

# Who can (not) call him/herself a SW engineer?

Novica Nosović, ETF, Sarajevo

16<sup>th</sup> DAAD Workshop “Software Engineering Education and Reverse Engineering”

Jahorina, Bosnia and Herzegovina, 21<sup>st</sup> – 27<sup>th</sup> August 2016

# Content

Motivation

SW engineering by definition

Local SW job market taxonomy

Historical view – an approach

SWE certification – private vs. public (HE)

Conclusions... if any!

# Motivation

- Question for the audience:
  - Who can call him/herself a software engineer?

Expected answers:

- All of us
- None
- Some of us (Who/Why!?)

# SW engineering by definition

- "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software"
- "an engineering discipline that is concerned with all aspects of software production"
- ... concerned with theories, methods and tools for professional software development....
- *"A discipline that deals with the building of software systems which are so large that they are built by a team or teams of engineers."* [Ghezzi, Jazayeri, Mandrioli]

# Webopedia on SW engineer

- A SW engineer is a **licensed professional engineer** who is schooled and skilled in the application of engineering discipline to the creation of SW.
- A SW engineer is often confused with a **programmer**, but the two are vastly different disciplines. While a programmer creates the **codes** that make a program run, a SW engineer creates the **designs** the programmer implements.
- By U.S. law no person may use the title "engineer" (of any type) unless the person holds a professional engineering license from a state licensing board and are in good standing. A SW engineer is also held accountable to a specific code of ethics.































My student – Software Engineer!

# ... and more SW engineers...!

Who We Are

Our people and their extraordinary individual experiences are Maestral.

Our creativity, ideas and experiences are the driving force behind Maestral. We encourage outside-the-box thinking that engages visitors. Our clients appreciate the value that our bring to each project we create. We're always looking for new ways to serve our clients better, for the simple reason that we love what we do.

 Abdurahman Software Engineer	 Amel Software Engineer	 Uzeir QA Engineer	 Ali Software Engineer
 Dino QA Engineer	 Vidana UX/UI Engineer	 Anita Software Engineer	 Merced General Manager - Santiago Office
 Matej Software Engineer	 Sulejman Chief Technical Officer	 Herman Lead Software Engineer	 Benjamin UX/UI Engineer
 Dzenan QA Engineer	 Mustafa Software Engineer	 Hermina Product Manager	 Selma Software Engineer
 Markus Software Engineer	 Ashraf UX/UI Engineer	 Vera Software Engineer	 Haris Office Assistant
 Salih Software Engineer	 Mehmed Software Engineer	 Irfan UX/UI Engineer	 Emir Software Engineer
 Edin Software Engineer	 Edvin Chief Operating Officer	 Vedran Software Engineer	 Tarik Lead UX Engineer

# Reaction from my colleagues:

“So what? That is normal!”

“That is the company`s internal policy!”

“Why not? They are developing SW!”

...



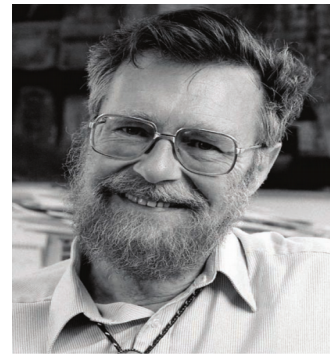
# Edsger Dijkstra (1930–2002)



When he got married in 1957, he could not enter the term “programmer” into his marriage record! (“programmer” became recognized in the early 1960s)

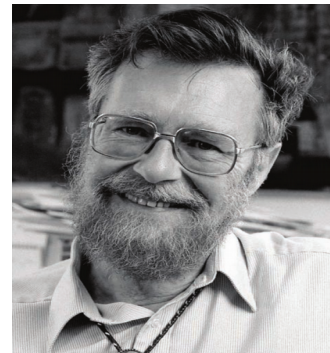
“At the time programming didn’t look like doing science; it was just a mixture of being ingenious and being accurate.”

# Mathematicians vs. Engineers



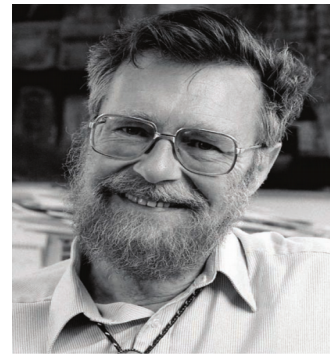
“During 1968 in Garmisch I learned that in the ears of the Americans, a “mathematical engineer” [such as we educated in Eindhoven] was a contradiction in terms: the American mathematician is an impractical academic, whereas the American engineer is practical but hardly academically trained.”

