

Produktivitet i IT-utvikling

Hva sier forskningen om forskjeller i produktivitet, hva de kommer av og hvordan man finner de mest produktive?

Magne Jørgensen



Scienta
&
SimulaMet



1

Google Scholar search results for "magne jørgensen 'productivity'".

Articles

About 1,180 results (0.08 sec)

HTML: Characteristics and generative mechanisms of software development productivity distributions
 M. Jørgensen - Information and Software Technology, 2023 - Elsevier
 ... This paper aims to identify properties of software development productivity distributions and gain insight into mechanisms that potentially explain these productivity differences ...
 ☆ Save ☆ Cite Cited by 3 Related articles All 4 versions Import into EndNote

What can we learn from surveys on the importance of software development productivity factors?
 M. Jørgensen - Anais do XXVI Congresso Ibero-Americano em ..., 2023 - sol.sbc.org.br
 ... productivity problems. We argue that it is challenging to interpret survey-based results on the importance of productivity ... previous experiences related to productivity factors and including ...
 ☆ Save ☆ Cite Cited by 25 Related articles All 2 versions Import into EndNote

To read two pages, I need 5 minutes, but give me 5 minutes and I will read four: How to change productivity estimates by inverting the question
 T. Hallgren, M. Jørgensen - Applied Cognitive ..., 2011 - Wiley Online Library
 ... the estimated productivity rate for short tasks to be lower than the estimated productivity for ... The traditional format, time for X work, would result in lower productivity estimates (less work ...
 ☆ Save ☆ Cite Cited by 25 Related articles All 6 versions Web of Science: 11 Import into EndNote

Gains from improved recruitment of software developers
 M. Jørgensen - IEEE Software, 2024 - IEEE Xplore
 ... Abstract— This paper is about how much software developers vary in their productivity and ... of the productivity distribution defends a stronger focus on avoiding selecting low-productivity ...
 ☆ Save ☆ Cite Cited by 25 Related articles All 2 versions Import into EndNote

Better selection of software providers through trialsourcing
 M. Jørgensen - IEEE Software, 2015 - IEEE Xplore
 ... work-samples, which we term trialsourcing in this article, was successfully used ... trialsourcing to detect differences in provider competence. We also discuss different types of trialsourcing ...
 ☆ Save ☆ Cite Cited by 13 Related articles All 6 versions Web of Science: 5 Import into EndNote

Awareness of the consequences of right-skewed effort and productivity distributions in software development
 M. Jørgensen - 2023 IEEE 25th Conference on Business ..., 2023 - IEEE Xplore
 ... At first sight, it may seem contradictory that we both claim a long tail toward higher use of effort (toward low productivity) in Section II and toward high productivity in this section. This ...
 ☆ Save ☆ Cite Cited by 25 Related articles All 2 versions Import into EndNote

KVALITET: En undersøkelse av programmerere fant en betydelig økning i kvaliteten på svarene fra de med høyest alder, skriver Magne Jørgensen.
 Illustrasjon: ISRock.

Femti, feit og ferdig - eller stadig mer produktiv?
 HVA ANDRE MENER: Arbeidsgiver og ansatt påvirker i hvilken grad alder fører til nedgang eller økning i produktivitet, skriver kronikkforfatter.

Magne Jørgensen, Simula Research Laboratory

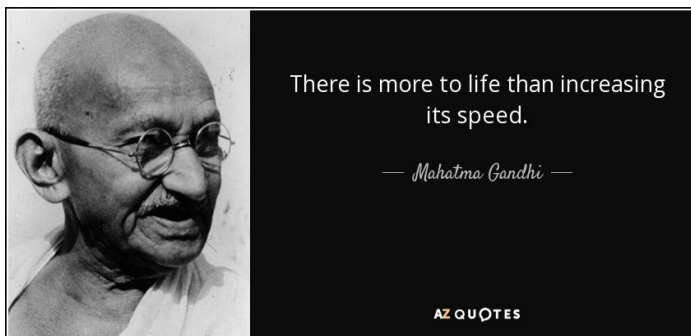
PUBLISERT: Torsdag 26. januar 2017 - 09:44. SIST OPPDATERT: Mandag 30. januar 2017 - 12:12

www.cw.no/hva-andre-mener-karriere/femti-feit-og-ferdig-eller-stadig-mer-produktiv/732981

2

Vanlig innvending. .. som jeg like godt tar med en gang ;-)

Ja, det er mye annet som er mer viktig enn produktivitet, men ...



