosentvis differanse mellom estimert og faktisk verdi per estimerer)

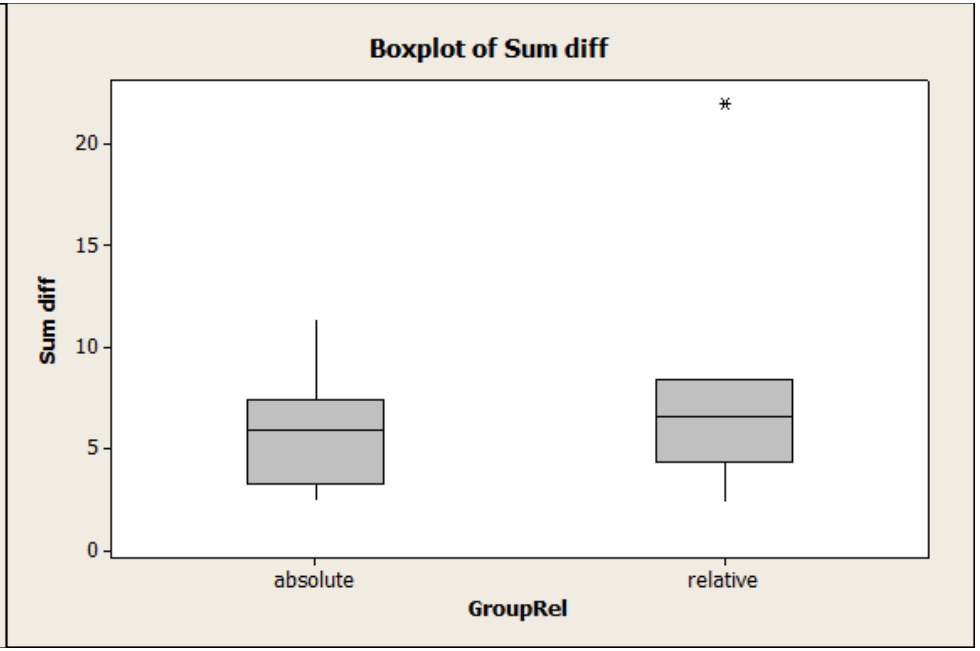
Er vi bedre til å estimere relativt?

Eksperiment 2:

Som eksperiment 1, men med estimering av vekten på fem dyr (løve, isbjørn, jaguar, hyene, tiger)

Resultater





Indikerer at:

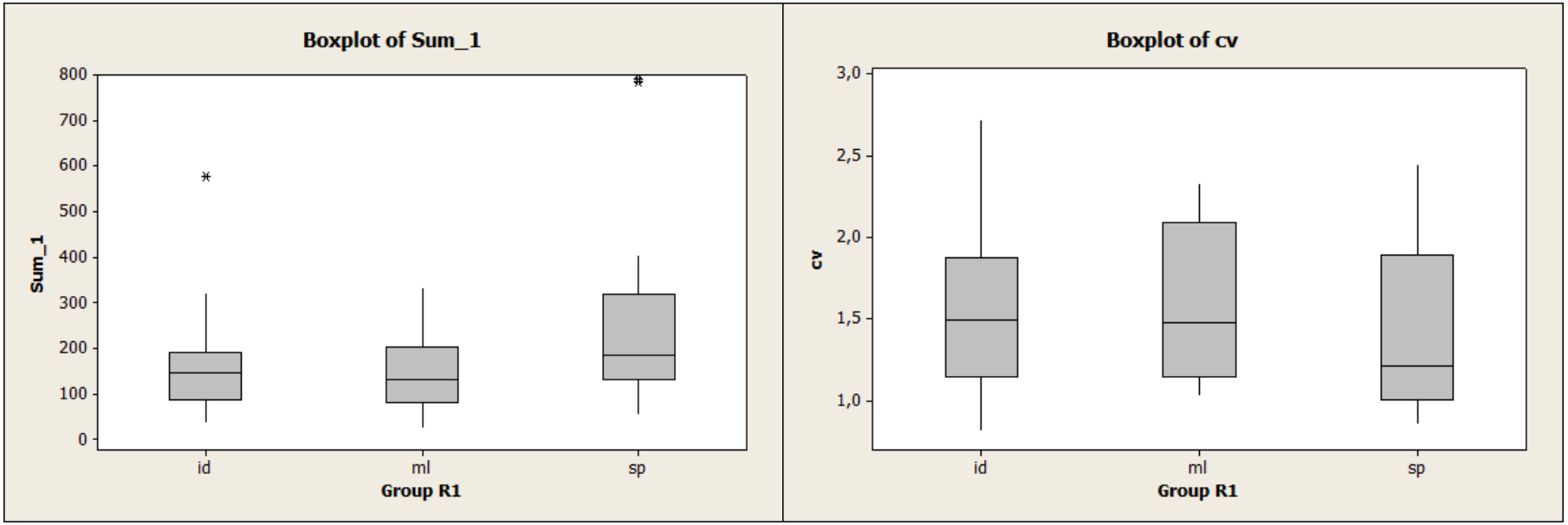
1) Nøyaktigheten ble her dårligere med relativ estimering,
2) De relative forskjellene fortsatt bedre representert ved estimering med fysisk enhet!!!

