Entirely Predictable Failures

Poul-Henning Kamp

phk@FreeBSD.org
phk@Varnish.org

POLSAG

”Integrated IT system for police”

Project start 2006

€ 21M

Roll-out complete 2009

POLSAG

2010 Pilot-test, smallest police district

Performance problems(!)

2012 entire project abandoned

Cost: € 65M

POLSAG

Time:	+100%
Budget:	+200%
Quality:	-100%

IC4

Contract signed 2000

All trains in traffic: 2004

Price: € 650M

IC4

Status:

- Trains not approved for service
- Unexplained failure to brake

Cost:

- Tricky.
- Much horse-trading
- High cost for renting replacements

Functionality:

- Hard to judge precisely
- Not at all popular with travellers

IC4

Prediction:
- Experimental service 2012
- Trains in limited service 2013

Cost:
- Parliament report 2011: € 1.2B

Functionality:
- Less than expected

IC4

Time:	+200%...+300%
Budget:	+100%...+150%
Quality:	-50%..-70%

RejseKort

€25M/y more expensive than paper tickets

…for each of the next 15 years.

Total extra cost: €365M

~= 1800 bus drivers

(Denmark has 570 bus routes)

&c

Digital Tinglysning
T:+200%, C:+200%, Q:-50%

System for ”comparing high-schools”
€ 5M, 6 years, never used

DeMars (SAP/3 for military procurement)
T:+100%, C:+15%, Q:-20%

…

In the pipeline

SKAT

9000 pages of specification
- not counting laws & regulations

EPJ (Electronic healthcare system)

3rd iteration
no luck with 1st & 2nd iteration
total costs unknown but > € 1B

Competent:

Average result better than expected

Incompetent:

Average result worse than expected

Competent:

Average result better than expected

Incompetent:

Average result worse than expected

Dunning-Krüger:

The incompetent don’t think so.

Inescapable Conclusion:

The Kingdom of Denmark is IT-incompetent

The mechanics of a disaster:

1. Formulate Goal
   Usually: ”Save money”

2. Decide how to reach goal
   Usually: ”Computerize”

3. Profit!
   Usually: Not!

Fresh Example:

1. Formulate Goal
   - Elections are too expensive
   - Reduced secrecy of voting for handicapped
   - Young people not interested
   - Man-power not readily available
   - Disqualified votes

2. Decide how to reach goal

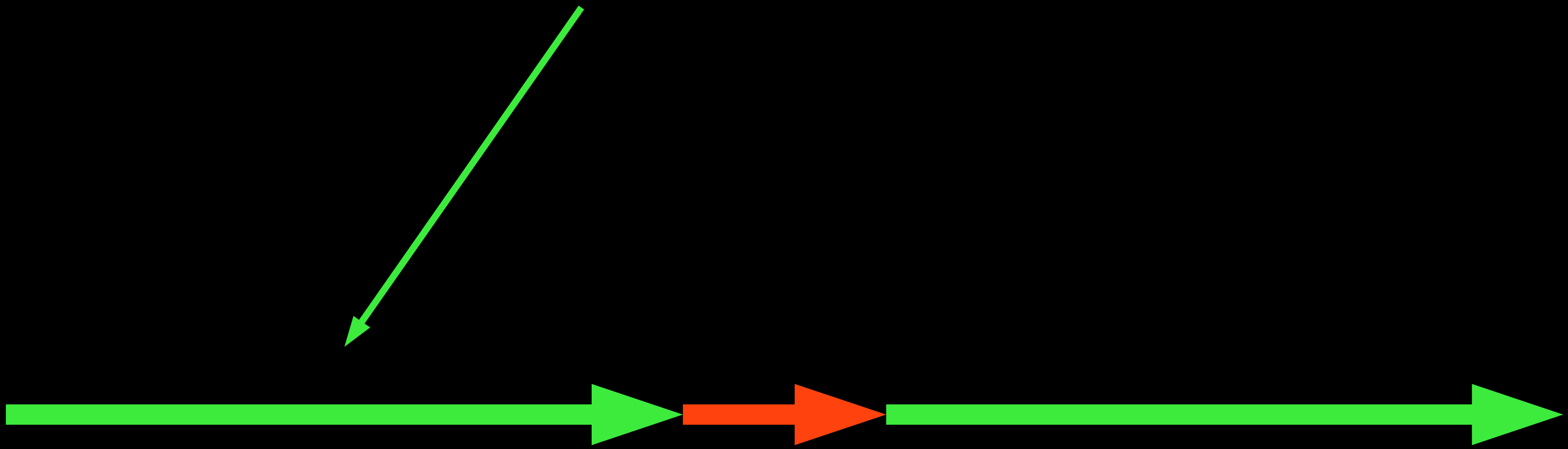
   Lets experiment with computer-voting

3. Profit!

   ?

Three phases of Danish elections:

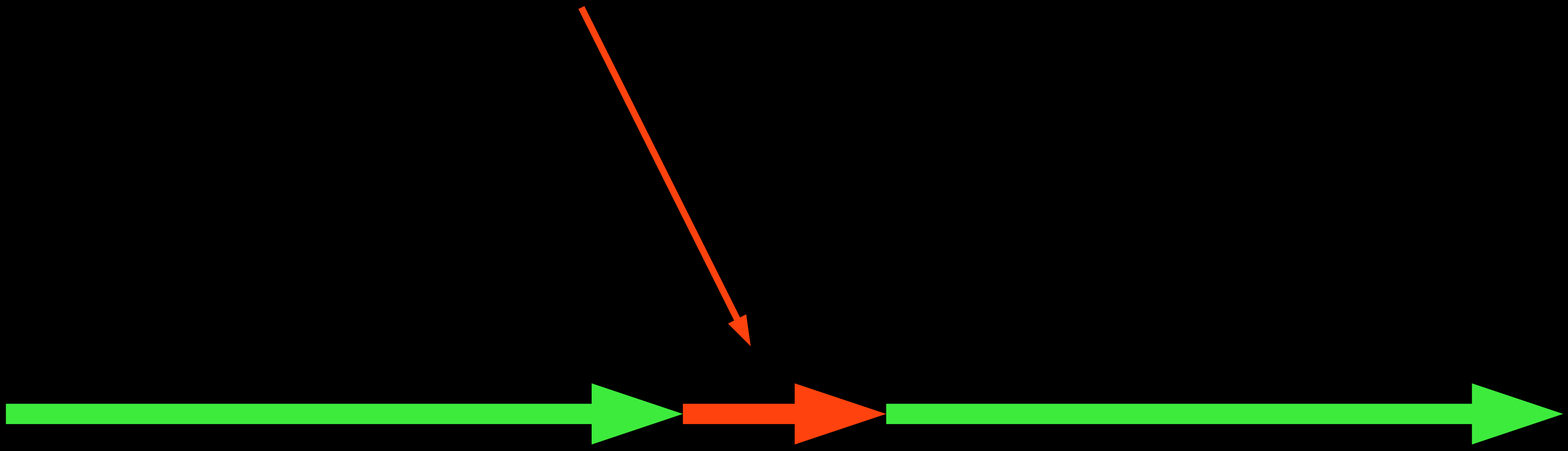
1: Totally open/transparent
Produce list of voters, ballots etc.
Mail election notice card to voters
Voter swaps notice card to ballot



