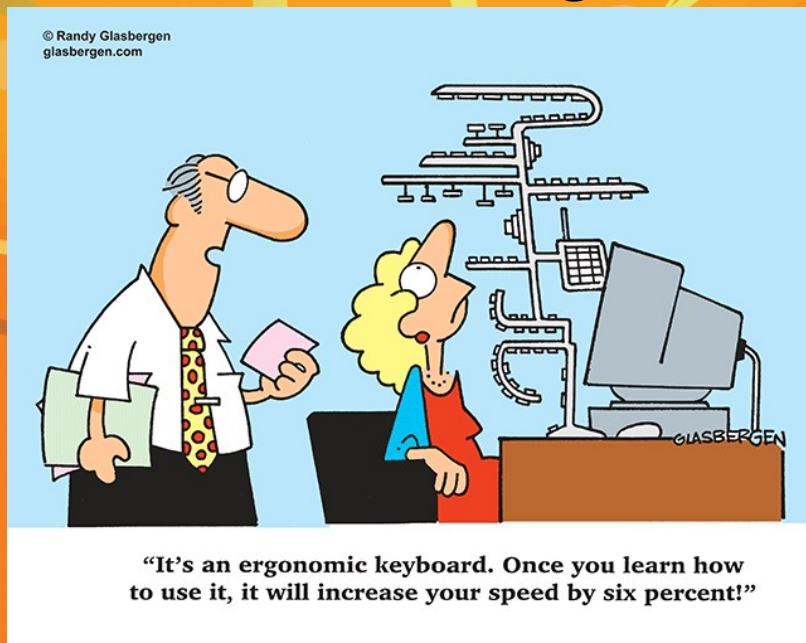


KLM

Mario Kamenjak

# Što je KLM?

- Keystroke level model
- Metoda prototipiranja
- Nije 'brzina računala' nego 'brzina korisnika'

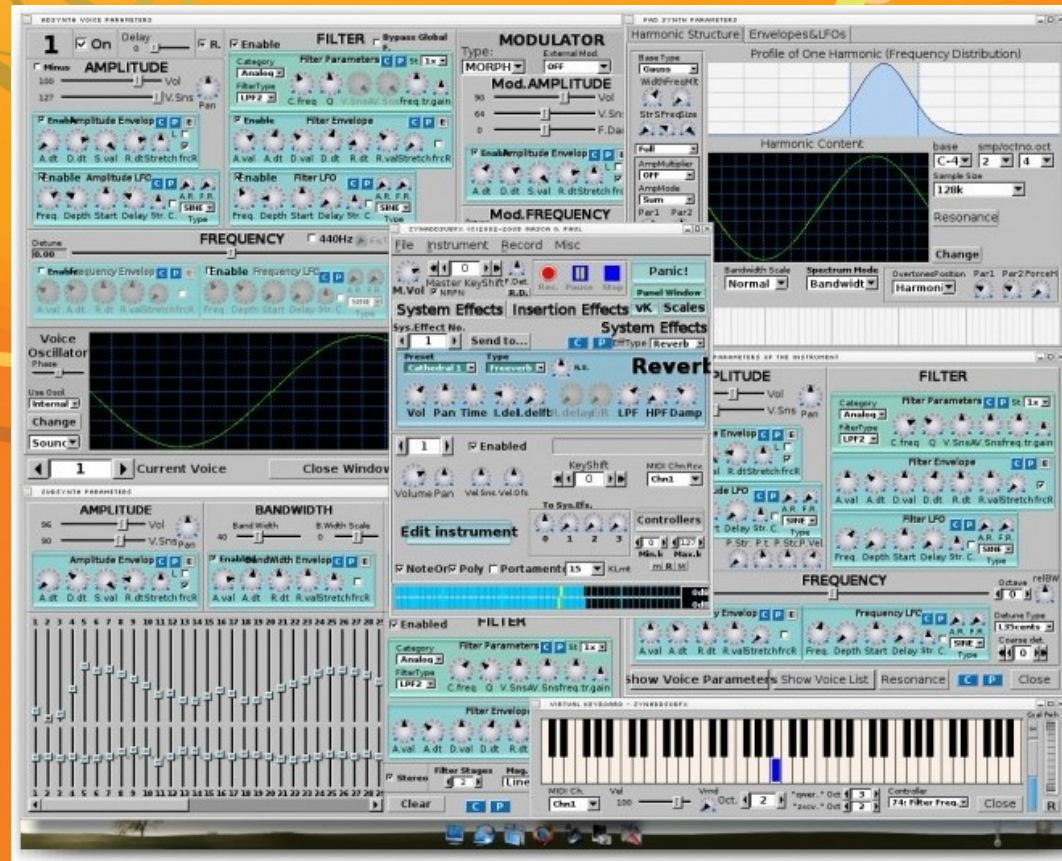


# Zašto KLM?

- Nije potreban prototip a niti testni korisnik (znači štedi novac i vrijeme)
- Trajanje operatora je testirano na uzorku od 20000 ljudi
- Brzina nekad pojednostavljuje

# Zašto ne KLM?

- Pretpostavke, pretpostavke, pretpostavke...
- Netreba ni sa KLM-om pretjerivati:



# Operatori

- 6 glavnih operatora:
  - K(Keystroke)-pritisak gumba ili tipke
  - P(Pointing)-'ciljanje' mišem
  - H(Homing)-premještanje ruke sa tipkovnice na miša i obrnuto
  - D(Crtanje)-kompliciran, na sreću većinom nepotreban
  - M(Mental)-mentalne operacije prije fizičke akcije, više na sljedećem slideu
  - R(system Response)-sistemski odziv, obično 0

# Operator M

- Operator mentalne pripreme za fizičku akciju, stavlja ga se po ovim pravilima:

Begin with a method encoding that includes all physical operators and response operations.

