



Hva prosjektledere bør vite om programmerere

Presentasjon NUUG 2007-10-09



Prosjekter går over tid og budsjett

– Gartners forslag til hvorfor det går galt (Etter digi.no):

- Udisiplinert prosjektledelse
- Dårlig fungerende eierstyring og selskapsledelse (*corporate governance*)
- Dårlig støtte fra høyere opp i systemet
- Uerfaren prosjektledelse
- Dårlig forståelse av behov for produktet

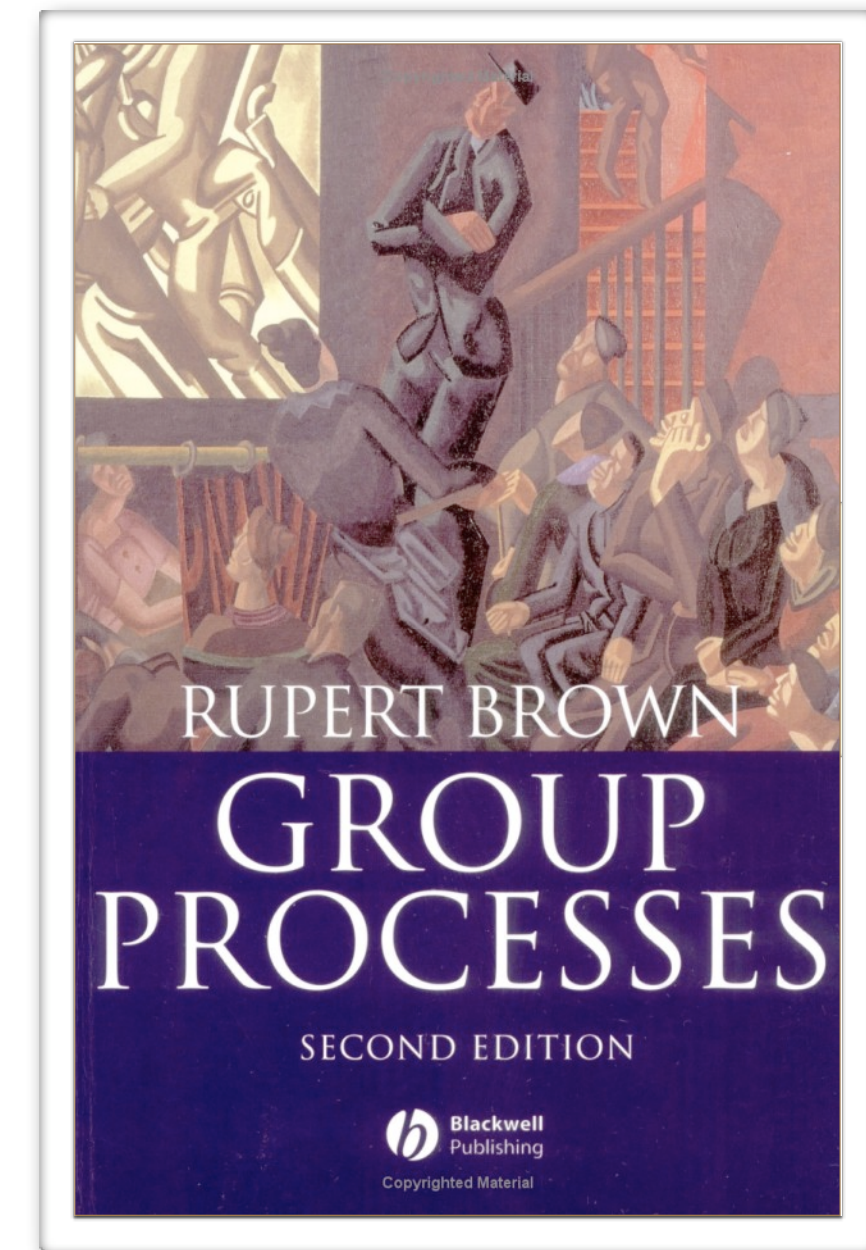
Kunne kanskje sagt mer om hvordan man unngår alt dette?



Empirisk evidens

- PsychINFO-databasen (<http://www.ub.uio.no/>) inneholder oversikt over all publisert psykologisk forskning siden 1875
- Det følgende er bl.a. basert på gjennomgåelse av forskning på programmerer de siste ~20 år
- Men også på studier av prosjekter

Gruppeprosesser



DNV-sertifisert



DNV
testbruker

- Innen alle fag danner deg seg grupper
- Rupert Brown beskriver at gruppetilhørighet kan bestemme ditt syn på andre mennesker selv når kriteriene for gruppetilhørighet er så vage at du ikke vet hvilken gruppe du selv er i
- Grupper innen IT: Driftere mot brukere; programmerere mot brukere; Windows mot Unix; Perl mot Python; C++ mot Java; driftere mot programmerere; driftere, brukere og programmerere mot selgere, etc.