# Early etical problems



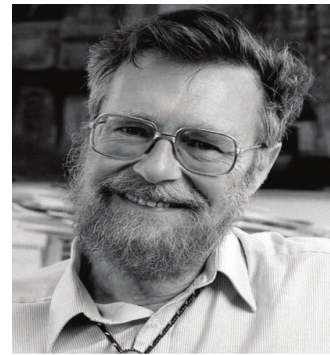
- 1964 was the year in which IBM announced the 360
- there were serious flaws built into the I/O organization of that machine - but that was clearly a part of the IBM culture
- Russian decision to build a bit-compatible copy of the IBM 360 is the greatest American victory in the Cold War

# Europe vs. U.S.?



- “One of the things that saved Europe was that until 1960 or so, it was not considered an interesting market. So we were ignored. We were spared the pressure. I had no idea of the power of large companies. Only recently I learned that in constant dollars the development of the IBM 360 has been more expensive than the Manhattan Project.”

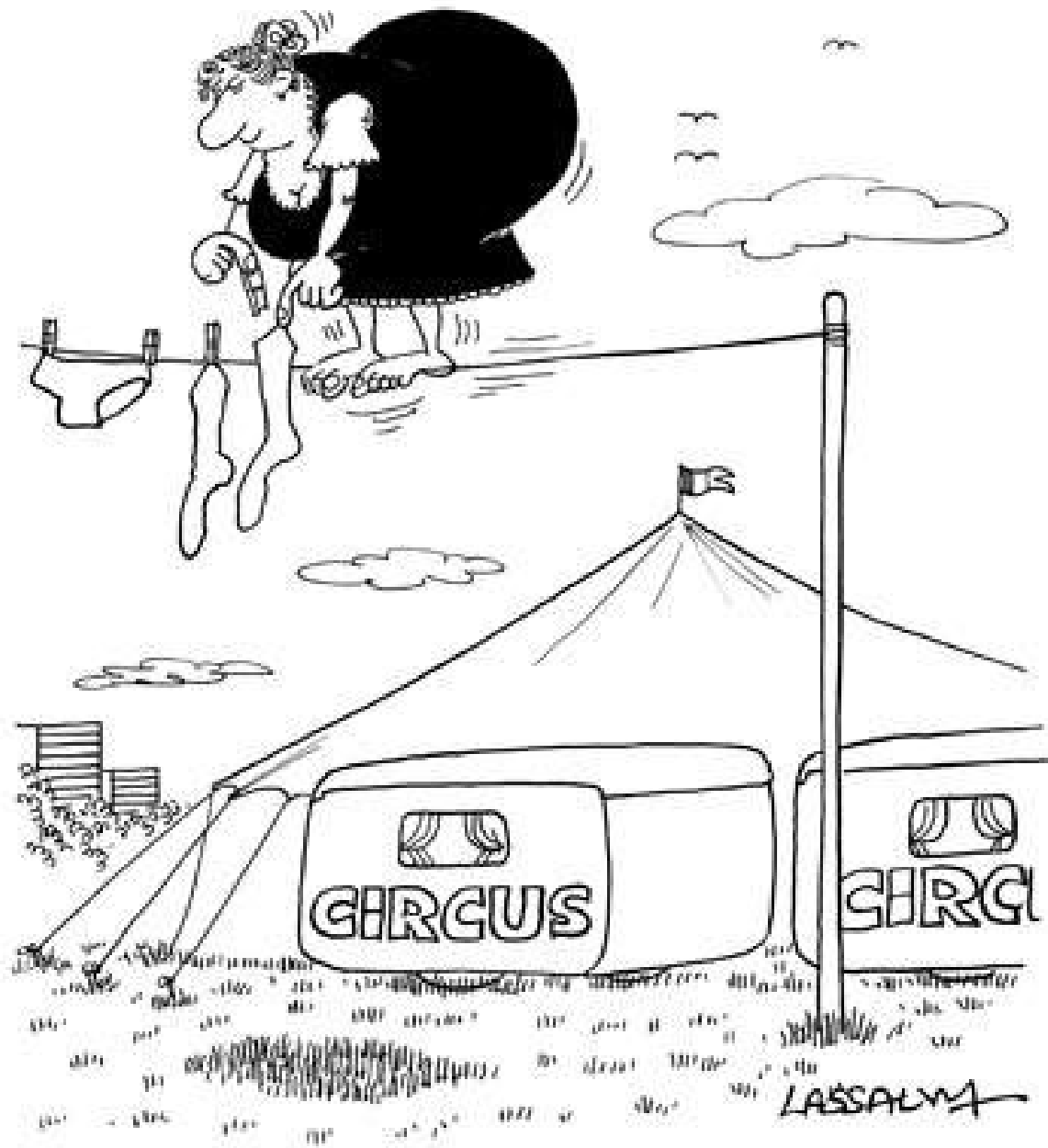
# CS often come out of EE in the U.S., not in Europe?

