Software Mining

Lecture 1
Introduction to
Empirical Software Engineering

Constantly Evolving Technology

Constantly Evolving Technology

CASE tools

structured design

abstract data types

maturity models

structured programming

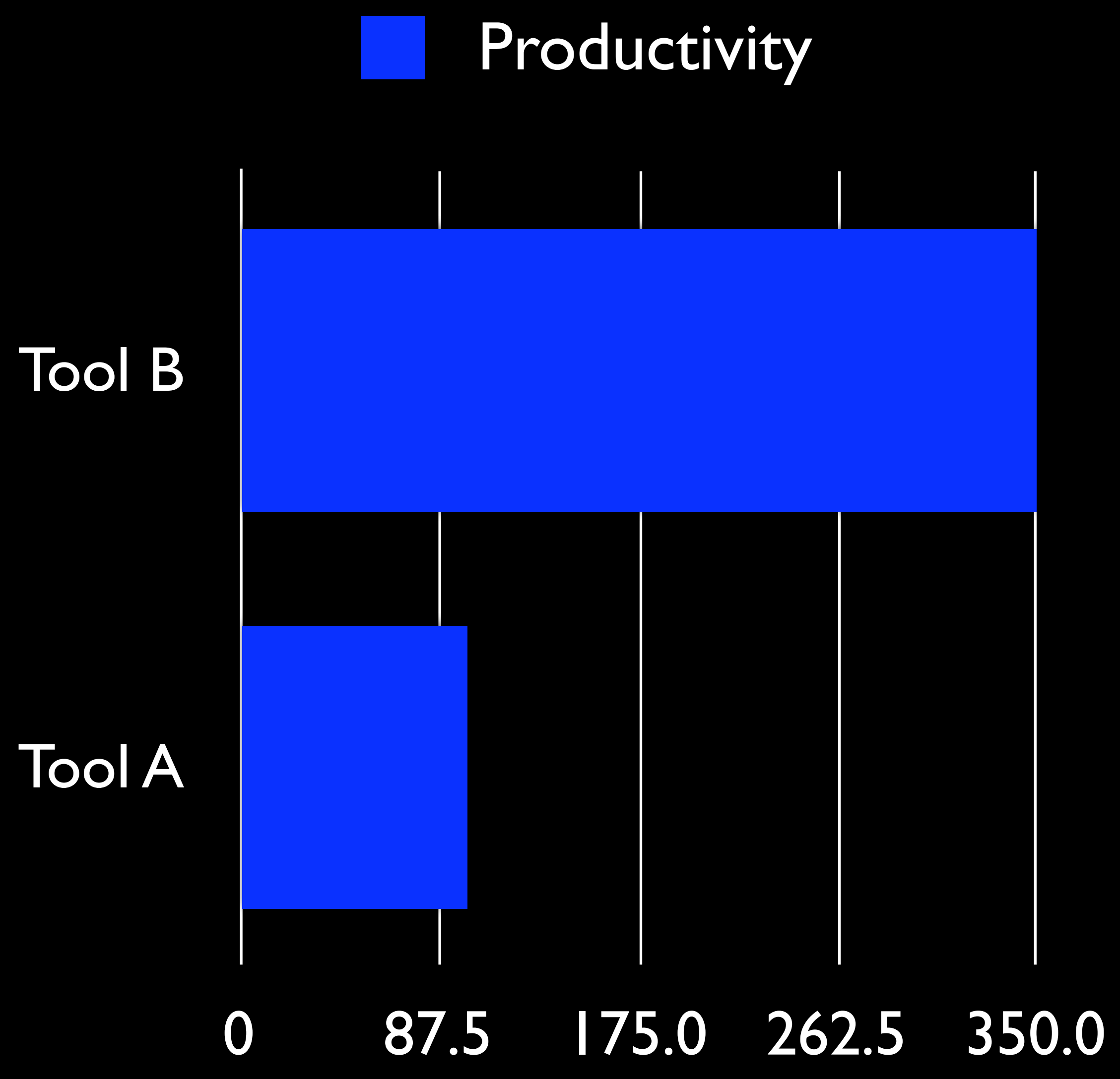
Research Claims

Research Claims

by Vendors...

Research Claims

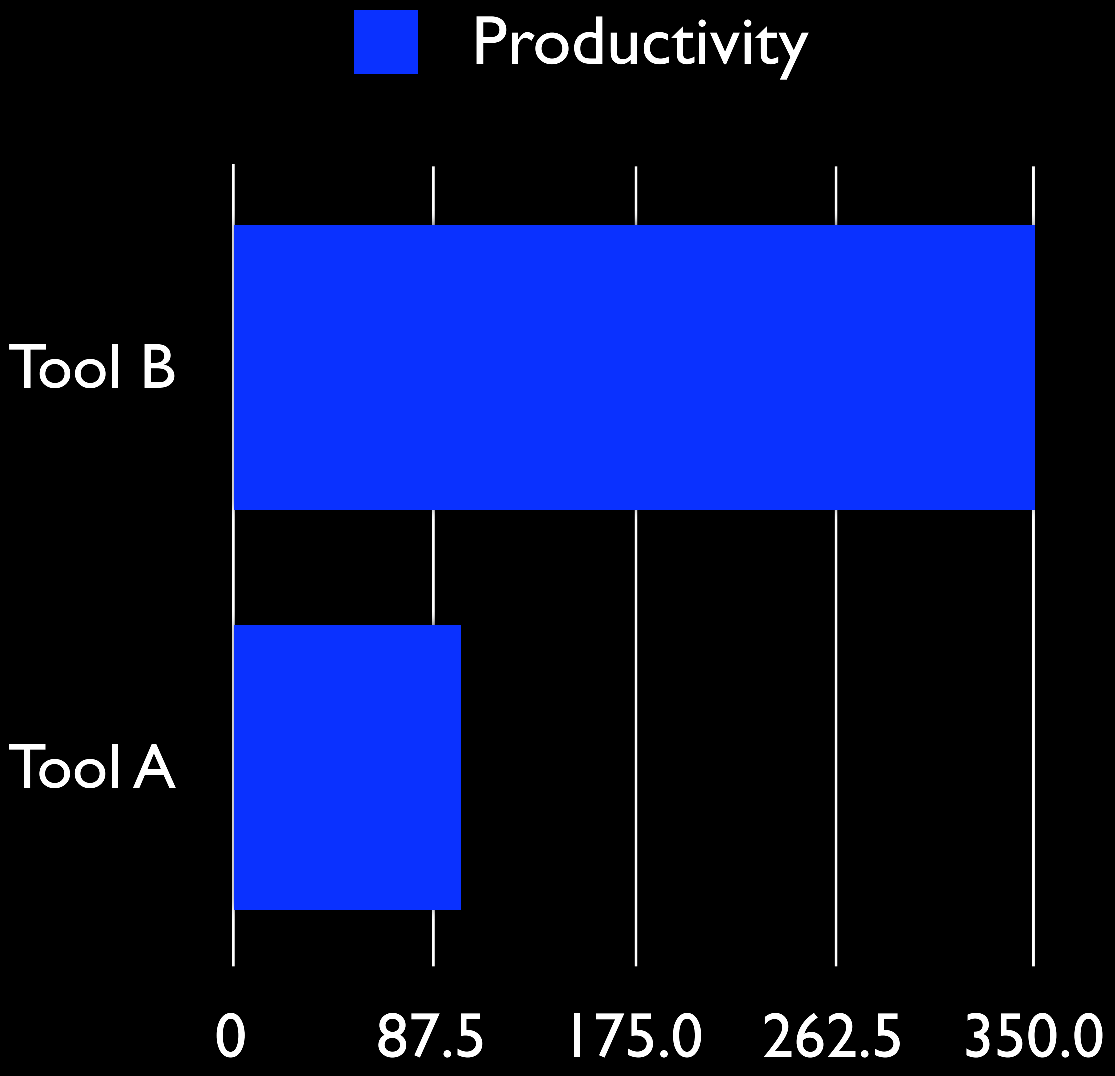
by Vendors...