Spesialister på rekruttering





To skoler innen forskningen

- Ingeniører forsker på hvordan programmerere blir mer effektive
 - Men bruker ofte pussige tilnærminger (16PF, MBTI, «søkte» statistiske metoder), slik at resultatene ikke blir sammenlignbare
- Psykologer er mest opptatt av hva som skal til for at programmere skal bli glade i hverandre
 - Til gjengjeld kan de sammenligne hverandres studier

Hva bør prosjektledere vite om programmerere?

- Den forskningen som finnes støtter ingen hypoteser om at det er sammenheng mellom personlighet og effektivitet som programmerer
- Men høy IQ er en fordel; og rent generelt, jo høyere desto bedre
- Det er sammenheng mellom personlighet og ønske om å bli programmerer
- Forskjellige personligheter passer best til konsulent, arkitekt og prosjektleder

Sosial omgang utad

- Den aller siste forskningen sier paradoksalt nok at programmerere ikke skal være for ryddige
 - Muligens fordi programmering fortsatt er så umodent som fag at fleksibilitet er en fordel
- Konsulenter må ha sosial teft, i våre dager skal du ikke bare gjøre oppgavene men også gå i dialog med kunden
 - (Enkelte kaller dette å *sel/ge*)

Sosial omgang innad

- Programmerere vet hvor mye de sjøl har levert
- Men de kan ikke se hvor mye de andre har bidratt
- Programmerere snakker ikke om ting hvis de kan slippe
- Det er risiko for at en eller flere av deltakerne i prosjektet ser på de andre som snyltere uten at dette blir kommunisert eller korrigert

Primadonna

- Programmereren var gjerne best i klassen og vil gjerne fortsatt være det.
- Programmereren vil løse problemet selv, og er overbevist om at det lar seg gjøre helt til det er litt (eller alt) for seint å nå målet
- Prosjektlederen må følge tett opp hvor langt programmereren er kommet.
- Men heller ikke programmerere liker micromanagement.

Hvilket programmeringsspråk er best?

- Det utviklerne kan.
- De dyktigste utviklerne kan flere forskjelligartede språk (F.eks. Prolog, Haskell eller Common Lisp i tillegg til de som «alle» kan).
- Å skifte språk, rammeverk eller utviklingsverktøy midt i prosjektet er å be om bråk.
- Men det vil alltid være diskusjoner.

Prosjektledere og estimering

- Prosjektledere glemmer uhyggelig fort hva som var problemene da de ledet prosjekter.
- Prosjektledere er sjelden hardcore programmerere.
- Jo lenger opp i linja en leder er, desto mindre realistisk bilde har vedkommende av årsaker til overforbruk av tid og penger, selv om lederen selv har ledet softwareutviklingsprosjekter.
- Dette kan føre til at prosjektledere rotter seg sammen mot lederne sine.

Forandring er konstant

- Planen vil endre seg.
- Planen bør endre seg pga. endringer i kunders behov, ikke pga. nye verktøy og nye metoder.
- Det må planlegges for endring.
- Agile metoder er antagelig bl.a. et svar på dette.
- Ikke ta i bruk agile metoder midt i prosjektet. Vent på neste prosjekt.

Kunder er merkelige dyr

- De vet ikke alltid hva de vil ha.
- Da kan ikke utviklerne vite det heller.
- Prosjektlederen må kunne både kundens og utviklernes språk. Da kan de utveksle erfaringer.

Ukultur

- Kunder ber om tilbud fra konsulentfirmaer.
- Begge sider av bordet vet at konsulentfirmets anslag er vel optimistiske.
- Kontrakten går til den som juger best.
- Det er vanskelig å holde disiplin når ingen tror på prosjektplanen.



Eierskap

- Har prosjektet ikke en eier – altså noen som har ansvar for å ta det ferdige produktet videre – renner det i hvert fall ut i sanda.
- De katastrofeprosjektene vi har hørt om har hatt en prosjekteier, dvs. kunden.
- Men eierskapet kan være dårlig definert også hos kunden.

Moral?

- Programmerere må snakke sammen seg imellom
- Prosjektlederen må føre kontinuerlig tidsregnskap med programmererne
- De samme problemene er typiske også i «normale» ingeniørprosjekter, noen avvik skyldes mangel på formelle metoder, men personlighet har antagelig også mye å si