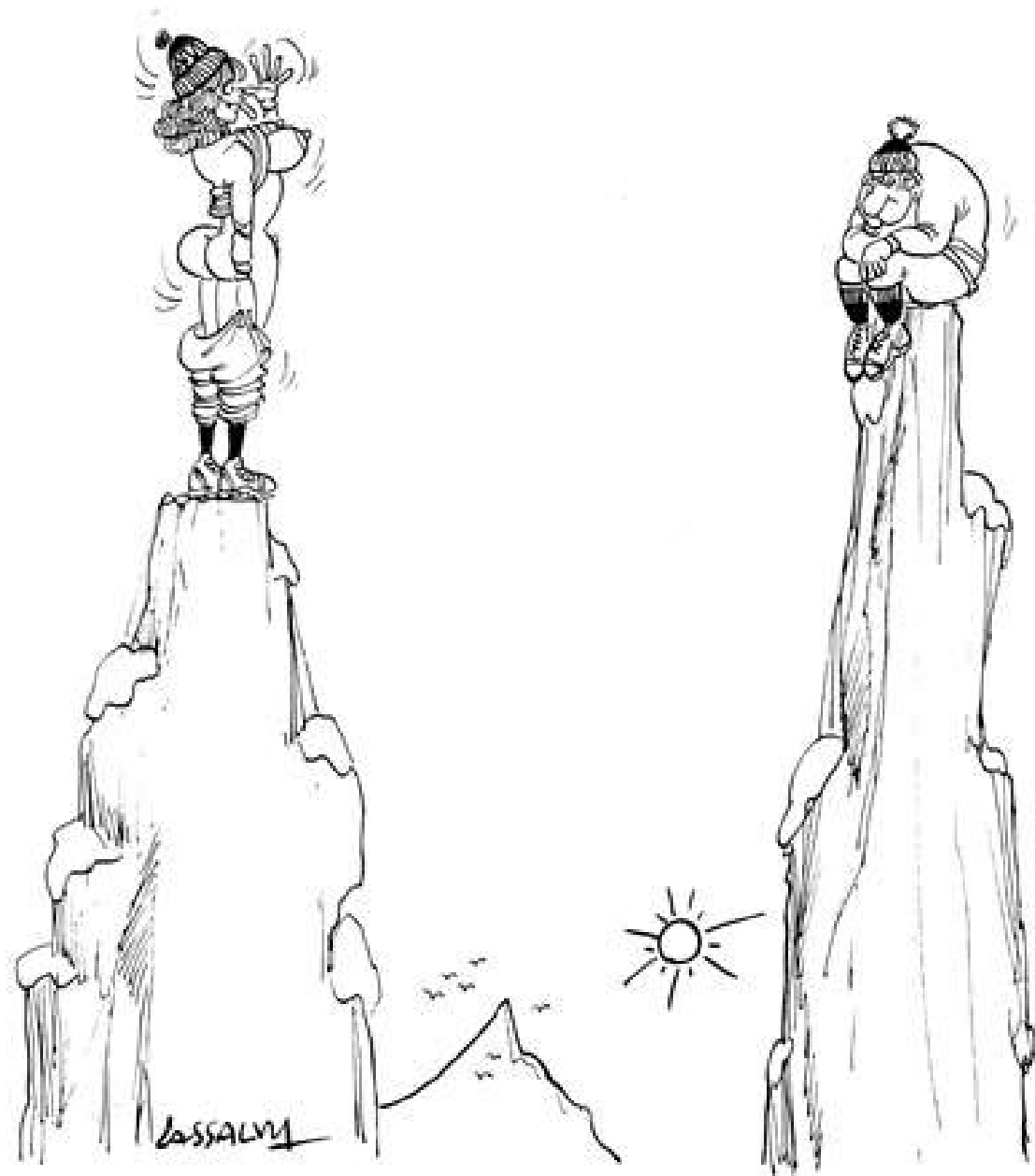


- A major reason is timing. For financial reasons, Europe, damaged by WWII, was later. So the American computing industry emerged earlier. The computing industry asked for graduates, which increased the pressure on the universities to supply them, even if the university did not quite know how.
- Association for Computing *Machinery*
- British *Computer Society*
- Computer Science, rather than *computing* science or *computation*. Europe was later, it coined the term Informatics.
- French pushed *informatique*
- English prefer Information Technology, IT, and Information Systems, IS
- University of Texas, still has Department of Computer Sciences