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

Studie:
Timeverk (ML) 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 G1, G2, G3:
  - 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.*"
- **NB:** Studiet sier dessverre ikke så mye om effekten av story point ved bruk av historisk velocity-data. Sier mest om hvilken effekt denne typen relativ estimering har på "diskriminering" mellom user stories. [Blir User Stories "likere" ved bruk av story ponts?]

Resultater



Indikerer at:

- Estimatene ble høyere med Story Points! (oversatt til timeverk vha at de estimerte referanse-user story også i timeverk.
- Idelle timer → ML fører typisk til 25-30% økning målt mot ML direkte.
- Minst diskriminering mellom User Stories (målt som *mean/std*) med Story Points!

Masteroppgave (Ivar Fredriksen)

- Studie av iterasjoner i et norsk IT-firma.
  - De som gjorde jobben (”interns” under opplæring) estimerte eget teams arbeid.
  - ”Eksperter” som estimerte arbeidet til ”internes”
- Noen sprints ble estimert i ideelle timer andre i story points
- Funn:
  - ”Story points” mer ”konsistent” enn ideelle timer (”konsistens” = bedre korrelasjon med faktisk arbeidsmengde)
  - Story points ville trolig gi mer realistiske estimater av arbeidsmengde enn ideelle timer (MEN, analysen er basert på antagelser som var vanskelig å verifisere)

Hva med ”ideelle timer”, så mest realistisk

- Studier fra andre domener finner at:
  - Prediksjoner under ideelle og ”realistiske” betingelser er svært like
  - Ved å først estimere under ”ideelle” betingelser, blir man mer oppmerksom på forskjelle på ideelle og realistiske betingelser når man skal estimere realistisk.
  - Se for eksempel: Tanner, R. J. and Carlson, K. A. (2009). "Unrealistically Optimistic Consumers: A Selective Hypothesis Testing Account for Optimism in Predictions of Future Behavior." Journal of Consumer Research **35**(5): 810-822.
- Egne studier:
  - JavaZone-eksperiment: Fant en stor effekt. De med ideelle timer først startet på nivå med mest sannsynlig-estimatene, og justerte seg ca. 30% opp.
  - Polen-eksperiment: Fant en stor effekt (30% økning, mye nærmere faktisk tidsforbruk)
  - To andre norske firma: Fant også ca. 30% økning ved å starte med ideelle timer.
- Gjennomsnittlig overskridelse i IT-prosjekter er på ca. 30%. Kan det være en sammenheng med at vi estimerer “ideelt” når vi tror vi er realistiske?

Noen anbefalinger

- Innfør en presisering av hva ”story points” betyr, og baser denne på relative forskjeller i arbeidsmengde
  - Jeg ser ingen god grunn til å beskrive ”story points” som et mål av det som er input (størrelse, kompleksitet, risiko,..) til målestørrelsen. Det er som å beskrive ”hastighet” som et mål på vei og tid, fordi de inngår som input.
  - Det er viktig å ikke blande arbeidsmengde og varighet. Dersom man velger en forståelse av story point som går på varighet, må man i det minste være tro mot denne oppfattelsen, blant annet ved å justere ”velocity” for ulik innsatsmengde.
- Story points har trolig en stor fordel av lettere bruk av historiske data og raskere estimering. Dette kan i seg selv være god nok grunn til å bruke den, f eks for å se hvor mye man rekker i de neste releasene.
- Re-estimer så snart antagelsene ”story point” estimatene 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.
- Et godt alternativ er å først be om ”ideelt antall timeverk”, så om ”mest sannsynlig” arbeidsmengde.