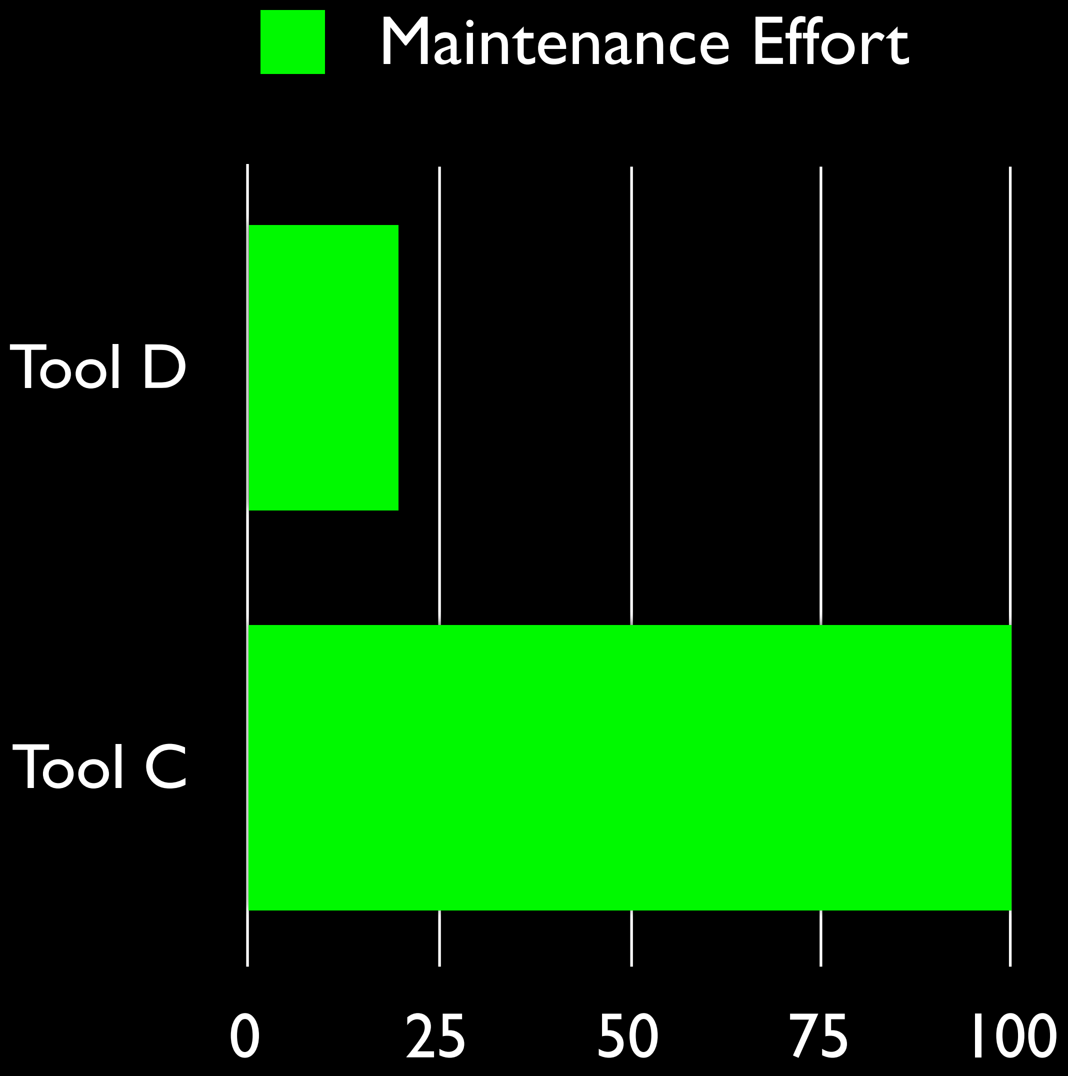
Productivity increases by 250%!

Research Claims

by Vendors...

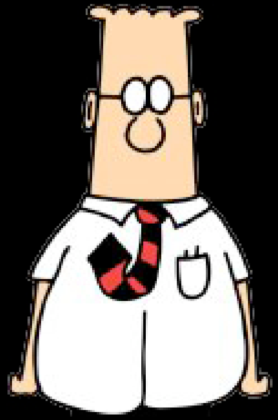


Productivity increases by 250%!

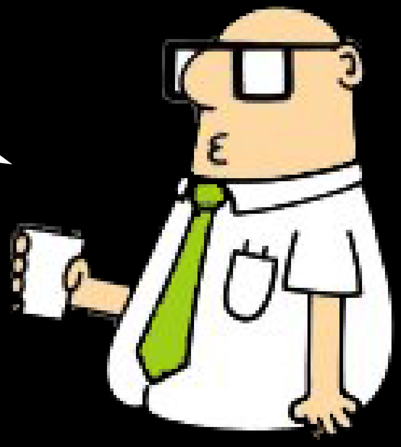


Maintenance Effort decreases by 80%!