# SW Engineers vs. Programmers

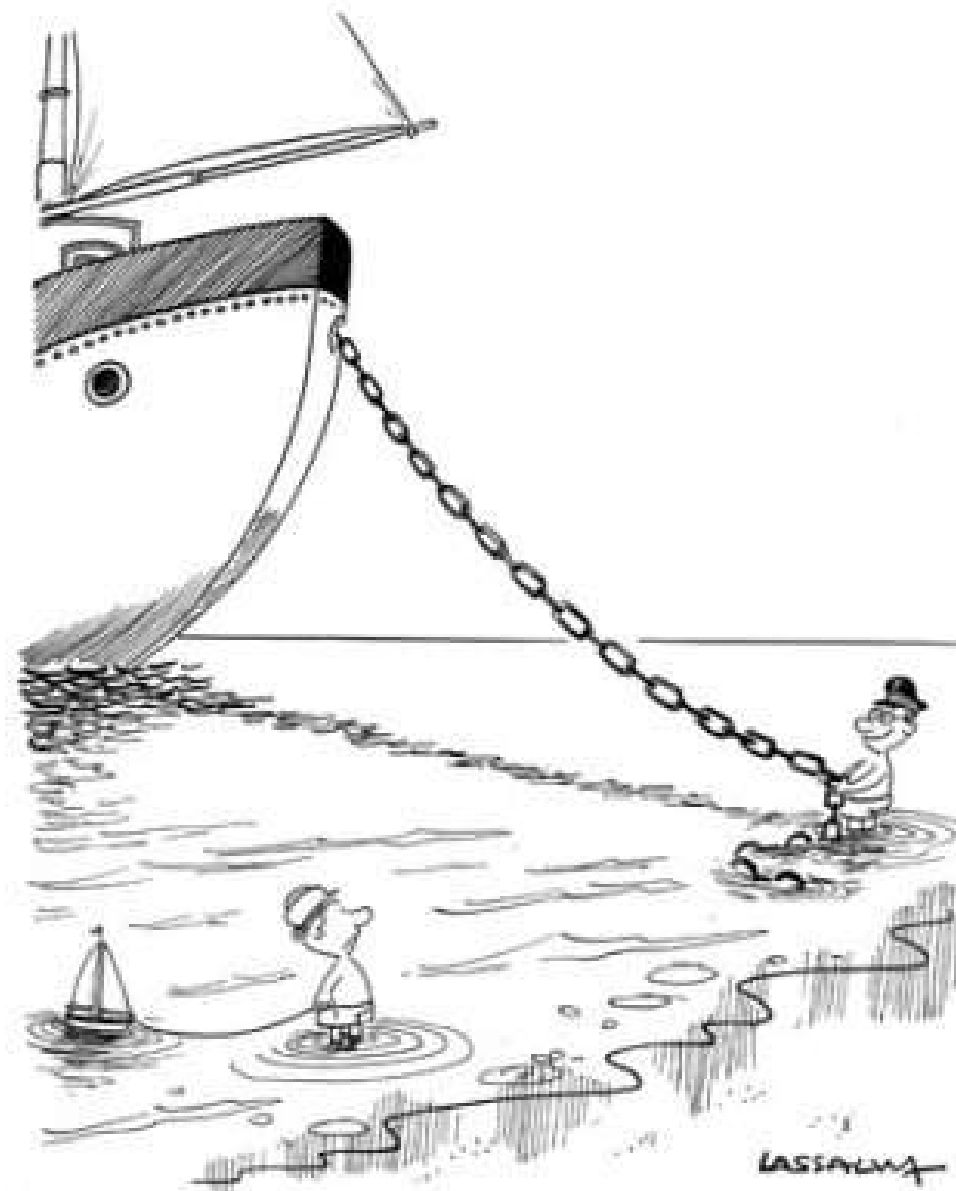