3

Noen grunner til å ha et forhold til og forsøke å måle/evaluere/forbedre produktivitet i IT-utvikling

- Det åpenbare:
 - Øker vi produktiviteten så tar ting kortere tid og koster mindre
 - Kan vi velge, så vil vi heller ha en utvikler/team/leverandør med høy enn en med lav produktivitet
 - Kunnskap om produktivitet trengs for å kunne estimere kostnader og tid
- Det mer skjulte
 - Det er en tendens til å bli bedre på det man evalueres på. Evalueres man på budsjettkontroll men ikke på produktivitet, så kan det gi lav produktivitet. God budsjettkontroll henger litt for ofte sammen med lav produktivitet (ref. Parkinsons lov)
- Det vanskelige
 - Hvordan evaluere og forbedre produktiviteten uten at det går ut over noe annet?
- **Hva i all verden er egentlig «produktivitet» i IT-utvikling?**

4



5

Kjernen i problemet: Hva mener vi med «output»-delen av produktivitet

- Verdiskaping/gevinster/nytte?
 - Da blir produktivitet det samme som «avkastning på investert kapital» og lønnsomheten til «business case» bestemmer produktivitet.
- Levert funksjonalitet (f eks målt som funksjonspoeng)?
 - Skiller dårlig mellom lav/høy kompleksitet, kvalitetsforskjeller m.m.
- Vi har dessverre ingen gode målestørrelser på «output» av IT-utvikling, og enda mindre noe gode måter å måle alle typer bidrag til output.
- **Hva gjør vi? Vi trenger kunnskap om produktivitet!**

6

To metoder som likevel går an å bruke - i alle fall for forskere - for å måle (relativ) produktivitet

- Gi samme oppgave/prosjekt til flere utviklere/team/organisasjon og se på forskjeller i bruk av arbeidsmengde
- Finn sammenlignbare oppgaver («benchmarks-oppgaver») gjennomført av ulike utviklere/team/organisasjoner

(I noen operative sammenhenger så kan – og har – den siste metoden blitt brukt.)

7

Produktivitetsforskjeller på
team/firma-nivå

8

Seks firma (alle med svært god score fra tidligere kunder og gode cv-er) som alle ga oss tilbud og utviklet det samme systemet – med oss som kunde.

**Better Selection
of Software
Providers
through
Trialsourcing**

Magne Jørgensen, Simula Research Laboratory

Tilbud og estimater ...

IEEE SOFTWARE | PUBLISHED BY THE IEEE COMPUTER SOCIETY

Provider		Mean client satisfaction rating*	No. of rated projects		Estimated price per work-hour (US\$)
A		10.00	6		9.5
B		9.96	96		12.3
C		9.90	124		9.1
D		9.84	32		6.4
E		10.00	4		28.8
F		9.65	44		17.0

* On a scale of 1 to 10, with 10 being the highest.

9

... og her er hvordan det gikk

TABLE 2

Performance characteristics.*

Pro- vider	Actual effort (work-hours)	LOC	No. of ac- ceptance test errors	Cyclomatic complexity per line of code	Estimated extension effort (work-hours)	Code read- ability
A	29	2,600 (6)	20 (4)	0.09	2 (7)	Good
B	84	2,900 (9)	36 (13)	0.09	12 (14)	Medium
C	187	2,800 (3)	25 (7)	0.17	22 (12)	Medium
D	242	5,800 (3)	30 (7)	0.13	17 (7)	Medium
E	527	19,000 (2)	40 (16)	0.14	54 (10)	Very poor
F	474	2,400 (4)	27 (5)	0.16	40 (8)	Poor

* For LOC, the numbers in parentheses show the number of relevant comments measured as the percentage of the lines of executable code. For acceptance test errors, those numbers are the number of major errors. For estimated extension effort, those numbers are the percentage of the total effort.