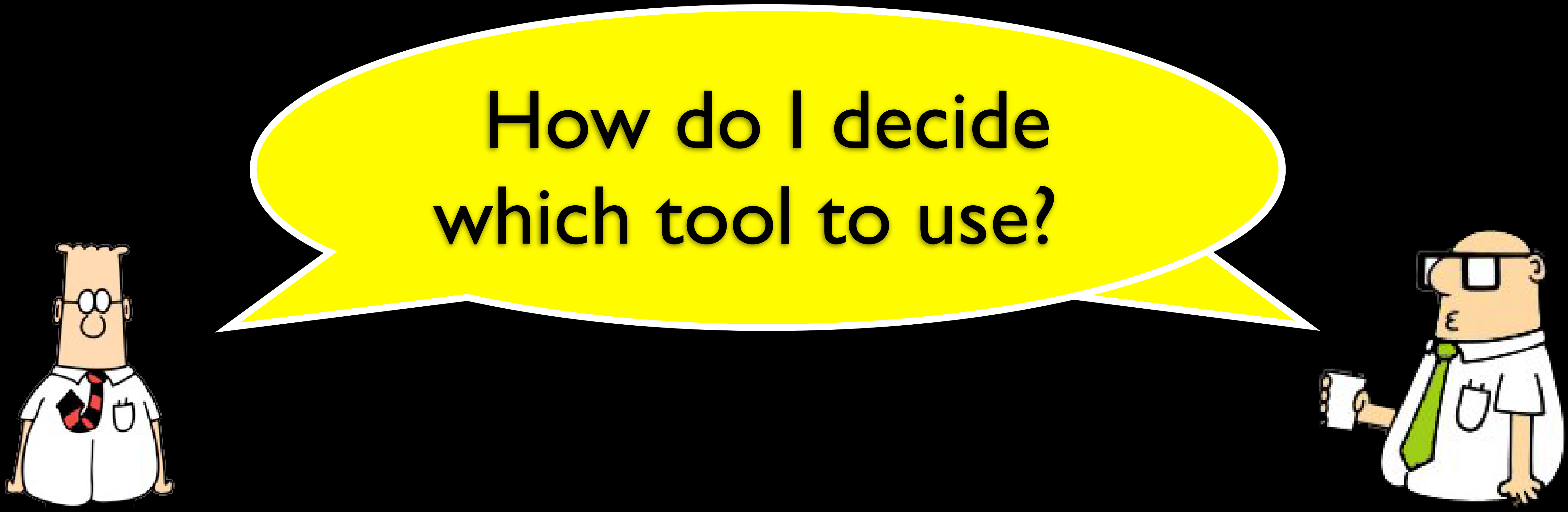
How do I decide which tool to use?



How do I decide which tool to use?



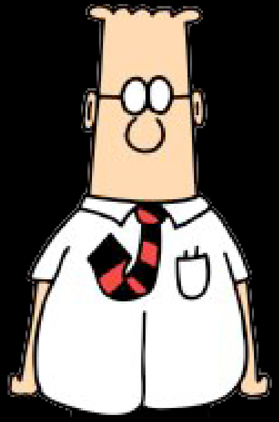
one way to find out...

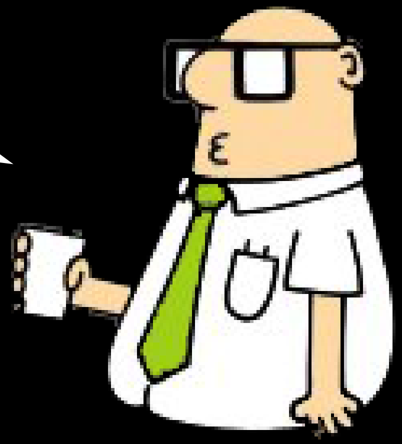


one way to find out...

Experimentation

How do I decide
which tool to use?





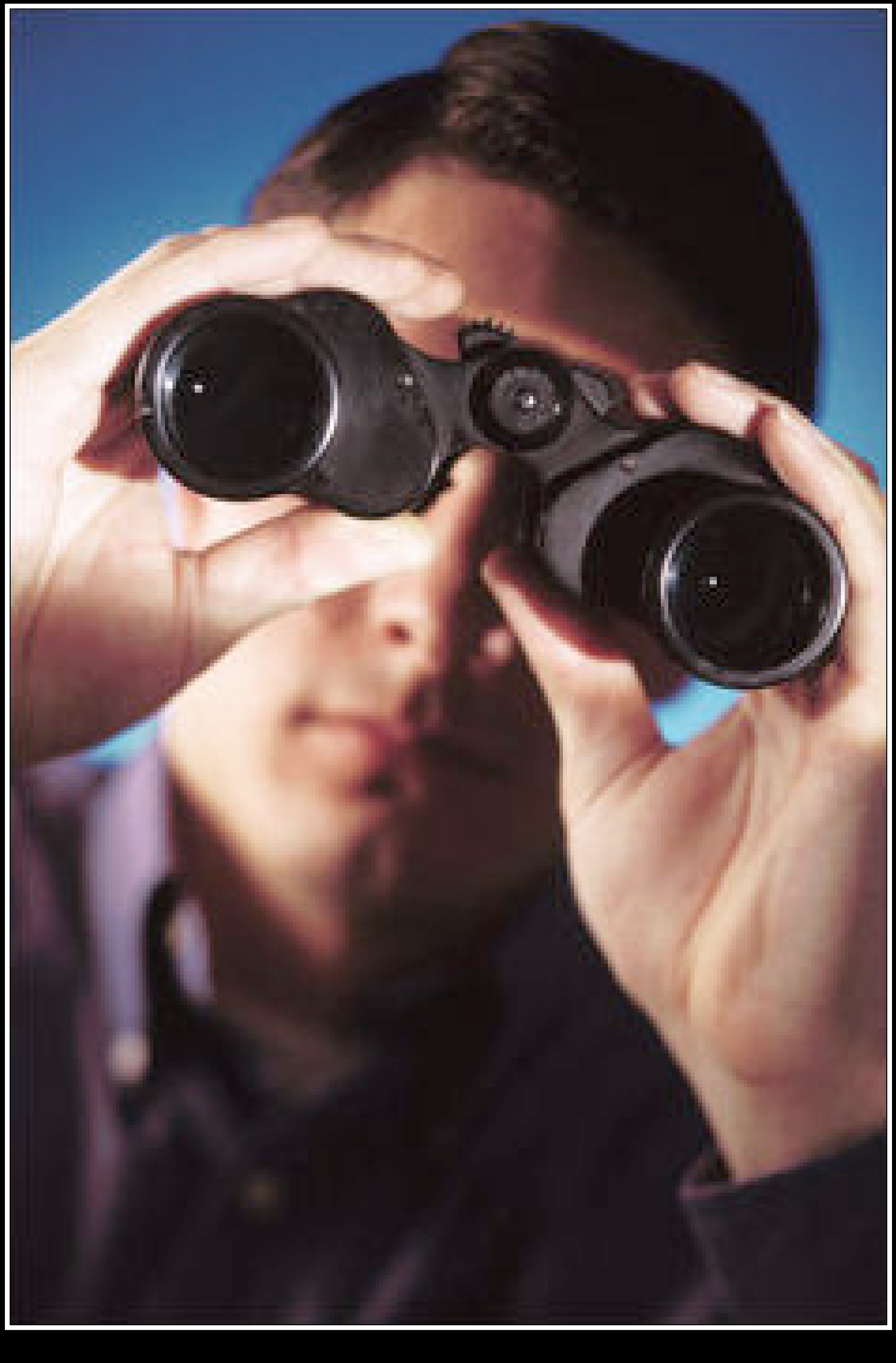
one way to find out...

