

A1

Kristoffer Kortbaek (jnq578, hold 5)

May 6, 2022

Contents

1	1	1
2	2	2
3	3	2
4	4	3
1	1	

Frøkjær, lektor i datalogi på Københavns Universitet, påpeger i sit debatindlæg “Finansministeriet”[2] en række årsager til en række IT skandaler hos SKAT. Hans første udpegning omhandler, at skandalerne starter efter et krag om at sænke udgifter i skatteadministrationen. En af måderne dette kan opfyldes, er ved at indføre en personalereduktion på 25%. Frøkjær mener, at dette kun kunne lade sig gøre hvis en række dårligt vedligeholdte IT systemer blev “saneret”. Herefter kommer Frøkjærs første egentlige pointe. Det lykkedes kun en andel af IT-saneringsprojekterne at blive gennemført, men planen om at lave personalereduktion fortsætter på trods af dette. Frøkjær sætter spørgsmål ved hvordan dette kunne lade sig gøre, og han nævner således. Frøkjær mener selv, at der burde blive stillet spørgsmålstegn til hvilken rolle finansministeriet har spillet skandalerne hos SKAT.

Dernæst kommer Frøkjærs anden pointe. Det var Finansministeriet der startede personalereduktionen på trods af manglende IT infrastruktur. Han påpeger desuden at det var Finansministeriet der pressede på med inddrivelsessystemet EFI, og han påpeget desuden det skulle have en urealistisk grad af automatisering. Endvidere nævnes at dette ikke er første gang at

Finansministeriet angiveligt har presset på med implementering af IT systemer der har skulle implementere urealistisk grad af automatisering.

2 2

Artiklen “Why Did Your Project Fail”[1] generaliserer en række punkter til ofte at være medskyldige i fejlende IT projekter, og kalder disse for fejl faktorer. Dette var gjort ved at udsende et spørgeskema med 88 spørgsmål omkring forskellige aspekter af softwareudvikling. Svarene fra spørgeskemaet blev besvaret af 304 forskellige projekter, hvoraf 70 af dem blev kategoriseret som værende fejlagtige IT projekter. Herefter blev der udarbejdet en rangering af de hyppigst forekommende grunde til at IT projekterne fejler. Disse inkluderer, leveringsdato påvirker udviklingsprocessen, manglende risikostyring gennem hele projektet, dårlig projekt estimering og flere. Desuden lægger artiklen vægt på forskellen mellem grundende til fejlende IT projekter afhængigt af om projektet blev udviklet in-house eller outsourced, da der angiveligt er forskel i disse. Mangel på kompensation for overarbejde giver blandt andet anledning til fejlende IT projekter for in-house udvikling, hvorad dette ikke er et problem når udviklingen var outsourced.

Artiklen nævner derudover at kategoriseringerne af elementer der er fælles for fejlende IT projekter danner et netværk af elementer der bidrager til at IT projektet kan fejle. Eksempelvis medvirker at projektet var underestimeret at flere medarbejdere var tilføjet til projektet sent, hvilket i sig selv er en fejl faktor som nævnes i artiklen.

3 3

Frøkjær påpeger tidligt i sit debatindlæg, hvorfor SKAT fortsatte personalereduktionen efter det allerede var klart at IT-saneringsprojekterne ikke ville blive færdiggjort. Herved påpeger han selvsamme fejl faktor som Cerpa og Verner påpeger i “Why Did Your Project Fail?”[1], når de beskriver risici der ikke bliver genovervejet, kontrolleret eller styret gennem projektet. I følge Cerpa og Verner kan denne fejl faktor hænge sammen med, at IT-projektet ikke har haft nok medarbejdere, og et ubehageligt arbejdsmiljø. Deres model siger dog ikke noget om hvordan disse tre fejl faktorer påvirker hinanden, men blot at der er en sammenhæng mellem dem. Det vil altså sige, at vi ved ikke om IT-projektet hos SKAT manglede medarbejdere, som derfor førte til dårlig risikostyring, eller den dårlige risikostyring førte til manglende medarbejdere. Dette minder på sin vis også om måden at

se teknologi som systemer. I stedet for at se teknologi som systemer som beskrevet af Hughes[5], så ser vi på typiske elementer tilfælles for fejlende IT projekter som et system. Heri består noderne af netværket af fejl faktorer, og ændringen i en af disse fejl faktorer kan påvirke resten af netværket.