Major costs: Postage, IT-transaction fees

Three phases of Danish elections:

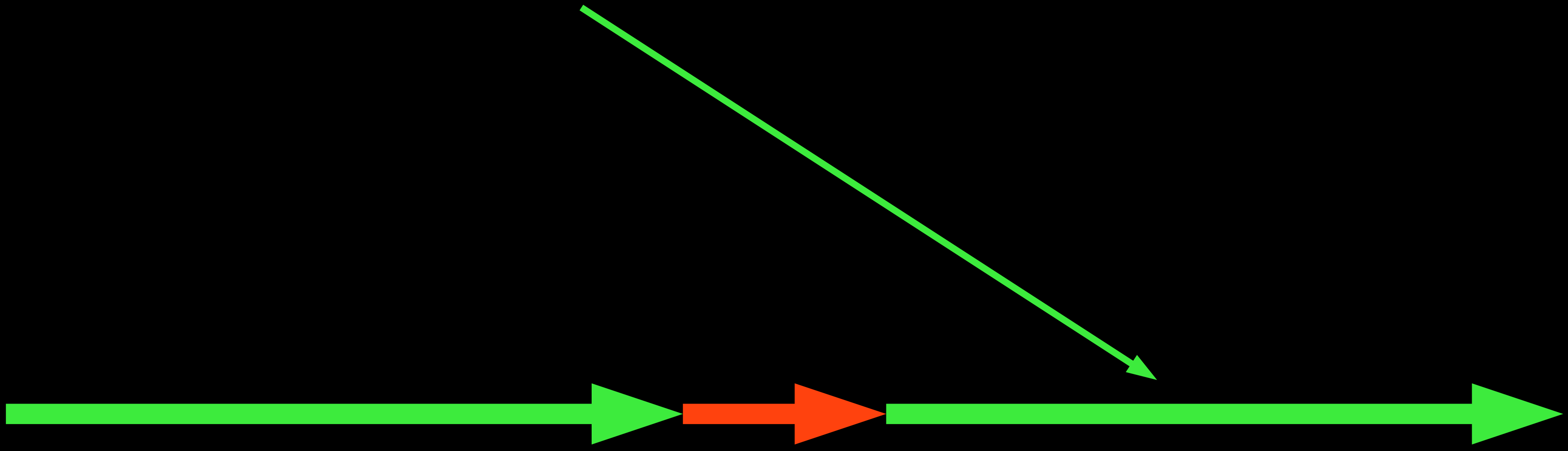
2: Totally secret
   Voter marks ballot
   Voter drops ballot in urn



Major cost: Food & pay for supervisors

Three phases of Danish elections:

3: Totally open/transparent
Ballots counted by hand
Cross-checks



Major cost: Public employees second count.

Quality of Danish elections:

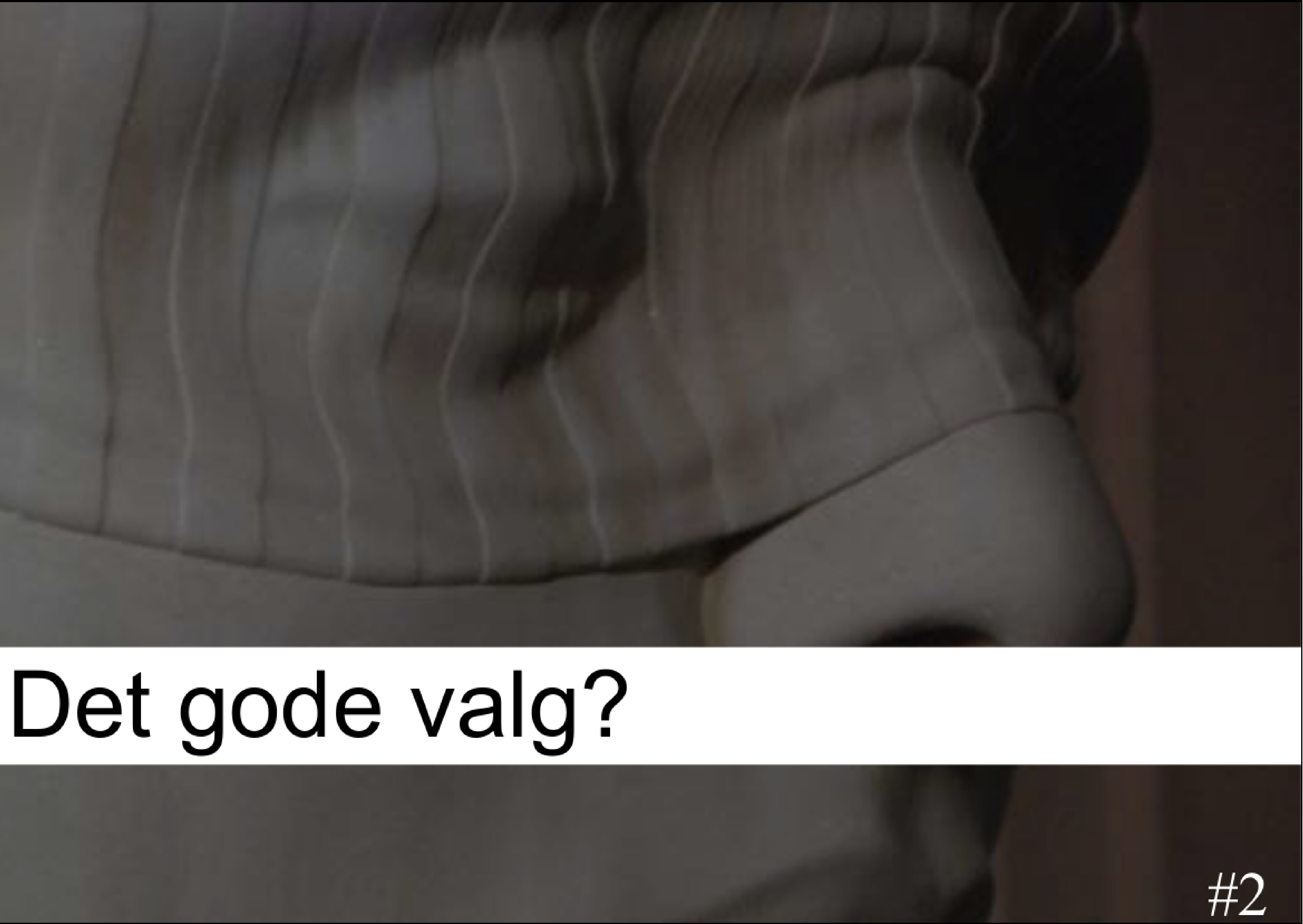
- Very high voting participation
- Very high trust in results
- Almost non-existent voting fraud
- Used as a model & exported to other countries

Cost of Danish elections:

- € 1.30 per capita per year.