10

IT-utviklere og testere varierer relativt lite i lønn

Lønnsnivå for utviklere i Norge 2023

Vår erfaring er at bruttolønnen for erfarne utviklere er mellom 700.000 NOK og 900.000 NOK. I grove trekk ser vi følgende lønnsnivåer fordelt på erfaringsnivå:

- Juniorutvikler: 550.000 – 700.000 kr
- Utvikler: 650.000 – 850.000 kr
- Seniorutvikler: 750.000 – 950.000 kr
- Arkitekt/lead: 950.000 kr+

witted.com/blog/software-freelancing/lonnsniva-for-utviklere-i-norge-2023

Hvor mye varierer de i produktivitet mhp å løse programmeringsoppgaver?



11

Journals & Magazines > IEEE Software > Early Access

Gains from improved recruitment of software developers

TABLE 1. Four studies on productivity ratios and right-skew in teams of seven developers with median values and intervals containing 80% of the values.

Study	Measure median values and [80% intervals]	Max-min ratio $\frac{\text{highest value}}{\text{lowest value}}$		Right-skew $\frac{(\text{highest value} - \text{middle value})}{(\text{middle value} - \text{lowest value})}$		Percentage of tasks solved by the 20% most productive in an organization (Q3)
	Context	Ratio of work-effort on the same task (Q1)	Ratio of produced output given same work- effort (Q2)	Right-skew of work-effort on the same task (Q1)	Right skew of produced output given same work- effort (Q2)	
Survey of the experiences of 75 software professionals		2.7 [2.0-5.2]	3.0 [2.0-5.4]	1.6 [1.0-2.8]	1.4 [1.0-2.5]	50% [30%-84%]
Data from 20 software professionals solving the same five tasks [4]		3.4 [2.7-5.3]	3.6 [2.7-5.3]	1.4 [0.5-5.4]	2.8 [0.7-6.6]	36% [34%-38%] ⁱ
Data from 494, mainly software professionals, solving the same 10 tasks [5]		4.3 [2.5-8.0]	4.3 [2.5-8.0]	2.1 [0.8-6.3]	2.0 [0.7-6.0]	43% [39%-52%] ⁱ
Data from 16 developers completing the same project [6]		5.5 [3.4-10.7]	5.5 [3.4-10.7]	2.0 [0.8-5.3]	3.0 [1.0-8.1]	46% ⁱⁱ

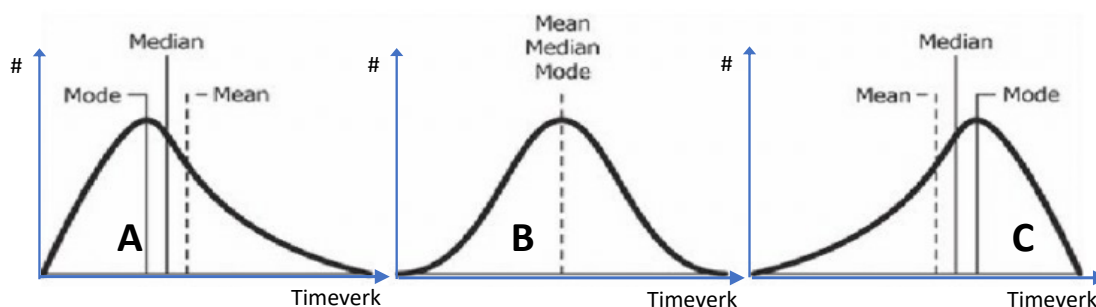
i) This interval represents the variation across tasks and is not directly comparable with the survey-based interval.

ii) Only one project, no interval possible.

12

Anta at du gir 1000 utviklere samme middels komplekse oppgave. Hvilken av fordelingene tror best gjenspeiler mengde arbeid (antall timeverk) for å løse oppgaven med akseptabel kvalitet?