Frøkjær kritiserer også Finansministeriets krav omkring mængden af automatisering der skulle være fundet sted i gældsindrivelsessystemet EFI. Dette ligger også op til fejl faktoren omkring urealistisk forventning[1]. I deres netværk af fejl faktorer, er denne særligt koblet til mange andre faktorer, som netop antyder at dette problem som Frøkjær tager op også førte til andre fejl faktorer i EFI projektet. Cerpa og Verner påpeger blandt andet at urealistisk forventning kan hænge samme med manglende medarbejdere, som i sej selv fører til at ekstra medarbejdere bliver tilføjet til projektet sent. Dette hænger også sammen med Frøkjærs påpegning at Finansministeriet satte ekstra bevillinger af til EFI projektet.

4 4

Hvis man skal se på hvilke aktører, der generelt kunne have negativ påvirkning på IT projekter, så peger Frøkjær hvertfald på eksterne interessenter i hans kritik af Finansministeriets rolle i skandalen hos skat. Man kan måske komme dette endnu nærmere, hvis man sætter Frøkjærs pointer overfor begrebet “Computational Thinking”[3] og de fejler faktorer[1] Cerpa & Verner nævner. Wing nævner nemlig, computational thinking metoden mest er kendt inden for datalogi og softwareudvikling[3], men at denne burde bruges i andre felter, da den ifølge Wing handler om “prevention, protection, and recovery from worst-case scenarios”[3]. Samtidigt snakker Cerpa & Verner om fejl faktoren om manglende revurdering af risici under IT projektet[1]. Hermed kan man slutte, at eksterne interessenter eller interne medarbejdere på et IT projekt, der ikke netop konstant under IT projektets levetid genovervejer risici har en negativ påvirkning på offentlige og private IT-projekters gennemførsel.

Hvori Frøkjær beskylder Finansministeriet for fejlene gennemførsel af IT projekter i SKAT, så kunne man omvendt også tænke sig, at det i stedet handlede om nogle af de interne arbejdsgange indenfor et IT projekt der får det til at fejle. Dette kunne eksempelvis være mangel på kompetencer eller disciplin inden for datalogi. Artiklen “Who are We”[4] diskuterer netop datalogi som profession, og nogle af de kriterier der burde være gældende for profession. Han nævner blandt andet der skal være grundlæggende metoder for at designe og implementerer sikre og holdbare software systemer[4]. En

udvidelse af Denning kunne hermed naturligt være, at der som del af datalogi professionen også skulle være krav omkring hvordan man opnår gennemførsel af IT-projekter. Sat sammen med “Computational Thinking”[3] der netop handler om kontinuerlig vurdering, så passer dette naturligt til datalogiens metoder.

References

- [1] Cerpa, Narciso og June M. Verner (dec. 2009). „Why did your project fail?“ Communications of the ACM, bd. 52, nr. 12, s. 130–134. doi: 10.1145/1610252.1610286.
- [2] Frøkjær, Erik (9. jun. 2017). „Finansministeriet“. Weekendavisen, nr. 23.
- [3] Jeannette M. Wing. ^a“Computational Thinking“. Communications of the ACM, bd. 49, nr. 3 (mar. 2006), s. 33–35. DOI: 10.1145/1118178.1118215
- [4] Peter J. Denning. ^a“The Profession of IT: Who Are We?“ Communications of the ACM, bd. 44, nr. 2 (feb. 2001), s. 15–19. DOI: 10.1145/359205.359239
- [5] Hughes, Thomas P. (2012). „The Evolution of Large Technological Systems“. I: The Social Construction of Technological Systems. New Directions in the Sociology and History of Technology. Red. af Wiebe E. Bijker, Thomas P. Hughes og Trevor Pinch. The MIT Press, s. 45–76.