Lene Hartig Danielsen Chef for Borgerservice Aarhus

#1

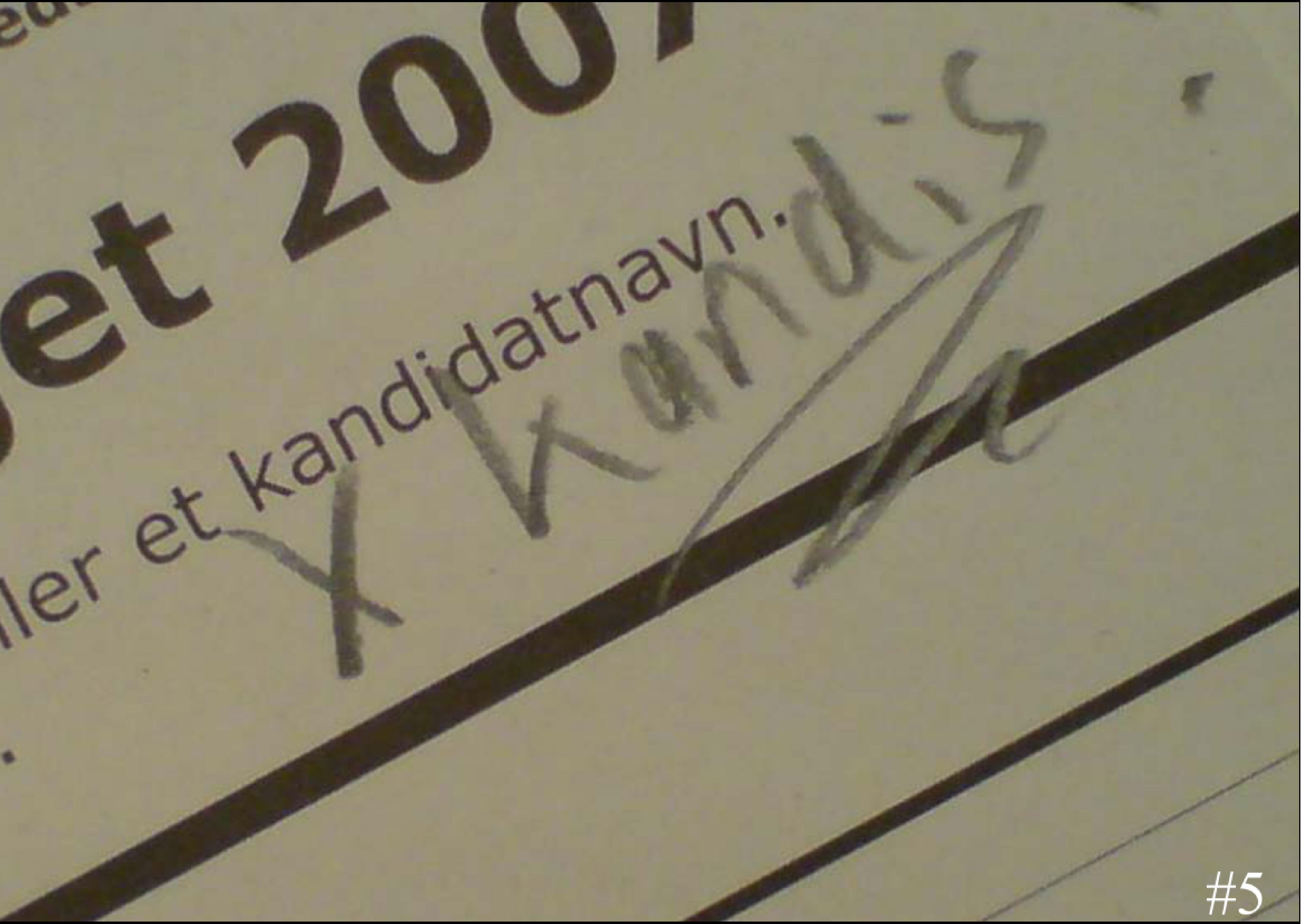


Det gode valg?

#2

#3

#4



edu

et 2007

ler et kandidatnavn.

X kandis

#5

#6

#7

#8

Say What ?

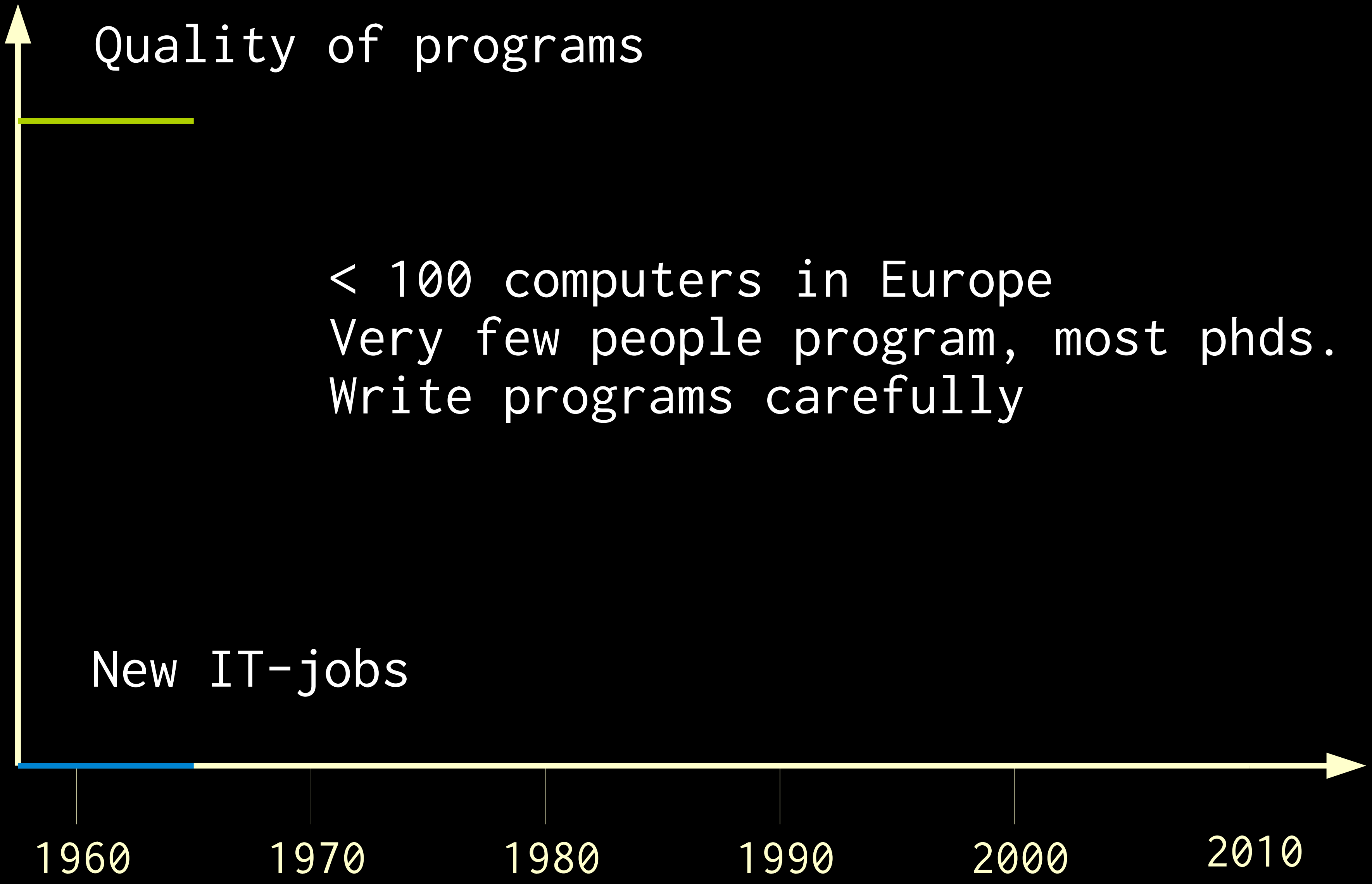
”We plan to use between €.13M and €13M”