Experimentation

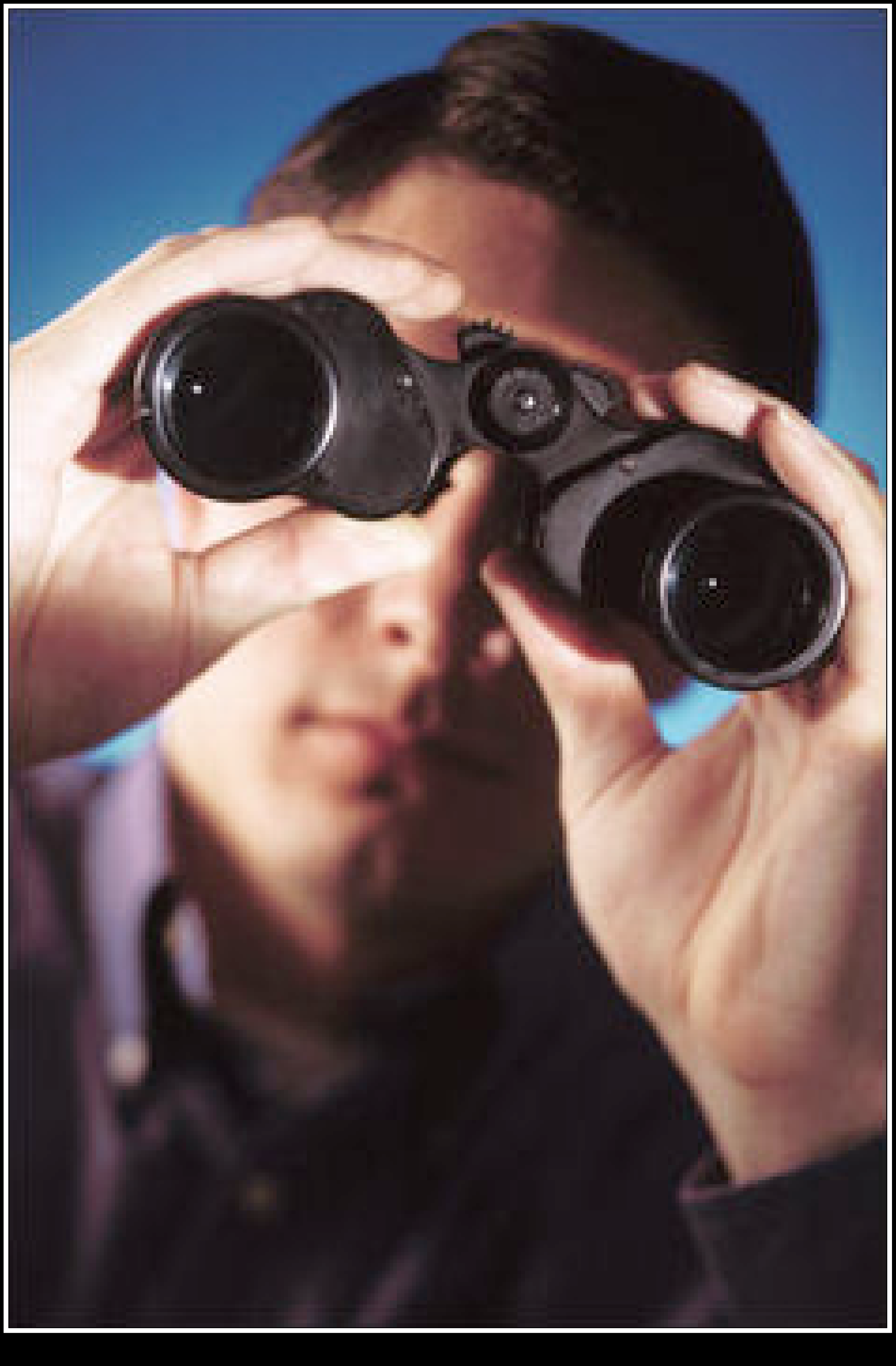
or

Empirical Software Engineering

Empirical means...?



Empirical means...?



- “Relying on or derived from observation or experimentation.”
- “Verifiable or provable by means of observation or experiment.”

Empirical Software Engineering

“...a branch of software engineering where the focus is to experiment on software systems including its products and processes.”

Empirical Software Engineering

"...a branch of software engineering where the focus is to experiment on software systems including its products and processes."

...or Experimental Software Engineering?

Analytical Advocacy Research

trust me... it will work!

