EMPIRICAL STUDIES TO BUILD A SCIENCE OF COMPUTER SCIENCE

by: **Victor R. Basili** and **Marvin V. Zelkowitz**

Comm. ACM 50(11), **pp**. 33-37, **Nov** 2007.

Presenter: Arie (Lev) Shterengartz
On: 2/6/2011 (my 24-th birthday)

THE WRITERS:

Victor R. Basili

- Emeritus Professor at the Department of Computer Science and the Institute for Advanced Computer Studies at University of Maryland.
- IEEE Fellow, elected 1990.
- NASA Group Achievement Award, 1996 . (and Many other awards).
- Worked in industry and government but mainly in the academy.
- Well known for his works on measuring, evaluating, and improving the software development process and empirical studies.

Marvin V. Zelkowitz

- Emeritus Professor at the Department of Computer Science and the Institute for Advanced Computer Studies at University of Maryland.
- IEEE Fellow, elected 1997.
- Has many awards for researches in the field of empirical software engineering and measurement.
- Many publications in this field.
- Mostly known for his academic work, rather than in the industry.

What is empiricism?

What are empirical studies?

- **Empiricism asserts that knowledge comes (only or primarily) via sensory experience.**

- **Empirical research is a way of gaining knowledge by means of direct observation or experience.**

+
–
Use a flashlight bulb and not a regular light bulb when doing the experiment.

Empiricism in traditional sciences

Not the case in CS…

Empirical studies is about:

- Building models of the field that is in our interests.
- Encapsulation of our knowledge.
- Checking that our knowledge is correct.
- Evolving the knowledge over time.

Experimentation in CS

The situation today:

Experiments are around:

- Algorithms evaluation
- Checking performance
- Workflow of a system

The problem: it ignores the fact programming is made by humans…

Experimentation in CS should involve human activity

Should evaluate data to understand and improve the work of the development staff

Who will do the researches? Where will they be conducted?

Research – Development synergy

- **Developers need researchers to:**
  - Understand how to build systems better by understanding their own environments.
  - Predict cost & quality.

- **Researchers need developers for:**
  - Evidence of what works and what doesn't.
  - When it works? What will work better?
  - Be live models to check their hypothesis.

A good research must have a good research question

Examples of questions that every software organization must be able to reply:

- What is the right combination of technical and managerial solutions for my problem and environment?
- What is the right set of processes for my business?
- How do we learn from our successes and failures?
- How do we demonstrate sustained, measurable improvement?

Results de-facto VS. Intuition

Two options:

I. The results are against our intuition. In this case we need to check why is it, and change our hypotheses accordingly.

II. The evidence support the intuition. We may think the experiment was unnecessary. It is ***not*** true!!! Experiments add lots of information even if the conclusion is what we expected.

Example: NASA…

The study made in NASA Goddard Software Engineering Laboratory (SEL) Between 1976-2002.

Goal: improve software development for ground-support system.

- Used some methods and techniques for using knowledge to better understand how to build systems. This was called the Quality Improvement Paradigm (QIM).
- The process of building and testing models was encapsulated in a model Called Experience Factory (EF).
- Data was collected and interpreted with the Goal Question Metric approach (GQM).

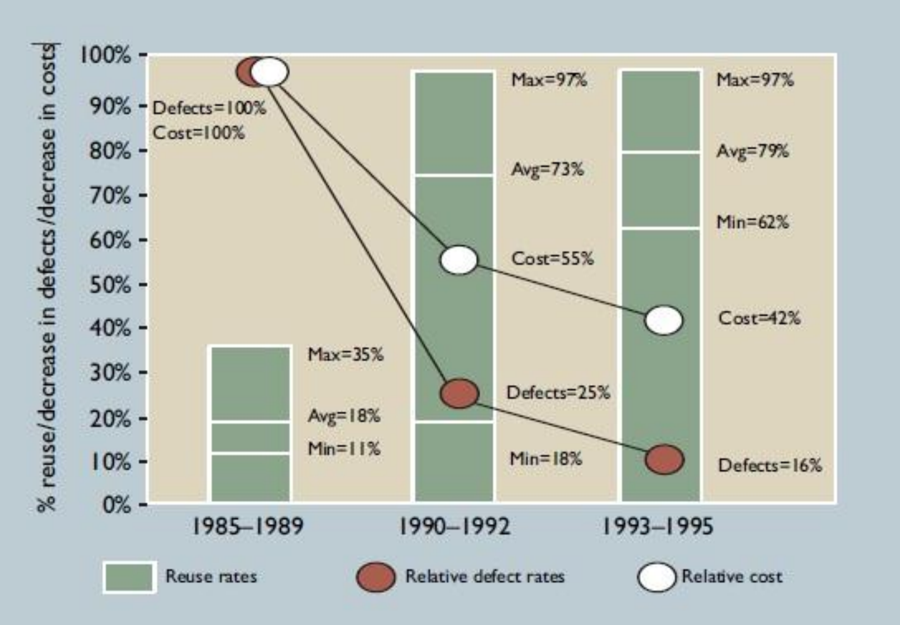
Adjustments in models could be made in real time and results deployed in future projects.

The studies were divided into two major classes:

Controlled experiments: applied to new techniques to mainly reduce the risks of applying it on live projects.

Case studies: applied for live projects to check the scalability of the technique and its advantages and disadvantages.

The results:



- ✓ Dramatic increase in code reuse
- ✓ Decrease in defects
- ✓ Decrease in costs

Conclusions:

- ➢ **empiricism is needed to build knowledge.**
- ➢ **Experimentation focused on people is necessary.**

The process is maybe slow – but it’s worth it!