”No, we don't have a business case”

No factual/numerical basis

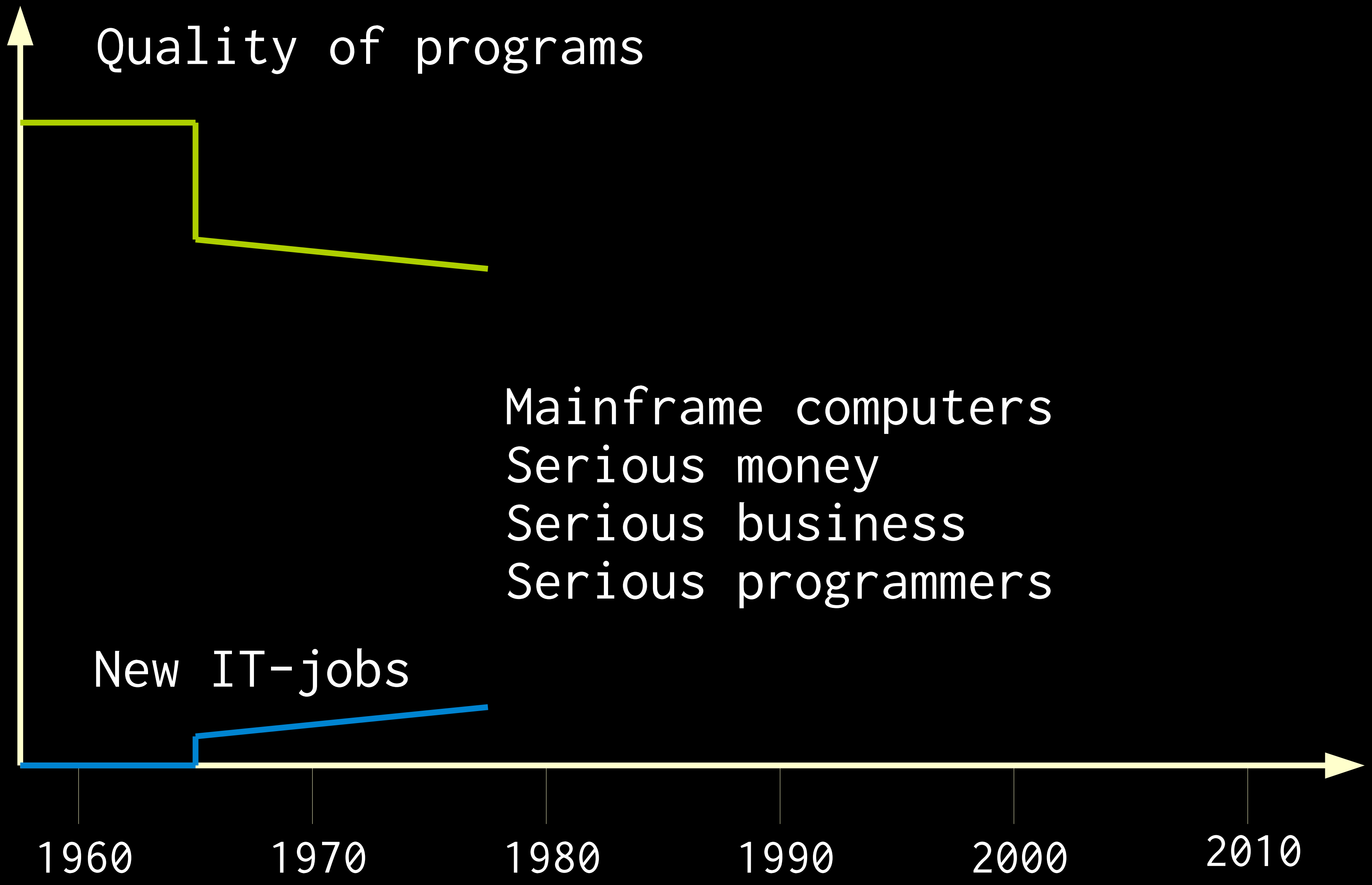
No credible theory that ”solution” would actually fix ”problems”

What happened ?