A) Høyreskjev (lang hale mot høyt arbeidsforbruk), B) Symmetrisk, C) Venstreskjev (lang hale mot lavt arbeidsforbruk)



13

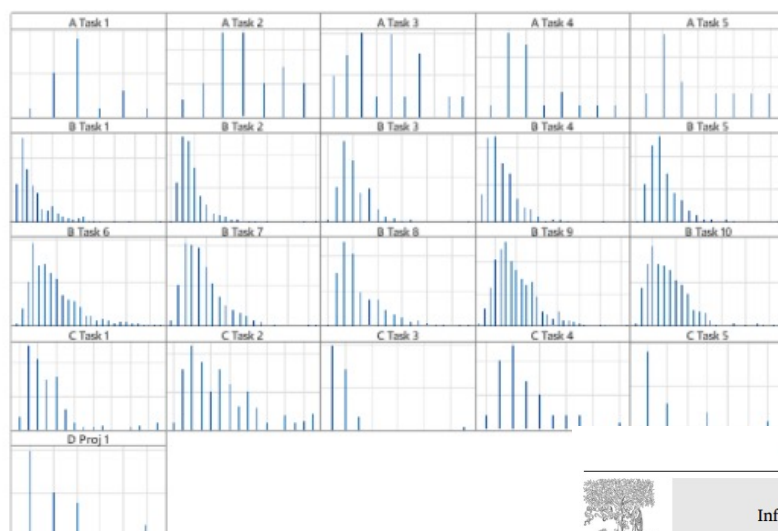


Fig. 1. Histograms of the empirical productivity distributi

Fordeling på arbeidsmengde brukt på samme oppgave (21 ulike oppgaver/prosjekter)

Høyreskjev.

Lognormal bruk av arbeidsmengde.

(som gir lognormal fordelinger på produktivitet)



Contents lists available at ScienceDirect
Information and Software Technology
journal homepage: www.elsevier.com/locate/infsof



Characteristics and generative mechanisms of software development productivity distributions

Magne Jørgensen

Simula Metropolitan Center for Digital Engineering, Oslo, Norway



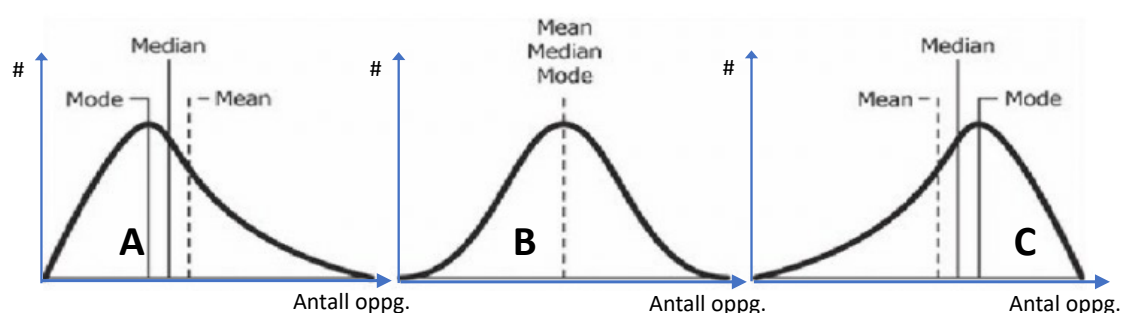
14

Anta at du har de samme 1000 utviklere som nå løser oppgaver av lignende størrelse og kompleksitet (men de rekker naturlig nok ulikt antall oppgaver).

Hvilken av fordelingene tror best gjenspeiler

antall oppgaver løst i løpet av år?