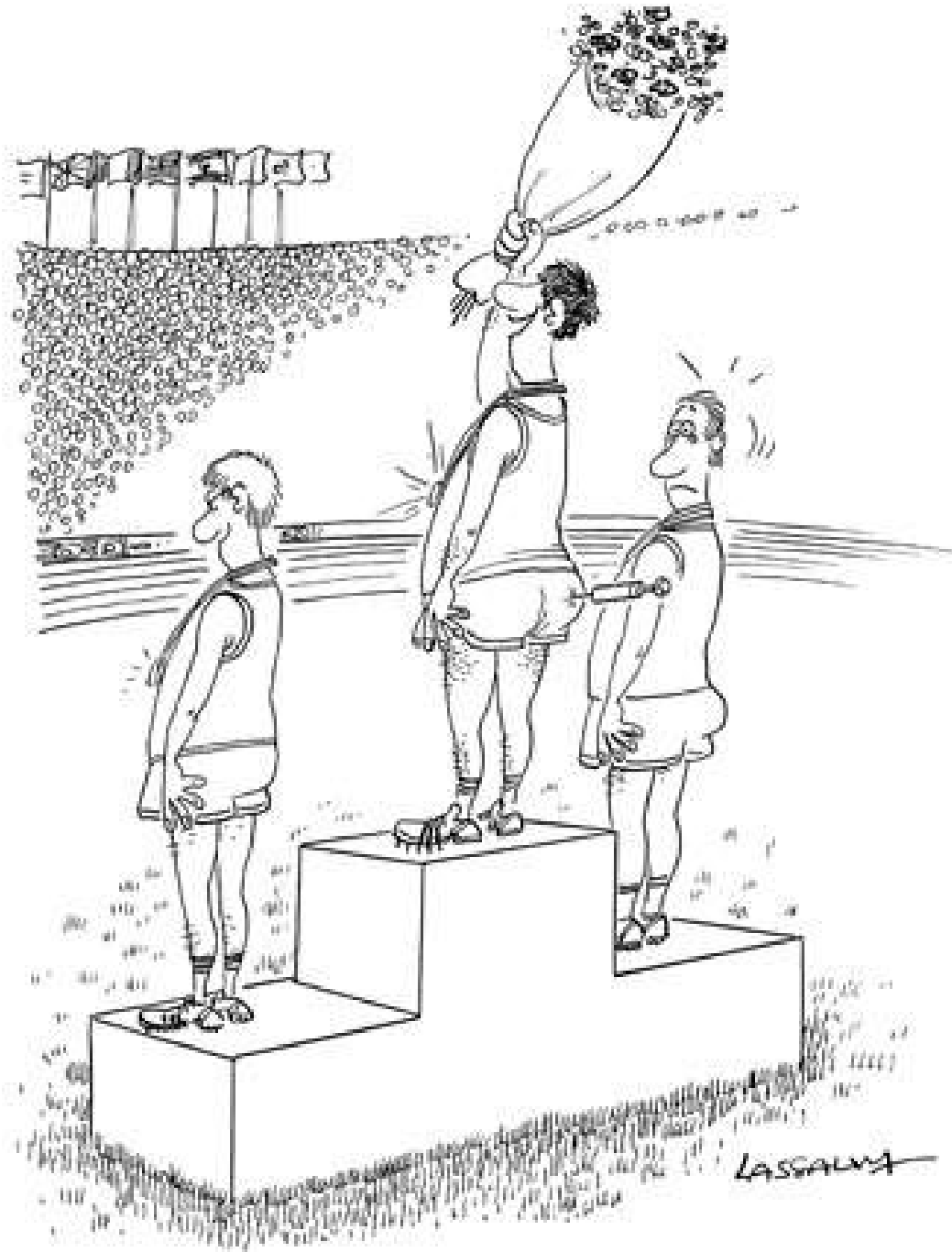


# Private vs. Public HE

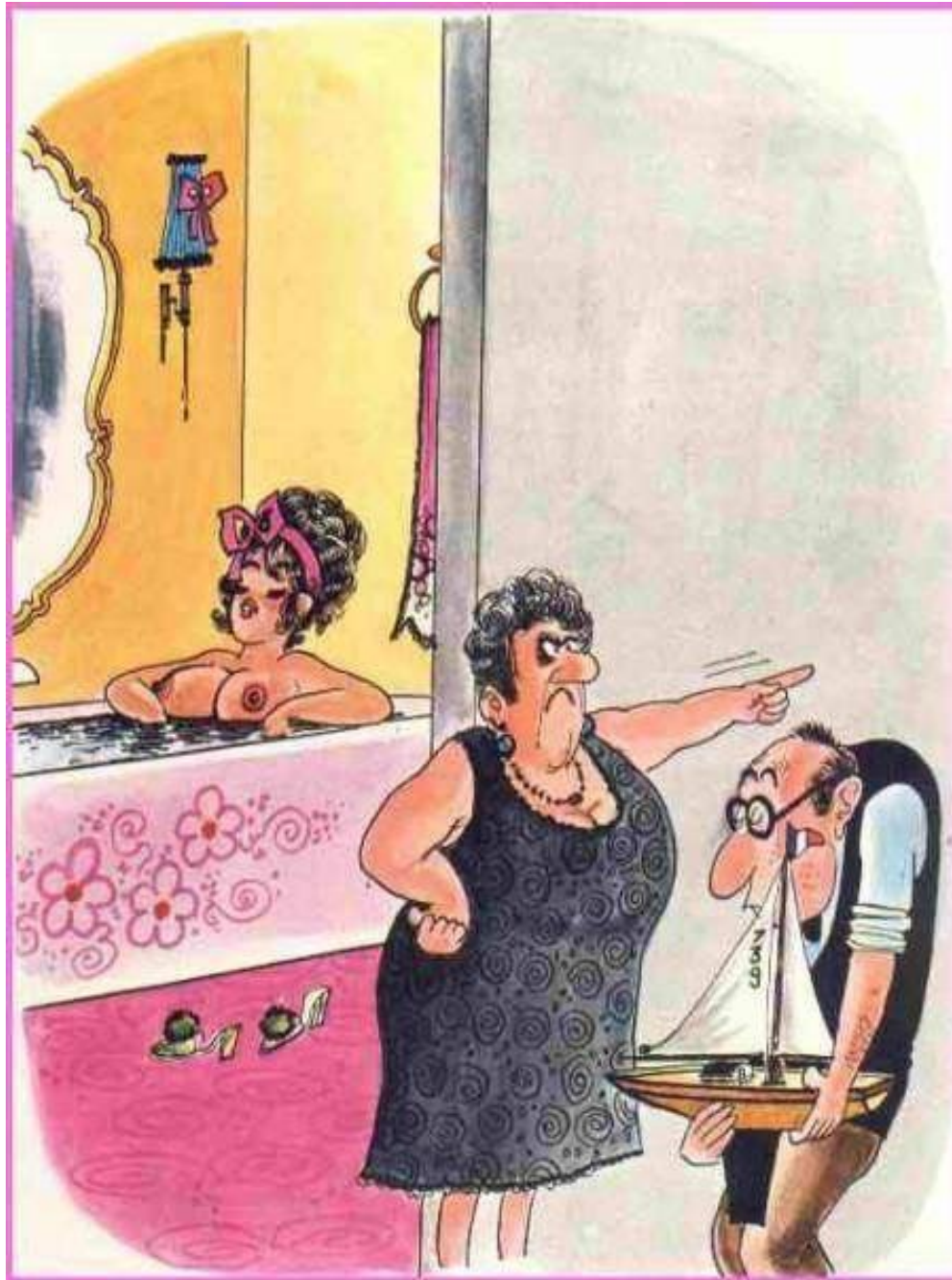


# Private vs. Public HE (2)









# Conclusions... if any!