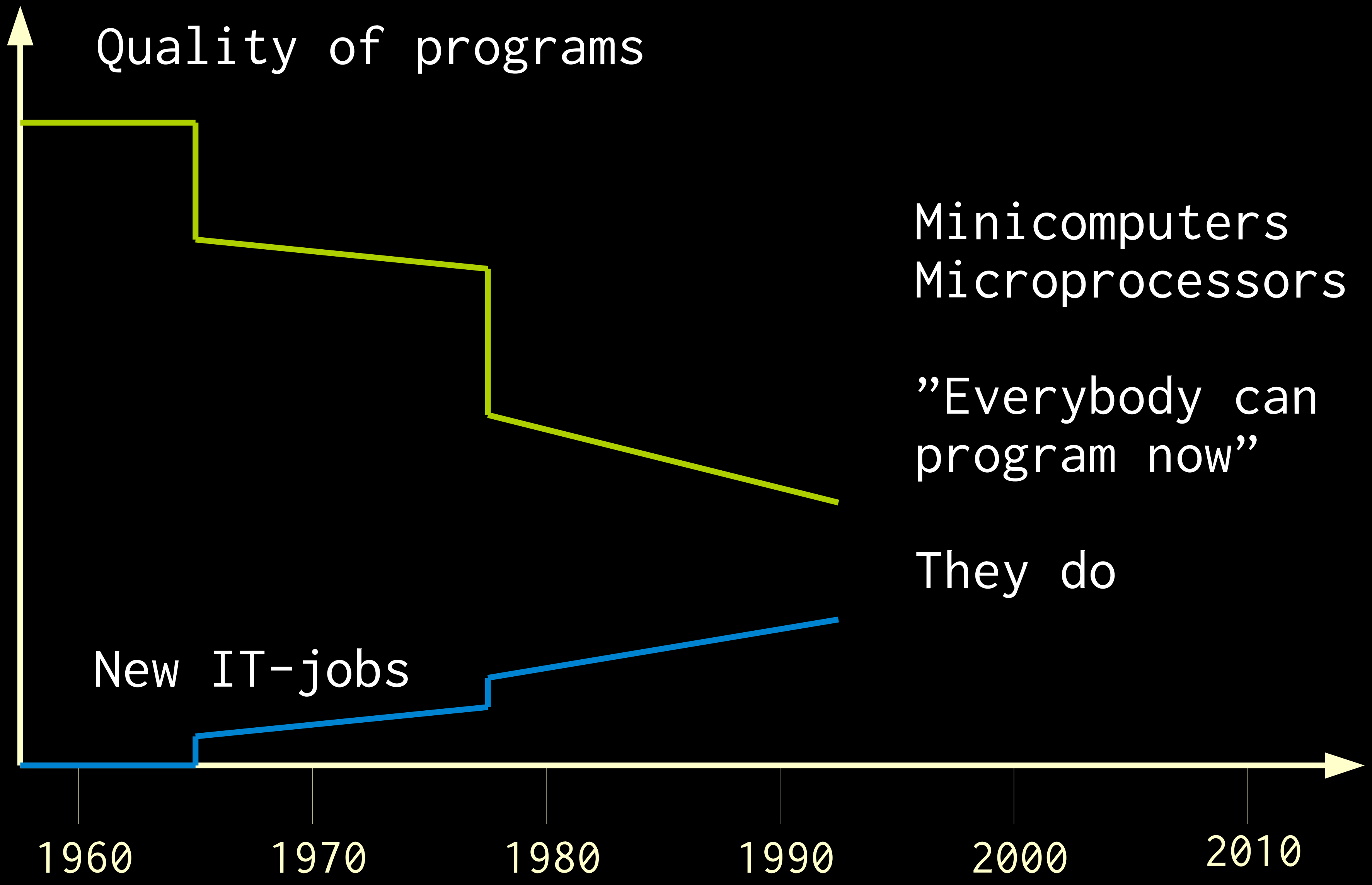


< 100 computers in Europe
Very few people program, most phds.
Write programs carefully

What happened ?



What happened ?

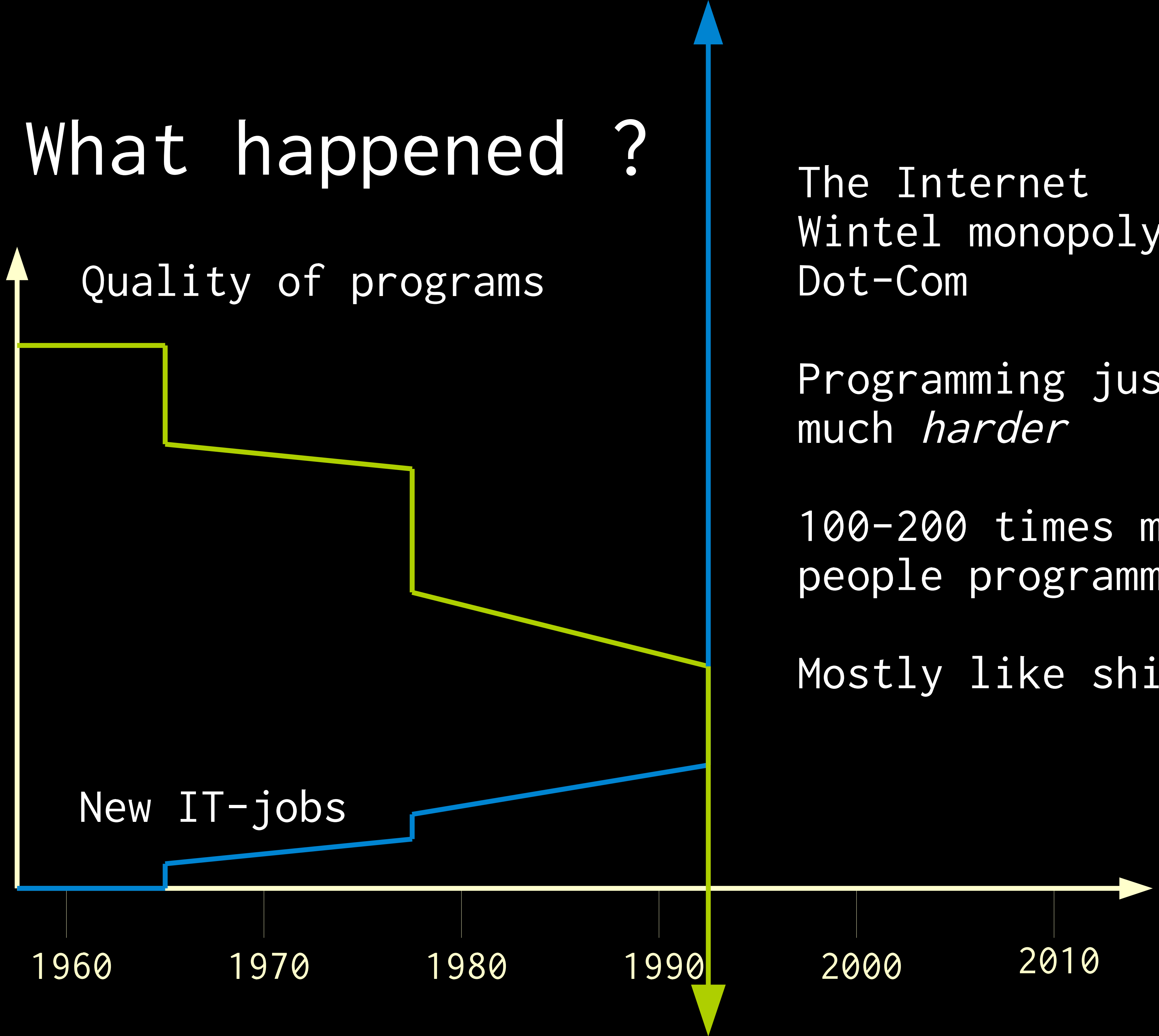


Minicomputers
Microprocessors

”Everybody can program now”

They do

What happened ?



