



Suurtest tarkvara-arendusprojektidest

Kiiresti, odavalt ja kõike?

Ander Tenno

Trinidad Consulting / TietoEnator Eesti



Täna jutuks

Mis on tarkvaraarenduses võimalik ja mõistlik tähtaja, projekti ulatuse ja ressursside mõttes?

- ☪ Tarkvara keerukusest
- ☪ Arenduskiiruse hindamine
- ☪ Mehitatuse arvutamine
- ☪ Mahu hindamine
- ☪ Mis siis, kui...?



Lühidalt esinejast

☉ Töökogemus:

- | 1995-2001 programmeerija (HUT)
- | 2001-2003 programmeerija (TietoEnator)
- | 2003-2004 projektijuht (TE)
- | 2004-2008 arendusjuht / tootmisjuht (TE)
- | 2008- konsultant, Trinidad Consulting

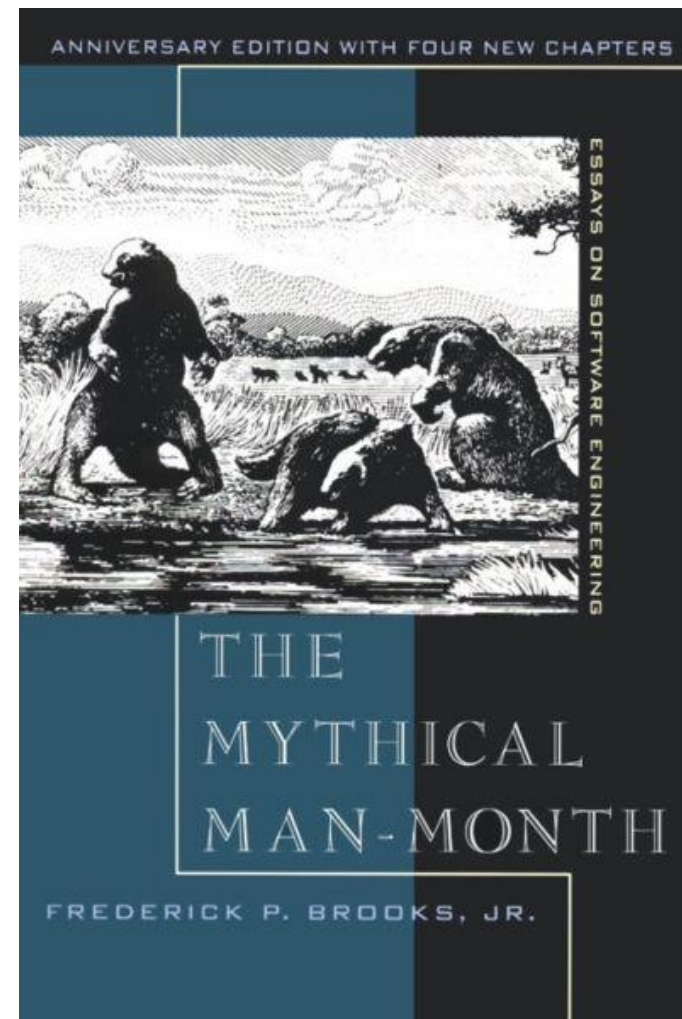
☉ Infotehnoloogiline haridus:

- | B.Sc. (Tallinna Tehnikaülikool 1999)
- | M.Sc. (Helsinki Tehnikaülikool 2000)
- | Dr.Sc. (Helsinki Tehnikaülikool 2004)