- A) Høyreskjev (lang hale mot høyt antall oppgaver løst),
B) Symmetrisk, C) Venstreskjev (lang hale mot få oppgaver løst)



15

Gains from improved recruitment of software developers

Table 2: Parameter input to the BCG model and benefits from improved validity of recruitment

Parameter values					
Evaluation score validities (r)	Two recruitment validity situations: <i>Low validity</i> with $r=0.2$ and <i>High validity</i> with $r=0.5$.				
Evaluation score distributions ⁱ	Low validity recruitment: Normal distribution with a mean of 100 and a standard deviation of 50. High validity recruitment: Lognormal distribution with a mean of 100 and a standard deviation of 50.				
Productivity differences	Standard deviation of value creation of 35.000 EUR per developer per year ⁱⁱ .				
Employment length	3 years				
Selectivity levels (p)	Very high $p=10\%$	High $p=30\%$	Medium $p=50\%$	Low $p=70\%$	Very low $p=90\%$
Resulting value creation improvement from going from low ($r=0.2$) to high ($r=0.5$) validity recruitment					
Expected improvement in value creation (rounded to nearest 1000 EUR)	78.000 EUR	38.000 EUR	21.000 EUR	11.000 EUR	3.000 EUR
Expected relative gain in value creation (rounded to nearest 10%)	310%	260%	230%	200%	170%

i) The actual evaluation scores and their distributions are assumed transformed, with no loss in generality, to the given distributions. The use of a log-normal, instead of a normal, distribution for the higher-validity situation is motivated by the typically right-skewness of the scores on work sampling tests and several other higher-validity, practice-based skill tests.

ii) The standard deviation is derived from a Norwegian context (www.menon.no/wp-content/uploads/2016/04/rapport-betydningen-av-ikt-v5.pdf), where the net mean value creation per year of software developers has been calculated to be about 70.000 Euro. Using this information and the empirically found coefficient of variance close to 0.5 documented for software developers in, for example, [10] and for many other types of work in [11], we get a standard deviation of 35.000 Euro.

Eksempel på konsekvens

Dersom vi går fra en tradisjonell (og dårlig) rekrutteringsmetode til en høy-valid metode (utprøvningsbasert, strukturerte intervjuer, flere typer tester) ...

og kan velger blant de 30% som gjør det best ...

så vil vi (norske forhold) øke verdiskapningen med en faktor på 2.6.

(gjør egne beregninger på:

tinyurl.com/bcg-recruitment)

16

Hvordan finner man de mest (og unngår de minst) produktive – og hvor lønnsomt er det med bedre rekrutteringsprosesser?

17

Hvordan ikke gjøre det ...

- Vektlegge lengde erfaring, om de er senior eller junior-utviklere, lengde på utdannelse og annen CV-basert informasjon. Disse har en svært lav validitet (korrelasjon med faktisk produktivitet på kun ca. 10%)
- Ustrukturerte intervjuer har litt høyere korrelasjon (20%), men fortsatt for svært lav validitet.
 - Her ser vi en stor «overtro» på egne evner til å avdekke kompetanse og produktivitet gjennom intervjuer.
- Vektlegge om en person estimerer høyt eller lavt for å gjennomføre en oppgave
 - Dunning-Kruger effekt: De som vet lite, vet også lite om hvor lite de vet.

18

Hva virker (bedre)

- Ingen metode for rekruttering er svært god! (Kanskje unntatt å faktisk bruke «prøvetid» på noen måneder effektivt for å teste flere.)
- Det som fungerer best (mange undersøkelser, få fra IT-utvikling) er
 - Structured employment interviews ($r=0.42$), i.e., interviews where all applicants are asked the same questions in the same sequence, preferably based on the same, pre-defined guidelines for how to score the answers.
 - Job knowledge tests ($r=0.40$), i.e., questions and tests tailored to identify the candidates' knowledge about the facts, principles, concepts, and other knowledge important for job performance.
 - Empirically keyed biodata ($r=0.38$), i.e., questions aiming at identifying candidates' characteristics (similar to personality evaluation) regarding, amongst others, work habits, problem-solving approaches, interpersonal relations skills, and situation perseverance.
 - Work sample tests ($r=0.33$), i.e., tasks to be solved by the candidate similar to those to be solved in the ordinary work context. For software developers, this may include small and medium-large programming tasks.
 - Cognitive ability tests ($r=0.31$), i.e., tasks that measure the candidates' ability to think, solve logical problems and learn. GMA (General Mental Ability) and IQ tests are two important examples of such tests.
 - Integrity test ($r=0.31$), i.e., questions designed to assess an applicant's tendency to be honest, trustworthy, and dependable.