Aeronautical Engineering

Aeronautical Engineering

Empirical analysis and testing

Formal Methods

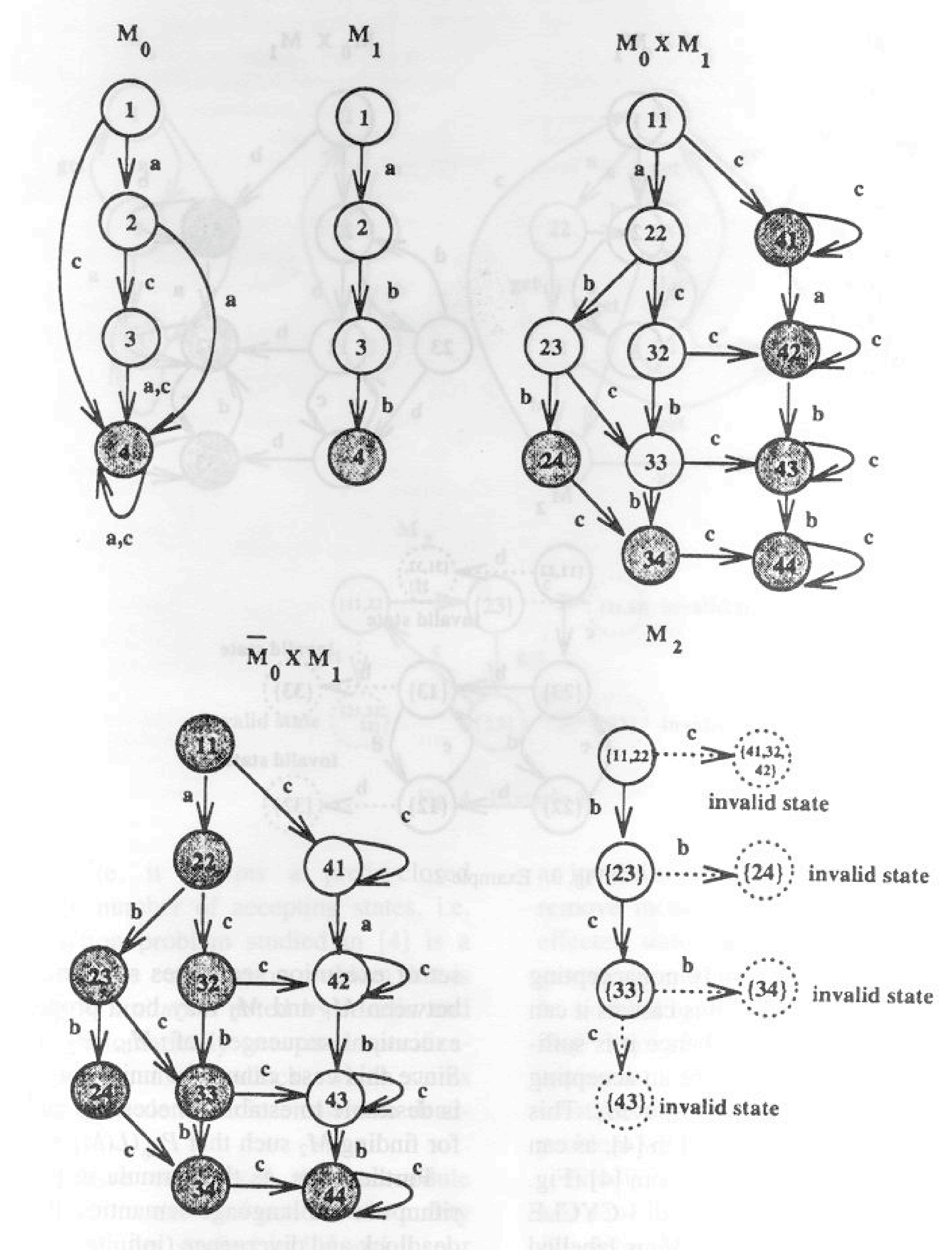


Fig. 2. Example 1.

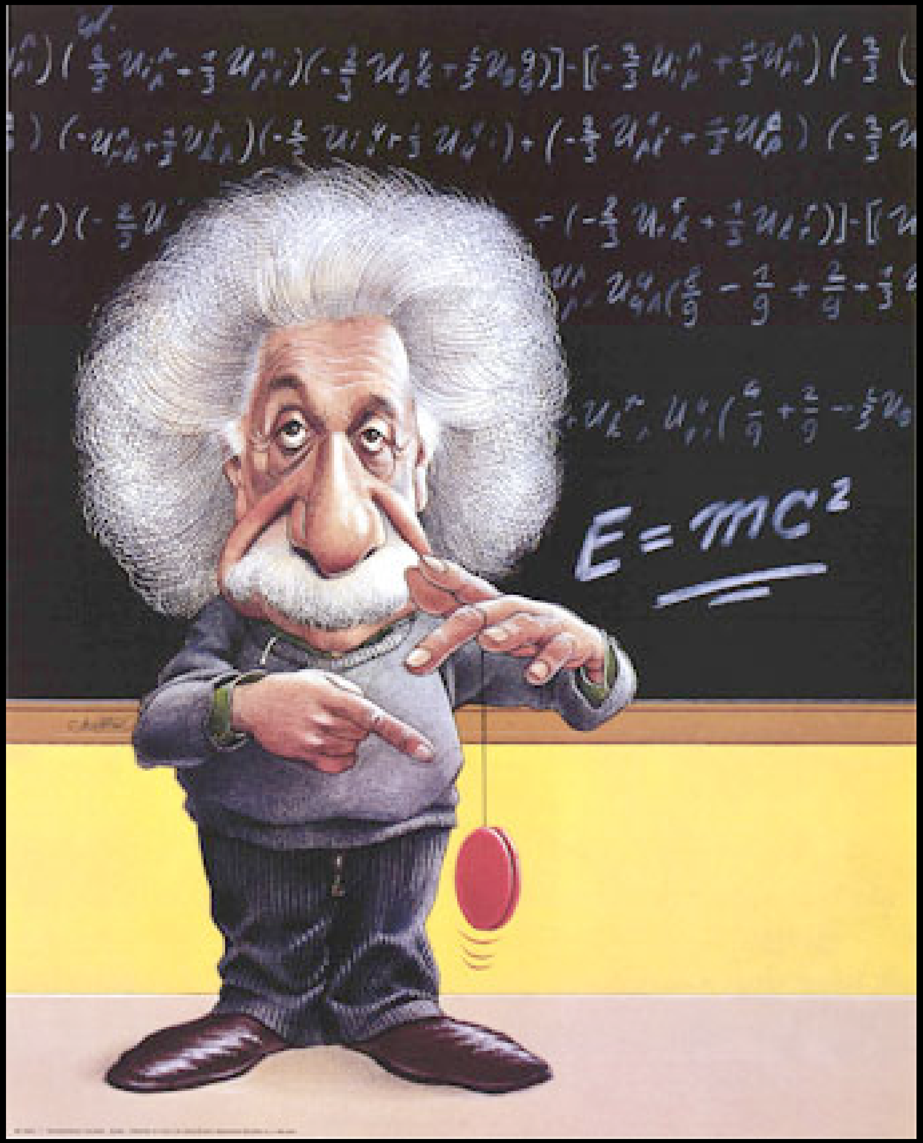
Evidence for use in fault tolerant systems?

Widespread appeal without rigourous experimentation.

Adoption requires revolutionary change in design.

IBM claimed a saving of $5.5 million, 9% and 60% fall in defects.

What can we borrow from other mature disciplines?