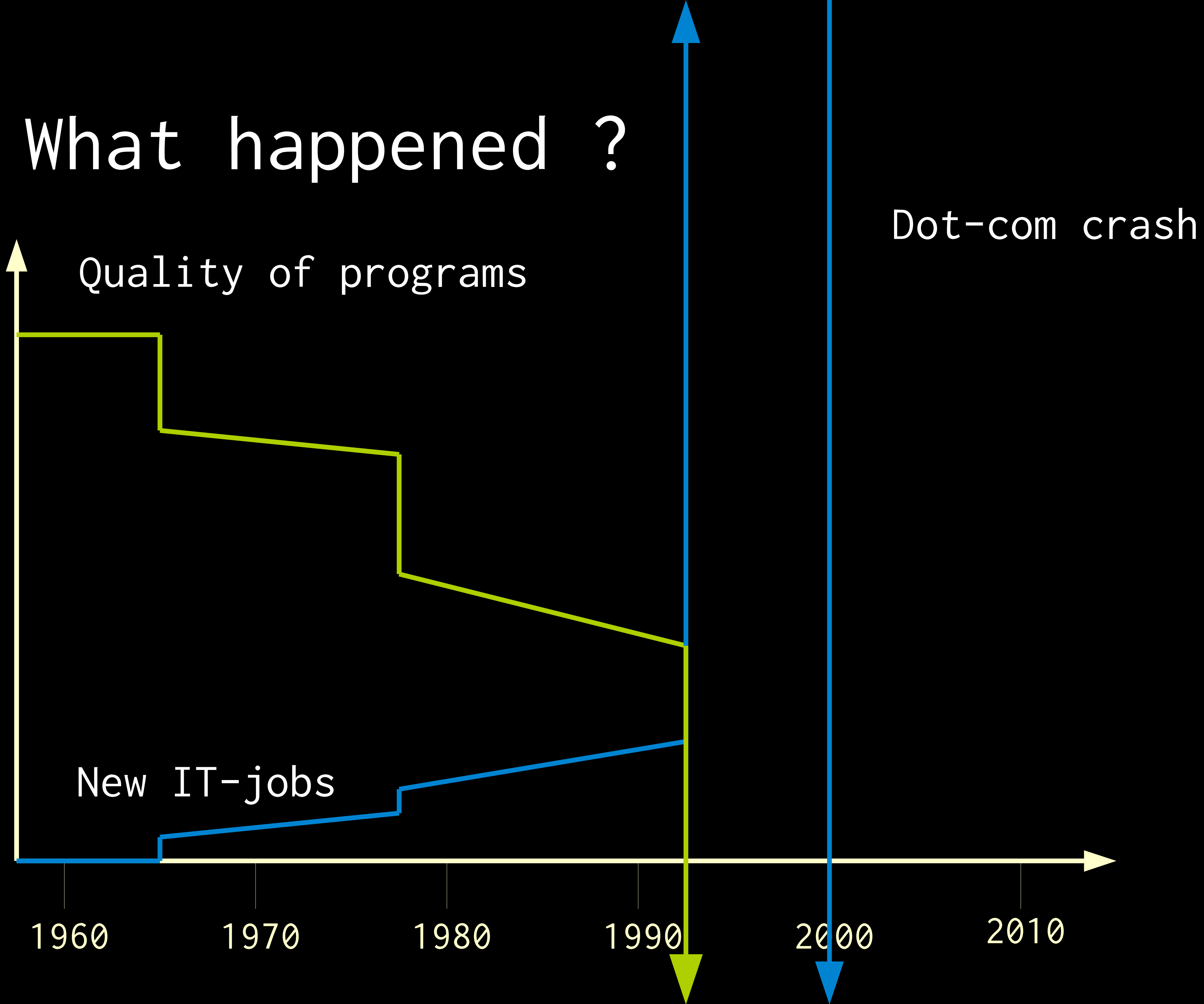
The Internet
Wintel monopoly
Dot-Com

Programming just got much *harder*

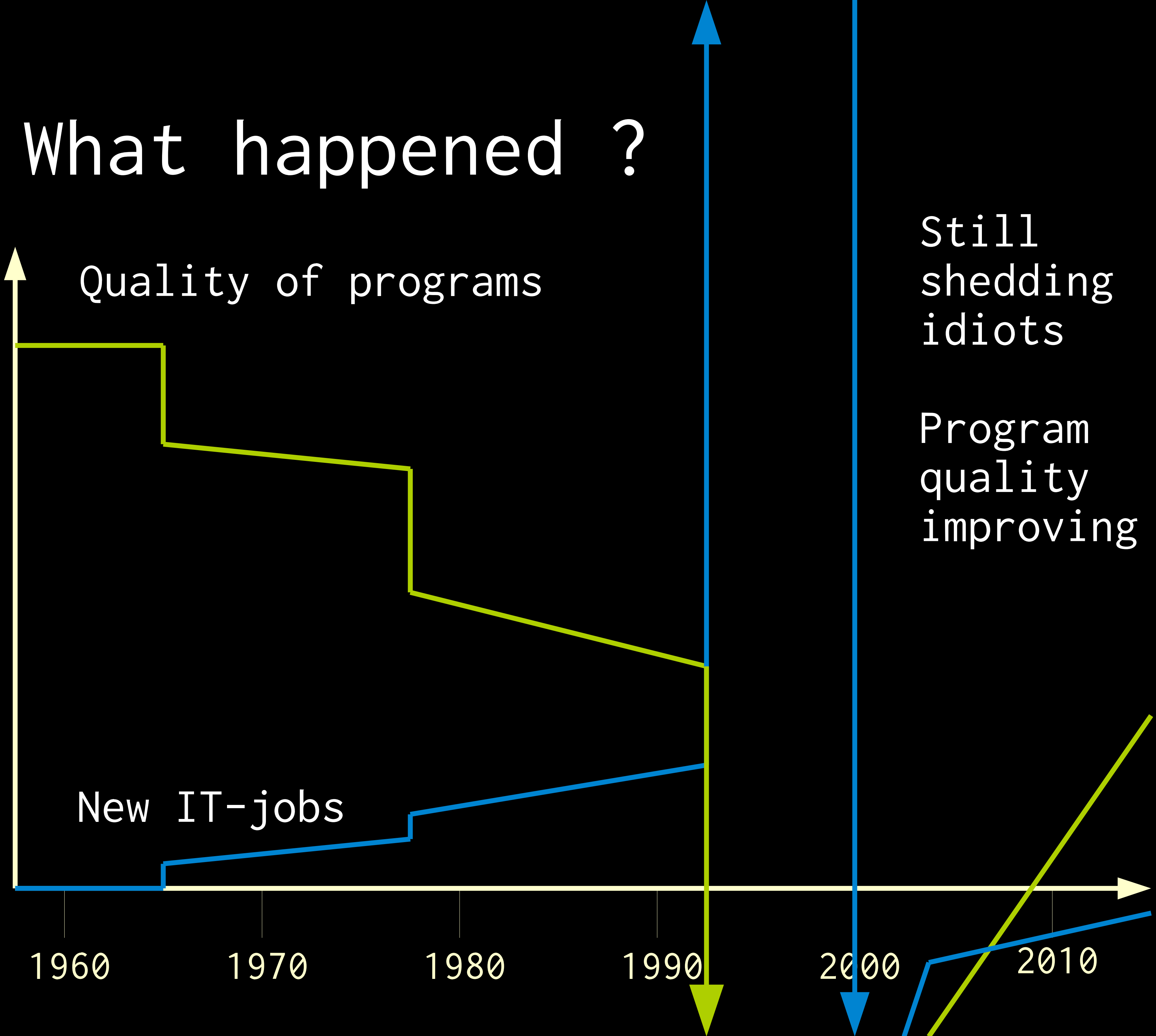
100-200 times more people programming