Tarkvaraarenduse keerukusest

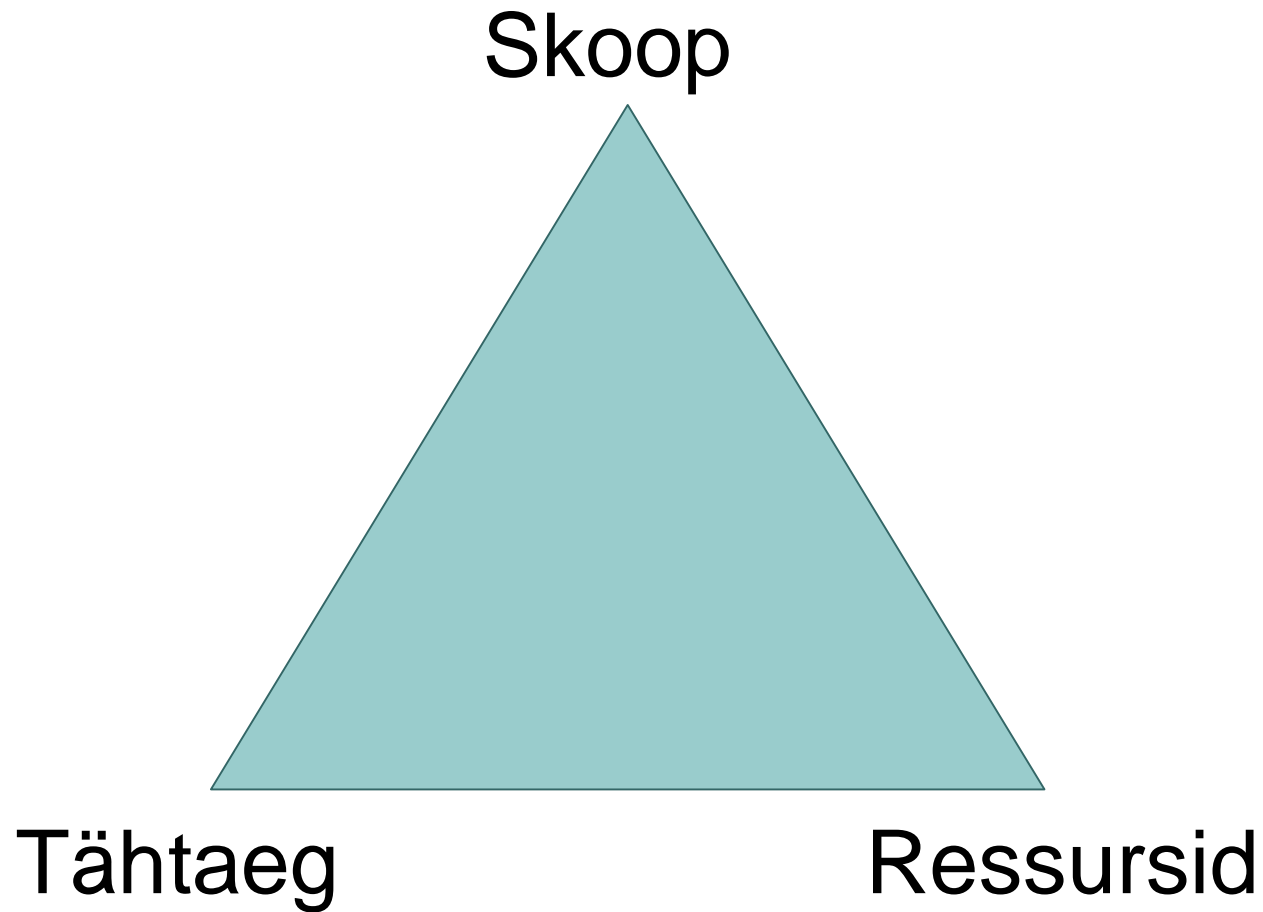
- ☉ Tarkvara omadused*
 - | Keerukus (*complexity*)
 - | Sobitatavus (*conformity*)
 - | Nähtamatus (*invisibility*)
 - | Muudetavus (*changeability*)
- ☉ Edasiarendused reeglina halvendavad kvaliteeti, kui seda fakti arvesse ei võeta

**There is No Silver Bullet* (Frederick P. Brooks, Jr)





Projekti “raudne” kolmnurk





Barry Boehmi valem uue projekti kestuse kohta

$$T = 2.5\sqrt[3]{MM}$$

$$T_{\min} = 0.75(2.5\sqrt[3]{MM})$$

- ☉ ***T*** – projekti valmimiseks kuluv optimaalne aeg
- ☉ ***T*_{min}** – projekti valmimiseks kuluv minimaalne aeg
- ☉ ***MM*** – töömaht inimkuudes
- ☉ **300 inimpäeva:** *T* = 6,1 kuud; *T*_{min} = 4,5 kuud
- ☉ **3600 inimpäeva:** *T* = 13,9 kuud; *T*_{min} = 10,4 kuud
- ☉ **Väga palju muutuvate** või **väga suurte** projektide korral võib olla vajalik lisada 10-25%.