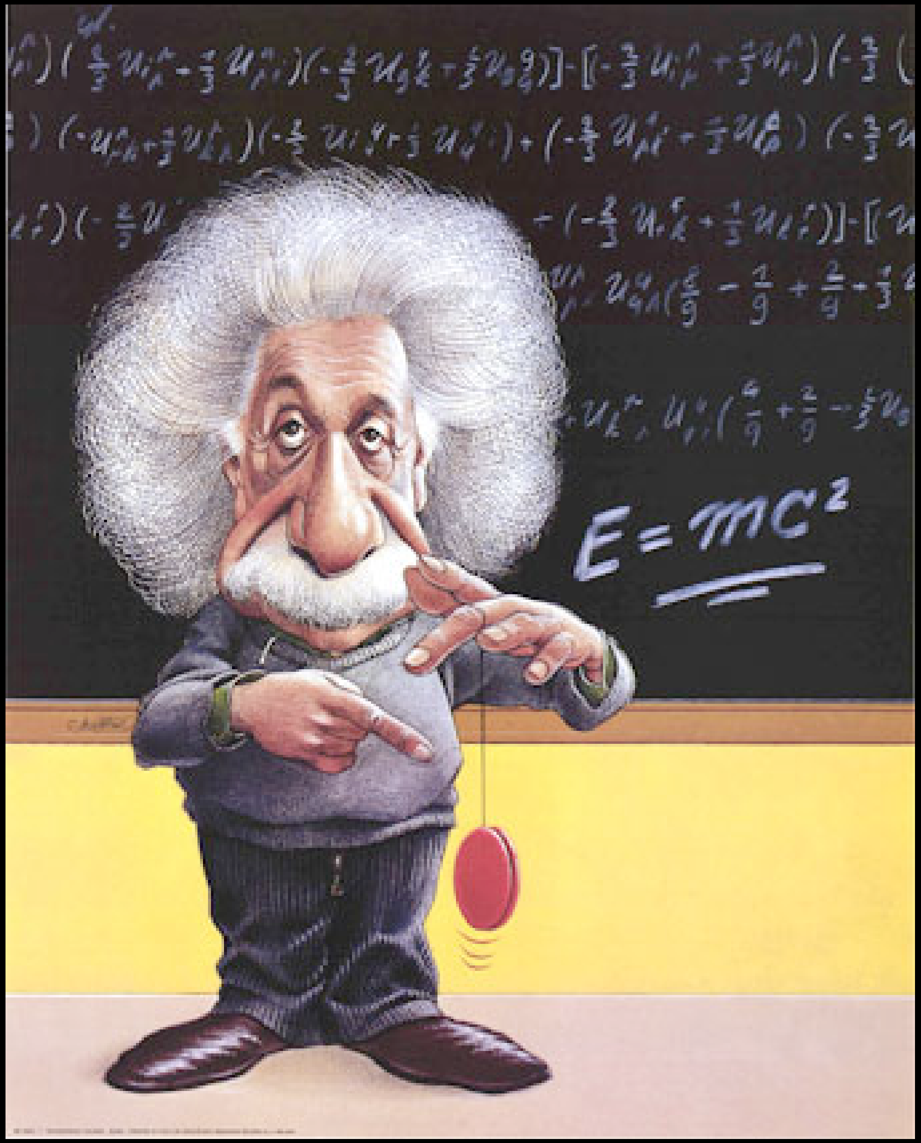
Isaac Newton

... law of mechanics described how particles respond to forces.

... law of gravity described how mass of objects is involved in their attraction for one another.

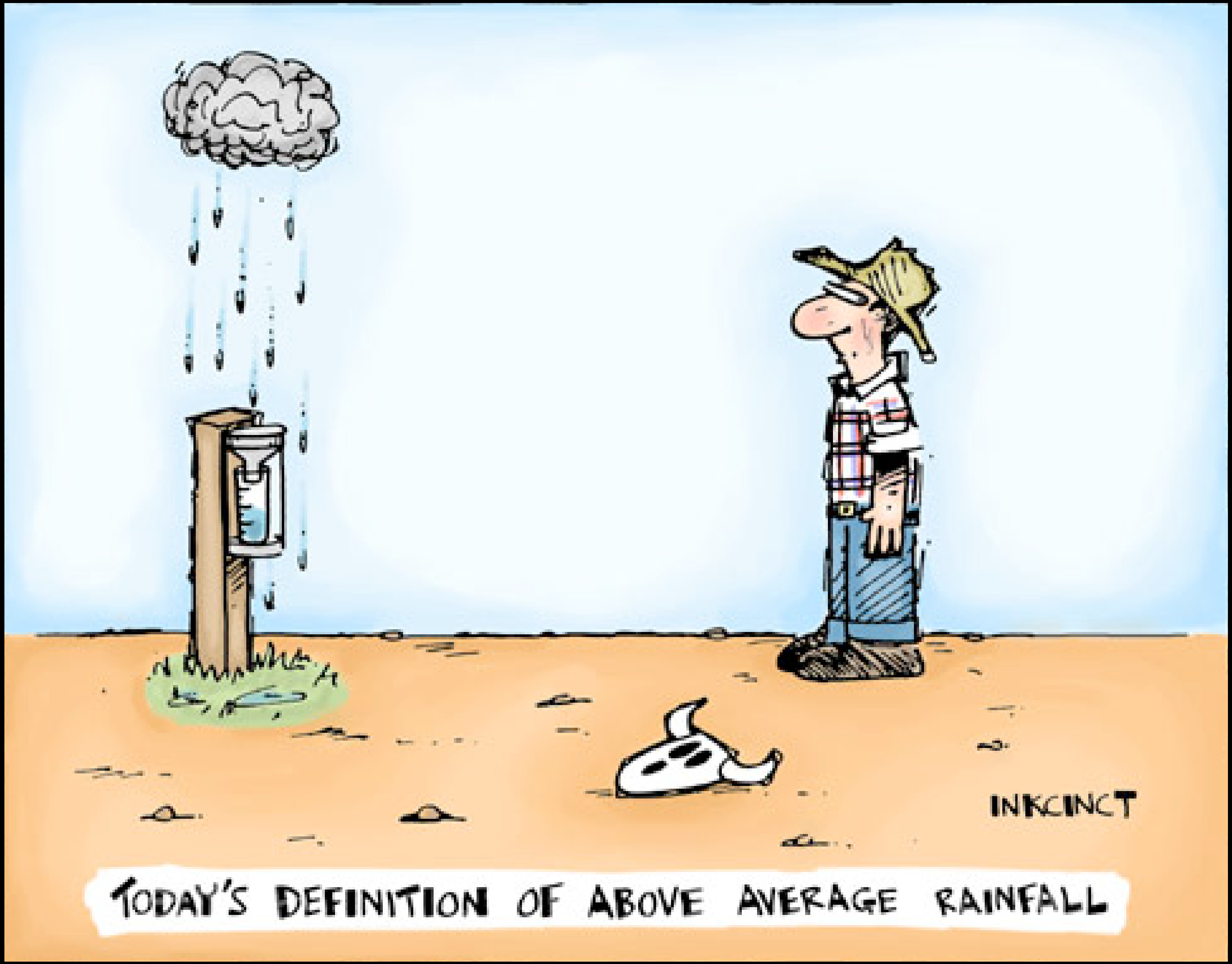
"most physicists believed that the world works in a rational way, and if they tried hard enough, they could find the rules by which this behaviour happens."