Mostly like shit

What happened ?



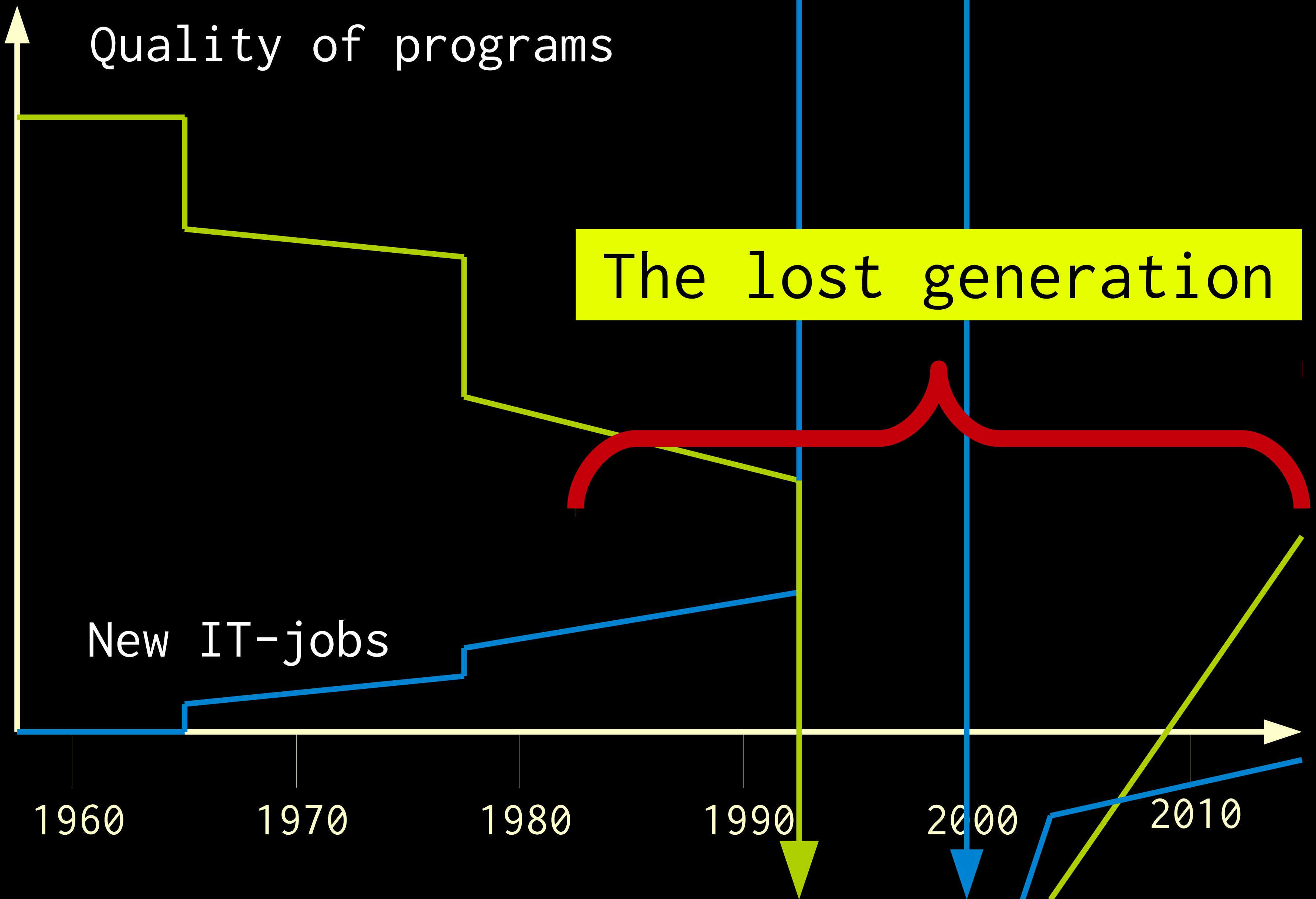
What happened ?



Still
shedding
idiots

Program
quality
improving

What happened ?