Kombinasjoner av disse kan gi validitet (korrelasjon) opp til 70-80%.

19

Hvor mye kan man tjene på å forbedre rekrutteringen

Her bruker jeg del velkjente (;-)) Brogden-Cronbach-Gleser (BCG) modellen

20

The expected increase (ΔU) in value creation from selecting a person among the developers with the p percent best scores instead of a random selection is then:

III) $\Delta U = (b_0 + r_{XY} \frac{SD_Y}{SD_X} \bar{q}_{(1-p)}(X)) - (b_0 + r_{XY} \frac{SD_Y}{SD_X} \bar{q}_{(0)}(X)) = r_{XY} \frac{SD_Y}{SD_X} (\bar{q}_{(1-p)}(X) - \bar{x})$, where $\bar{x}$ is the mean value all persons (the expected value with random selection). Comparison of ΔU for test strategies with, for example, different test score validities will now give their difference in expected value creation.

Vi antar en lineær sammenheng mellom test-score og ytelse (verdiskapning), bruker standard regresjonsanalyse, gjør noen statistiske transformasjon, og vipps så har vi et uttrykk der forbedring i verdiskapning i all hovedsak er en funksjon av validitet til test (korrelasjon mellom test-score og ytelse, r_{XY}), variasjon i verdiskapning (SD_Y), og hvor selektive vi er i utvelgelsen (p).

21

Da kan vi f eks regne ut at vi - ved en 3 års ansettelse – får en

gevinst på ca. 450.000 NOK (38.000 Euro) ved å gå fra lav (20%) til høy (50%) validitet i rekrutteringen, antatt en normal produktivitetsfordeling, norske forhold på verdiskapning per utvikler, og kun rekruttering blant de med 30% best evaluerings-score.

Productivity differences	Standard deviation of value creation of 35.000 EUR per developer per year ⁱⁱ .				
Employment length	3 years				
Selectivity levels (p)	Very high p=10%	High p=30%	Medium p=50%	Low p=70%	Very low p=90%
Resulting value creation improvement from going from low (r=0.2) to high (r=0.5) validity recruitment					
Expected improvement in value creation (rounded to nearest 1000 EUR)	78.000 EUR	38.000 EUR	21.000 EUR	11.000 EUR	3.000 EUR
Expected relative gain in value creation (rounded to nearest 10%)	310%	260%	230%	200%	170%

22

Det vi vet med rimelig sikkerhet om produktivitet i IT-utvikling ...

- Vi har i de aller fleste tilfeller ingen gode metoder for å måle produktivitet i IT-utvikling (Mangler mål på «output»)
 - For å få til gode produktivetsmålinger må vi sammenligne kostnader for «samme» output (-> trenger kontrollerte undersøkelser)
 - Vi klarer ikke å prise konsulenter relativ til den nytten de leverer
- Forskjellene i produktivitet er typisk store, både på individ, team og organisasjonsnivå.
 - Hvor store forskjellene er avhenger av mange faktorer. På individnivå, og middels vanskelige programmeringsoppgaver så vil typisk de 20% mest produktive levere 40-50% av leveransene.
- Distribusjonen av produktivitet er høyreskjev og trolig «lognormal»
 - «Synergi» (produktet er mer enn summen av delene) mellom faktorer som gir produktivitet
- Det er mye å spare på gode rekrutterings- og seleksjonsrutiner.
 - Ingen enkeltmetode for rekruttering er veldig bra (korrelasjon med ytelse lavere enn 0.5), men flere sammen gjør at validiteten kan bli bra (korrelasjon mellom 0.6 og 0.7).
 - Gode metoder inkluderer: strukturerte intervjuer, «work sampling testing», «job knowledge tests», «cognitive biodata», kognitive tester og «integritet testing».
 - Lite effektive metoder inkluderer: «ustrukturerte intervjuer», lengde erfaring og lengde utdannelse.

23

The end is here ...