Albert Einstein

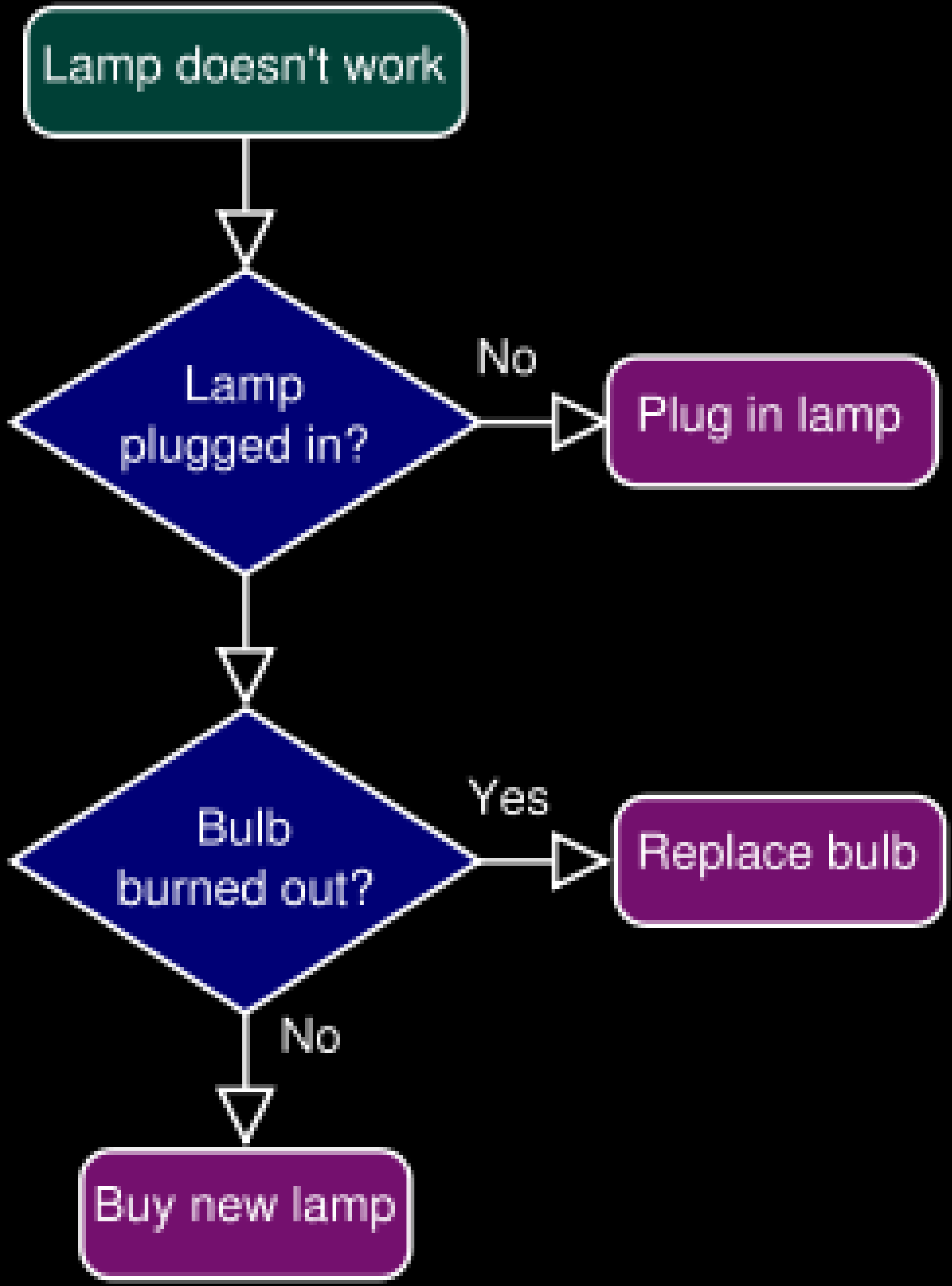


Theory of Relativity

Goal-Question-Metric



Poor Experimental Design



Ben Schneiderman and colleagues showed that flowcharts to not help progammer's comprehension.

Later, David Scanlan conducted experiments to suggest otherwise. He also pointed many design flaws in the above experiment.

Why Experiment?

Experiments don’t prove a thing.

Edsger Dijstra

University of Texas, Austin

Experiments can only show the presence of bugs in a theory, not their absence.

Fallacy 1

Traditional Scientific Experimentation is Inapplicable

Fallacy 2

The current level of experimentation is good enough.