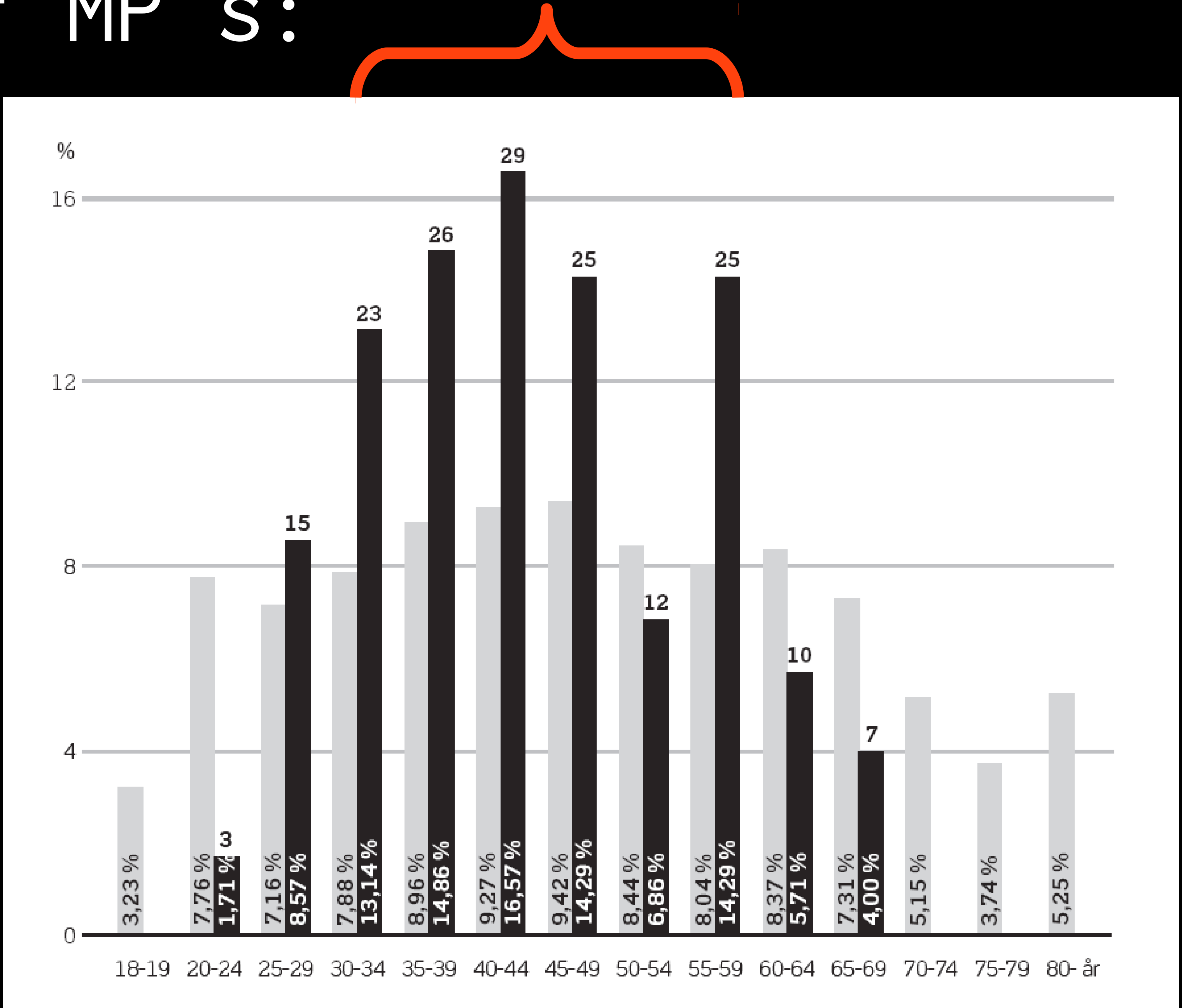


The lost generation

For the lost generation…

- ”Recovery strategy” means ”reboot”
- Printer drivers can crash the OS
- ”Security” comes from virus-scans
- “Programming” == Visual Basic/Excel Macros
- Spreadsheets are (also) databases
- And it all works as if by magic…

Age of MP's:


%
16
12
8
4
0
3,23 %
7,76 %
3
1,71 %
7,16 %
15
8,57 %
7,88 %
23
13,14 %
8,96 %
26
14,86 %
9,27 %
29
16,57 %
9,42 %
25
14,29 %
8,44 %
12
6,86 %
8,04 %
25
14,29 %
8,37 %
10
5,71 %
7,31 %
7
4,00 %
5,15 %
3,74 %
5,25 %
18-19
20-24
25-29
30-34
35-39
40-44
45-49
50-54
55-59
60-64
65-69
70-74
75-79
80- år

For the lost generation…

```
10 DIM A$(20)
20 READ A$
30 PRINT A$,”is going to solve all our problems”
40 WAIT 63115200
50 GOTO 20
100 REM XXX: Incomplete list
110 DATA “Win95”, “Win2000”, “Java”, “.NET”, “WinXP”
120 DATA “MySQL”, “PERL”, “LAMP”, “Scripting”, “XML”
130 DATA “SAS”, “C#”, “UML”, “Cloud”, “SMP”, “IPv6”
140 DATA “Prototyping”, “F#”, “AJAX”, “Agile”, “XP”
150 DATA “DevOps”, “AntiPatterns”, “KanBan”, “NoSQL”
160 DATA “UX”
```

…hope springs eternal

```
10 DIM A$(20)
20 READ A$
30 PRINT A$,”is going to solve all our problems”
40 WAIT 63115200
50 GOTO 20
100 REM XXX: Incomplete list
110 DATA “COBOL”, “FORTRAN”, “ALGOL”, “SIMULA”
120 DATA “BASIC”, “PASCAL”, “SQL”, “Waterfall”
130 DATA “AI”, “Yourdon”, “PL/1”, “RDBMS”, “CASE”
140 DATA “4GL”, “UNIX”, “C++”, “Win95”, “Win2000”
150 DATA “Java”, “.NET”, “WinXP”, “MySQL”, “PERL”
160 DATA “LAMP”, “Scripting”, “XML”, “SAS”, “C#”
170 DATA “UML”, “Cloud”, “SMP”, “IPv6”, “Prototyping”
180 DATA “F#”, “AJAX”, “Agile”, “XP”, “DevOps”
190 DATA “AntiPatterns”, “KanBan”, “NoSQL”, “UX”
```

How to start an IT-fashion

- Get random idea
- Give it a edgy & hip name
- Try it with a small group of talented people
- Write book, claim it solves all problems
- Go on speaking/course circuit to flog book
- Goto step 1 before they catch on to you.

There is no Silver Bullet

- 1987 paper by Frederic P. Brooks
- Who wrote “The Mythical Man-Month” in 1975

…Which, amongst much other wisdom, said you can arrange your dev-team any f**king way you want, as long as it has lots of talent.

Why there is no Silver Bullet:

- Dijkstra, IFIP congress 1965:

“Programming considered as a human activity”

“I have only a very small head and must live with it.”

Computing is hard!

- Complexity 1000 times higher than stuff

- Air Craft Carrier CVN-78 “Gerald Ford”
  1 mio parts, total.

- FreeBSD Kernel
  1.8 mio lines of code

- F22 “The Final Fighter”
  2 mio lines of code

- F35 “JSF” fighter
  8 mio lines of code
  (estimated, only 50% compl.)

Code is buggy

- “Space shuttle Quality”

  0.1 bug per 1000 lines of code

- “High Quality”

  1 bug per 1000 lines of code

- “Release quality”

  15-50 bugs per 1000 lines of code

Neoclassic computing ?

We are slowly relearning what we knew pre 1990:

- Programming is complex and difficult
- Best shot: Intentional programming
- Craft & Techniques
- Think/Architect/Design before prototyping
- Always throw the prototype away
- Don’t fuck with important stuff if it works

A long uphill battle:

- Banish IT-homoeopathy / Cargo-Culting
    - Use Scientific Tools
    - Transparency
    - Accountability
- Investigate & Publish ”Lessons Learned”
- Official ”IT-disaster investigators” ?

Those[1] who don't learn from history are doomed to repeat it

[1] A.k.a: IT-professionals