24

Table 4 Development and maintenance efficiency data for each of the projects

CASE tool	Tool A		Tool B		Tool C			Tool D		
Dev. project	1	2	3	4	5	6	7	8	9	10
UDE	0.12	0.14	0.22	0.79	0.88	0.91	0.17	0.41	0.63	0.08
ADE	0.16	0.23	0.26	0.66	0.71	0.94	0.36	0.40	0.54	0.16
MBTE	71	72	21	n.m.	2.0	9.0	n.m.	n.m.	35	84
Dev. WE	38.1	35.4	12.3	14.3	3.7	8.0	67.0	20.0	27.1	26.7
Dev. WA	0.03	0.03	0.08	0.07	0.27	0.13	0.02	0.05	0.04	0.04
Dev. WS	4.5	5.1	2.7	11.3	3.3	7.2	11.1	8.9	17.0	2.2
MBTS	n.m.	1100	82	n.m.	22	115	n.m.	n.m.	580	230
Maint. WS	n.m.	15.3	3.9	n.m.	11.0	12.8	n.m.	n.m.	16.6	2.7
TP	2.9	4.3	2.1	0.7	2.0	1.6	3.6	1.0	3.6	1.7

(Dev. = Development, UDE = Unadjusted Development Efficiency, ADE = Adjusted Development Efficiency, MBTE = Maintenance Benchmark Task Effort (Tasks 1-7), WE = Work Expansion, WA = Work Avoidance, WS = Work Speed, MBTS = Maintenance Benchmark Task Size (Tasks 1-7), Maint. = Maintenance, TP = Time Pressure)

[An empirical study of the correlation between development efficiency and software development tools](#)

M Jørgensen, SS Bygdås - Elektronikk, 1999

Budskap:

- 1) Det er mulig å måle produktivitet.
- 2) Målingene bør ha et formål
- 3) Resultatene er vanskelig å tolke og er en kombinasjon av individuell og team-»skill», utviklingsmiljø, «IT-løsning» og mange eksterne faktorer (som kunde/produkteier ..)

Flere måter å måle produktivitet på i denne oversikten. De viktigste:

- 1) UDE: Function points/timeverk (ADE: Funksjonspoeng justert for kompleksitet)
- 2) WA (Work Avoidance): Linjer kode/timeverk
- 3) MBTE: Benchmark-oppgave (alle teamene gjør – eller estimerer – samme vedlikeholds-oppgave i sitt utviklingsmiljø) og tiden måles.
- 4) Hovedfunn: a) Store produktivitetsforskjeller (10x forskjell i output – målt i function points - per timeverk, UDE, og 6x for ADE). Størst forskjeller mellom, men også en del innenfor, samme utviklingsmiljø. Verktøy betyr trolig mye for produktivitet. B) Enda større produktivitetsforskjeller med utgangspunkt i «benchmarksoppgaver» (estimat av »samme» utvidelser). Helt opp til 35x! Her kommer også forskjeller i hvor enkle systemene er å utvide kraftig med.

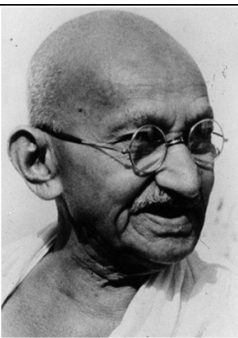
25



Mange mener noe om produktivitet

Trond Viggo Torgersen

Livet er innmari stutt
Før du får ut finger'n kan det alt være slutt
Har du mye du sku' gjort
Så må du ikke gå så fort



There is more to life than increasing its speed.