Fallacy 3

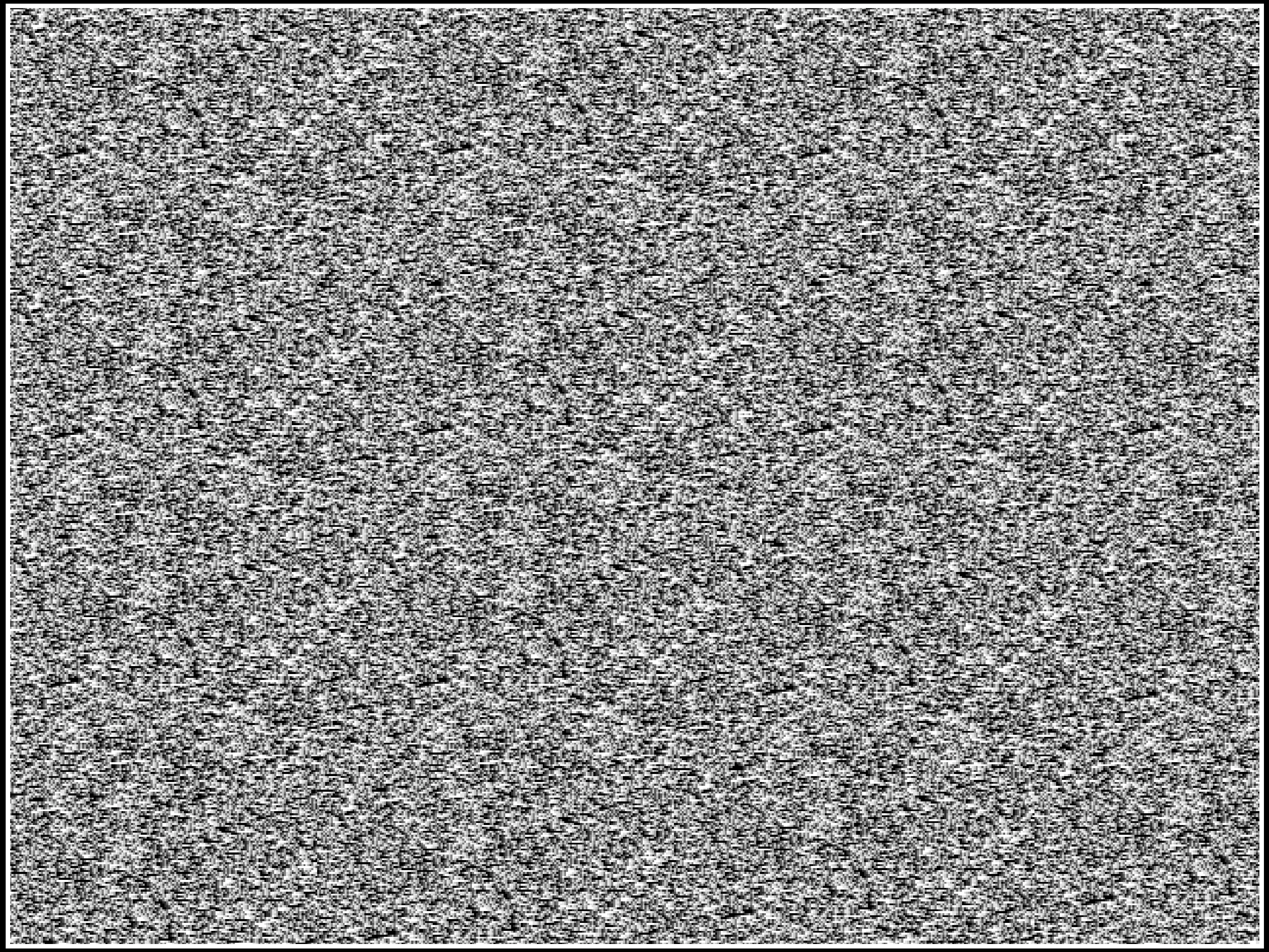
Experiments cost too much

Fallacy 4

Demonstrations will suffice.

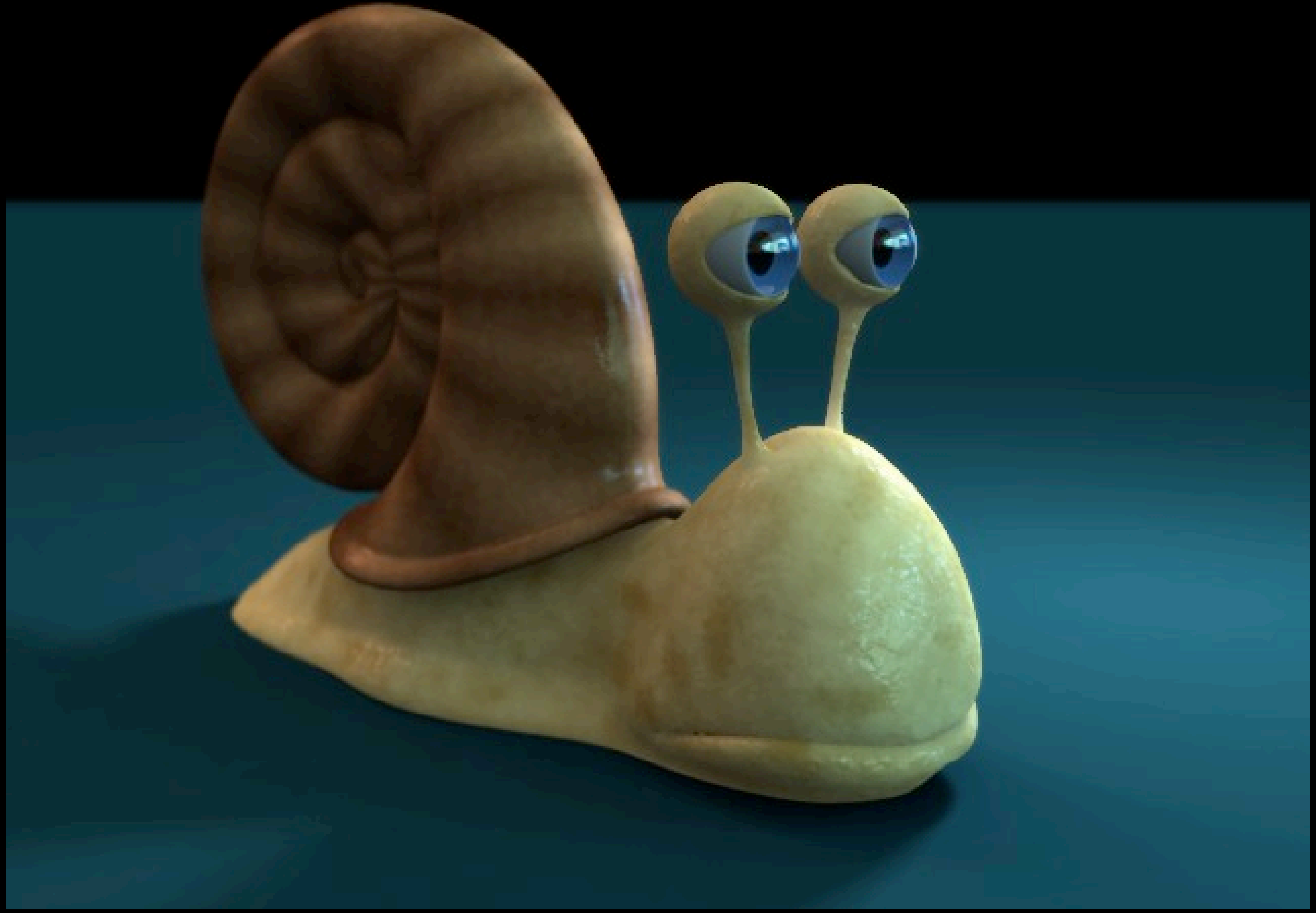
Fallacy 5

There is too much noise.



Fallacy 6

Experiments will slow progress.



Fallacy 7

Technology changes too fast.

Fallacy 8

You'll never get it published.

Conclusions

Conclusions

- ...but, currently little empirical evidence to confirm improvements.

Conclusions

- ...but, currently little empirical evidence to confirm improvements.
- need to carefully assess costs and benefits.

Conclusions

- ...but, currently little empirical evidence to confirm improvements.
- need to carefully assess costs and benefits.
- Perhaps, a widespread demand for change might change things?

Sources of Material

- W. F. Tichy, "Should Computer Scientists Experiment More?", IEEE Computer, 31, 1998, pp. 32-40.
- D. E. Perry, A. A. Porter, and L. G. Votta, "Empirical Studies of Software Engineering: A Roadmap", Limerick. Ireland, 2000.
- Fenton, N., S. L. Pfleeger, et al. (1994). "Science and Substance: A Challenge to Software Engineers." IEEE Software 11(4): 86-95.