Miks Boehmi minimaalsest kestusest kiiremini ei saa?

☉ Inimesi ei saa lõputult lisada:

| Töö pole mõistlikult tükeldatav

| Suhtlemisele kuluv aeg kasvab järsult:

$$t_n = n(n-1)/2 \quad [t_5 = 10, t_9 = 36, t_{24} = 276]$$

| Järjest kasvavad nõudmised tellijale

| Eriti hull lõppfaasis, kus tuleb mängu Brooksi seadus: *Lisades mehi projekte, mis on jäänud hiljaks, jääb see veel rohkem hiljaks* (vanade olijate aeg kulub uute õpetamisele)

☉ Kiirustamise tagajärg: tehniline võlg



Kvaliteedi roll

- ☉ *Kui tarkvara ei pea olema kvaliteetne, võib ta täita kõik muud tingimused*
- ☉ Eeldusel, et sellega absurdini ei minda, tekivad küsimused:
 - | Kuidas mõõta kvaliteeti?
 - | Kui kaugele kompromissidega minna?
 - | Mida tähendab, et tarkvara(komponent) on “valmis”? (vt. näide)



Mehitatuse arvutamine

- ☪ Praktiline näide ja selgitused
- ☪ Mida suurem projekt, seda rohkem oodatakse tellijalt
- ☪ Kui palju tellija ressursi vaja?
 - | KMA: 1/3 täitja omast, aga võivad ka muud tööd teha (50-80% ajast)
 - | Vajalikud rollid peavad olema esindatud
- ☪ Kui nii palju pole, usaldada rohkem, oodata vähem ja vältida *benji*-juhtimist



Mahu hindamisest

- ☪ Eraldi ettekande teema, siin põgusalt
- ☪ Ilma kasutuslugudeta reeglina võimalik ainult empiiriline hinnang, näit. MinMax:
$$m = (min + 3 * avg + max) / 5$$
- ☪ Kasutuslugudega näit. *Use Case Points*
- ☪ Vajab kogemusi; suuri projekte reeglina alahinnatakse esimesel korral
- ☪ Võib kõikuda ärilistel põhjustel



Aga ikkagi. Kas saab...?

- ☪ Boehmi miinimumist saab üldiselt kiiremini ainult erakorralistel juhtudel:
 - | Tasemel areendusprotsess (eraldi teema)
 - | Mõlema poole tugev pühendumine (sh äripool, juhid) ja vastastikune usaldus
 - | Väga hästi klappivad kompetentsed tiimid
 - | Stabiilsed nõuded
 - | Valmisolek tingida kvaliteedis
- ☪ Miinimumini jõudmine on samuti saavutus, mis ei saabu juhuslikult (vt. eelmine punkt)



Veel teemasid tulevikuks

- ☪ Tarkvaraarenduse seosest äripoolega
- ☪ Lähteülesande täpsest sisust ja detailsusest
- ☪ Kuidas räägitu põhjal hankeid läbi viia
- ☪ Mida nõuda täitjalt ja vastupidi
- ☪ Projektijuhtimise metoodikatest
- ☪ Järelvalvest ja tema tegevustest
- ☪ Testimisprotsessist
- ☪ Juurutusest
- ☪ Mis siis ikkagi on tarkvara kvaliteet?
- ☪ Kasutatavuse järjest kasvavast rollist
- ☪ Mahuhinnangute metoodikatest täpsemalt



Aitäh!

Ander Tenno (ander.tenno@trinidad.ee)

Telefon: 52 636 52