— Mahatma Gandhi —

AZ QUOTES



Livet er for kort til å tygge

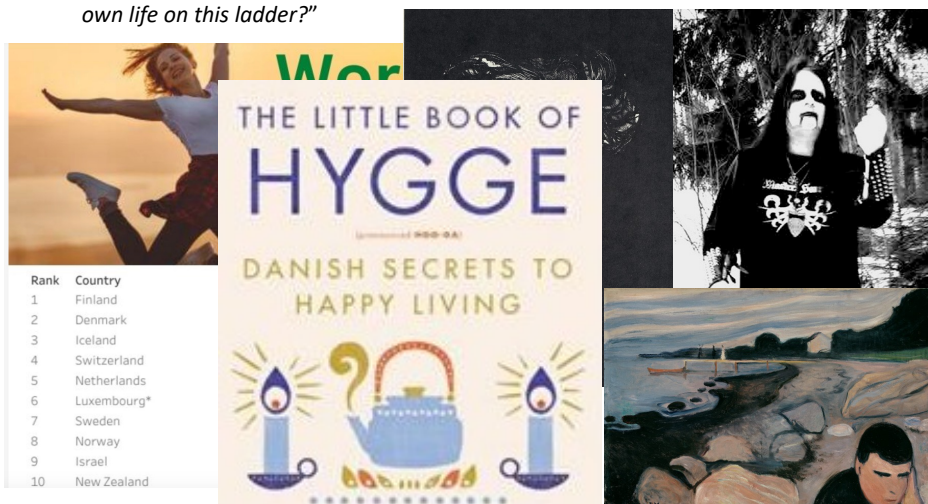


If you want something done, give it to a busy man (Sturges)

26

Lykke (fornøydhet) og produktivitet er nært knyttet sammen

The Cantril Self-Anchoring Scale: "Think of a ladder, with the best possible life is a 10, and the worst possible life being a 0. Where would you put your own life on this ladder?"

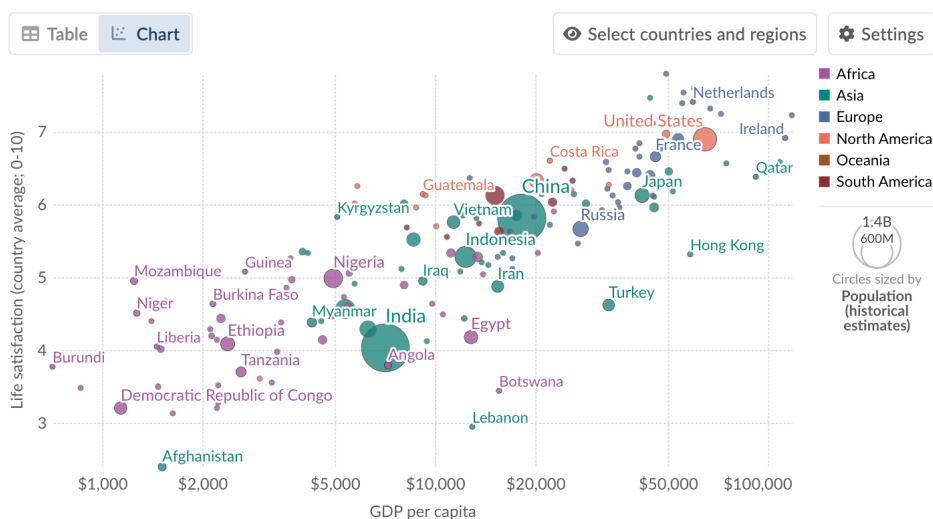


27

Ikke kom og si at rikdom ikke er viktig for et lands lykke

Self-reported life satisfaction vs. GDP per capita, 2022

Self-reported life satisfaction is measured on a scale ranging from 0-10, where 10 is the highest possible life satisfaction. GDP per capita is adjusted for inflation and differences in the cost of living between countries.



28

Investeringer i IT-utvikling → Høyere produktivitet →
Mer velstand → Mer fornøyd befolkning



Information Economics and Policy
Volume 25, Issue 3, September 2013, Pages 109-125



ICT and productivity: conclusions from the empirical literature

M. Cardona ^a, T. Kretschmer ^{a, b}, T. Strobel ^b

“... 10% increase in ICT investment leads to a 0.6% increase in growth”

«... the growth impact of ICT has grown over time.”

«... ICT as a General Purpose Technology enables subsequent innovations.”

Introduction of ICT has had a larger effect on productivity than steam, and at least a similar impact to that of electricity.

Crafts, Nicholas, The Solow Productivity Paradox in Historical Perspective (January 2002). CEPR Discussion Paper No. 3142.