No Silver Bullet – Essence and Accident in Software Engineering

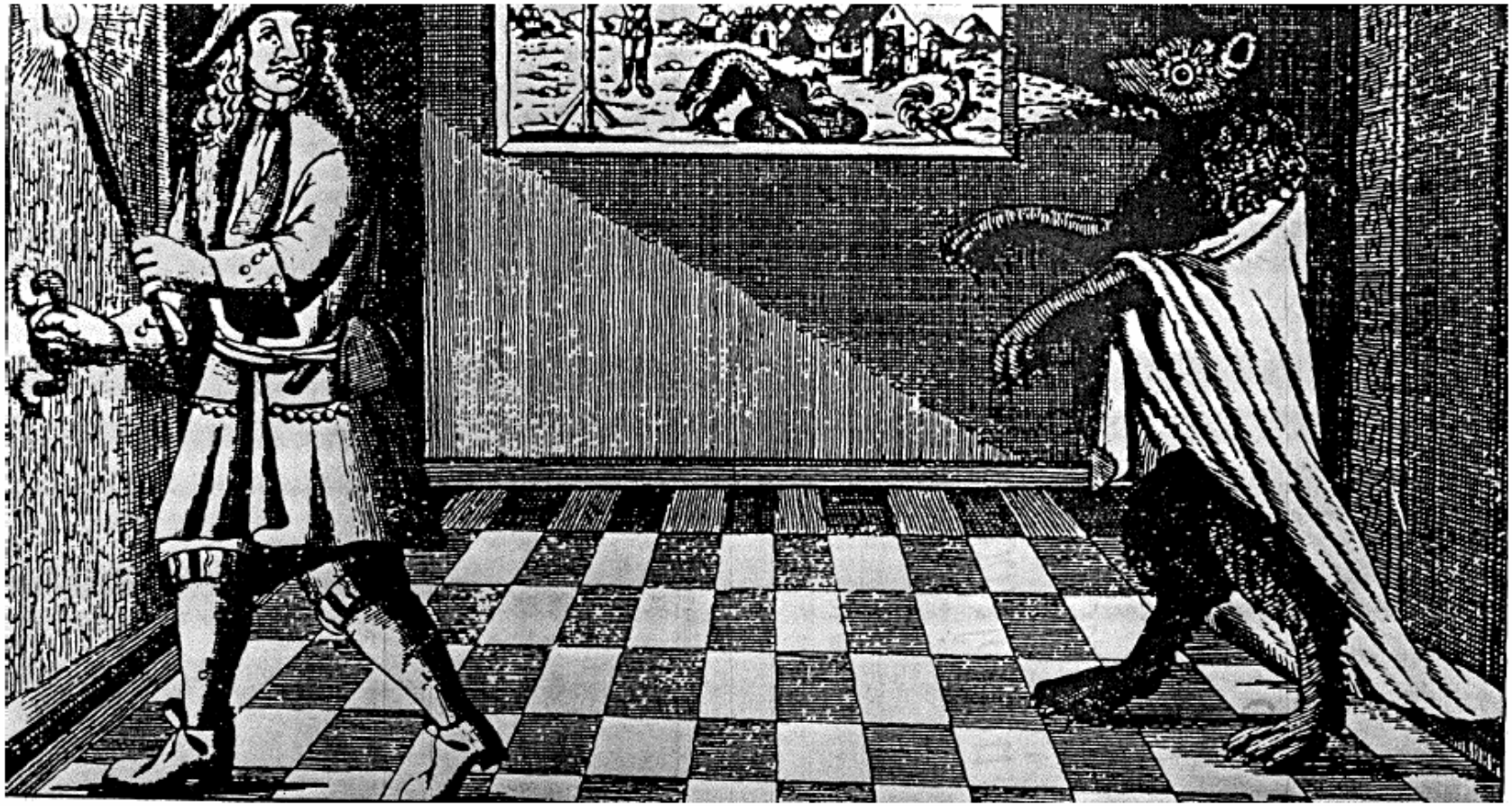
פיתוח מערכות תוכנה מבוססות Java

אוהד ברזילי
אוניברסיטת תל אביב

חלק השקפים הבאים לקוחים ממצגת של יורי ארביטמן ממכון ויצמן למדע

- *“There is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity”* (1986).

Silver bullet



No Silver Bullet – Essence and Accident in Software Engineering

- Silver bullet: a way to defeat werewolves.
  - Generally [in folklore]: any straightforward solution perceived to have extreme effectiveness.

- Compares software to hardware:
  - The anomality is not that software progress is so slow, but that computer hardware progress is so fast.

- ***Essence***—the difficulties inherent in the nature of the software
- ***Accidents***—those difficulties that today attend its production but that are not inherent [but *incidental*].

Essence Difficulties

- **Essence:**
  - Complexity
  - Conformity
  - Changeability
  - Invisibility

Essence Difficulties

- ***Complexity***
  - enormous number of states (orders of magnitude more than in hardware), so conceiving, describing and testing is hard
  - increases non-linearly with its size
  - introduces a lot of difficulties:
    - communication among team members
    - enumerating (much less understanding) of all possible states of the program
    - management problems:
      - conceptual integrity is hard to achieve
      - learning curve: personnel turnover becomes disaster
    - others

Essence Difficulties

- ***Conformity***
  - Physics example: looking for simplicity in complex structures
  - Software: the complexity is arbitrary, forced by existing systems to which the interfaces must conform.
    - cannot be simplified by any redesign!

Essence Difficulties

- ***Changeability***
  - Software is constantly under pressure for change, partly because it can be changed more easily than a building.
  - Two processes are at work:
    - Demand for extended function (a result of success)
    - Suitability for a new hardware is needed

Essence Difficulties

- ***Invisibility***
  - Unlike other disciplines, where geometric abstractions serve as a powerful tool, software is not inherently embedded in space
  - Several general directed graphs, superimposed one upon another appear while trying to create a representation
    - the graphs are nor planar neither hierarchical

Hopes for the silver

- What helped to overcome some of accidental difficulties in the past?
  - High-level languages
  - Unified programming environments

- Hopes for the silver:
  - OOP:
    - Hierarchical
    - Data hiding

    Helps in design, but do not solve design complexity problem

Hopes for the silver

- AI (expert systems)
  - *May be very useful*
- AI ("Automatic programming"): generation of a program from problem specification
  - Used successfully for very specific tasks (differential equations,...)
  - Hard to imagine having a general solution
- Graphical programming:
  - No hope, for software is difficult to visualize

Hopes for the silver

- Program verification
  - Might reduce the program-testing load, not eliminate it
  - A lot of work
  - Can establish that a program meet its specification. But the hardest part is to get such complete and consistent specification!
- Better workstations, environments and tools
  - are welcomed, but magical enhacements cannot be expected

Addressing Essence

- Buy vs. Build
  - Discusses the process of wide-spread use of software “today” compared to 60-s, adopting procedures to existing software
- Requirements refinement and rapid prototyping
  - *“The hardest single part of building a software system is deciding precisely what to build”*
  - Thus, rapid prototyping tools are one of the most promising efforts that attack *the essence* of software development problem.

Addressing Essence

- Incremental development
  - Write vs. Build
  - Build vs. Grow (top-down design, stubs…)

- Great designers
  - *“The difference between the great and the average approach an order of magnitude”*
  - Gives hints as to how to grow great designers