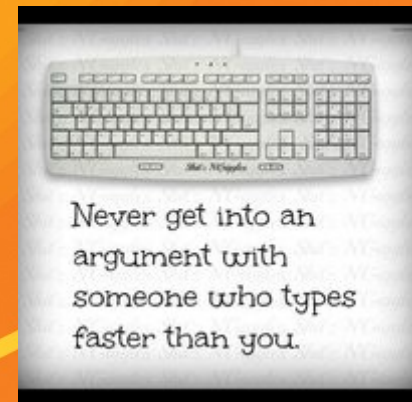
Use Rule 0 to place candidate Ms, and then cycle through Rules 1 to 4 for each M to see whether it should be deleted.

Rule 0	Insert Ms in front of all Ks that are not part of argument strings proper (e.g., text strings or numbers). Place Ms in front of all Ps that select commands (not arguments).
Rule 1	If an operator following an M is fully anticipated in the operator just previous to M, then delete the M (e.g., PMK -> PK).
Rule 2	If a string of MKs belong to a cognitive unit (e.g., the name of a command), then delete all Ms but the first.
Rule 3	If a K is a redundant terminator (e.g., the terminator of a command immediately following the terminator of its argument), then delete the M in front of the K.
Rule 4	If a K terminates a constant string (e.g., a command name), then delete the M in front of the K; but if the K terminates a variable string (e.g., an argument string) then keep the M.

# Trajanje operatora

- K-od 0.08s do 1.2s ovisno o vještini tipkanja
- P-1.1s
- H-0.4s
- D-.9nD +. 16 ID
- M-1.35s
- R-ovisno o sustavu

(ovo je sve statistički mjereno na uzorku od nekoliko tisuća osoba)



# 1. primjer

- Korisnik briše file tako da ga vuče u koš za smeće
- Pretpostavit ćemo prosječnog korisnika
  - (Brzina tipkanja)  $k=0.2s$



# Brisanje 'vučenjem'

- 1-'Početak' brisanja-M
- 2-Pronalazak ikone datoteke-M
- 3-Ciljanje na ikonu-P
- 4-Klik miša-K
- 5-Vučenje ikone na 'kantu za smeće'-P
- 6-Ciljanje na početni prozor-P

# Brisanje vučenjem

- $3P+K+2M=3*1.1+0.2+2*1.35=6.2s$



## 2. primjer

- Korisnik briše file pomoću kratice ctrl+t
- 1)Započeti-M
- 2)Pronaći ikonu file-a-M
- 3)Naciljati ikonu-P
- 4)Kliknuti mišem-K
- 5)Premještanje na tipkovnicu-H

# Brisanje ctrl+t

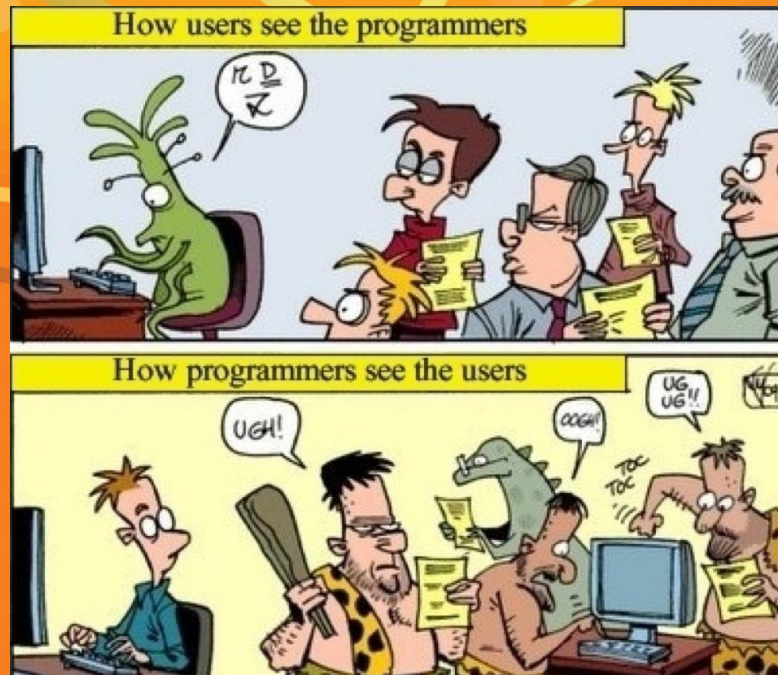
- 6) Stisnuti ctrl-K
- 7) Stisnuti t-K
- 8) Vratiti ruku na miša-H
- $P+2H+3K+2M=1.1+2*0.4+3*0.2+2*1.35=5.2s$

# Analiza

- Koji primjer je brži?
- Znači li da je on uvijek bolji?
- Mogu li imati prednosti od obadva?

# Zaključak?

- Brzina je važna...
- ...ali ima i drugih važnih stvari.
- Koja je vaša ciljna publika?



# Više

- <http://thebluedrag.blogspot.hr/2015/11/gui-vs-clickm-speed-analysis.html>
- [https://en.wikipedia.org/wiki/Keystroke-level\\_model](https://en.wikipedia.org/wiki/Keystroke-level_model)