Another example: High-End computing (HEC)

The subject:

Defense Advanced Research Projects Agency (DARPA)

The project:

many processors to achieve faster and more powerful computers
for various computation needs.
High Productivity Computing System (HPCS).

The formula:

"Time to solution = Development time + Execution time"

The innovation:

Instead of putting the emphasis on *execution* time –
Invest more time in the *development*!

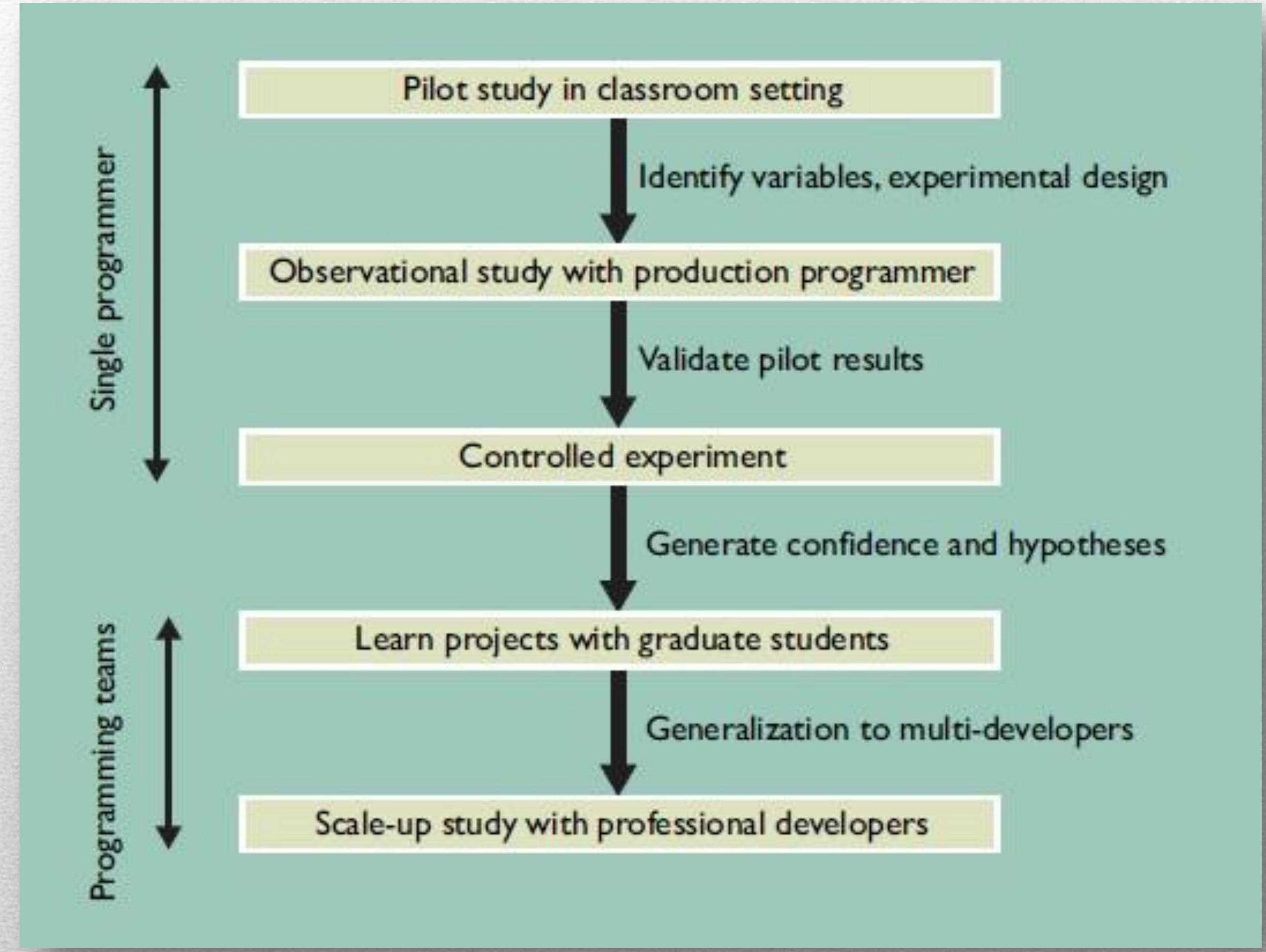
Same same but different

The research model in the HPCS is different than the one in SEL:

- Different personnel (pro. Programmers VS. comp. scientists & physicists).
- Different interests: answer "science" questions using a computer VS. how well the computer program actually work.

So the model is a bit more complex and it looks like this:

The research workflow:



Single programmer

Programming teams

Pilot study in classroom setting

Identify variables, experimental design

Observational study with production programmer

Validate pilot results

Controlled experiment

Generate confidence and hypotheses

Learn projects with graduate students

Generalization to multi-developers

Scale-up study with professional developers

Formalizing “folklore”

Throughout the research steps there were collected stories, notions and sayings from developers and users in government, industry and academy.

This folklore is formalized with 4 sequential activities:

I. Identify terminology and relationships looking for consensus or disagreement.

II. Identify “variables” that may affect validity of the first. (surveys etc.)

III. Develop hypotheses that can be specified and measured.

IV. Verify these hypotheses. (experimentation)

So, why experimenting?

- There are many differences between users and programmers level.
- New computers, processors and technologies are evolving frequently.
- Different characteristics, goals and needs of different organizations.

Computer scientists must understand the current state and identify the relationships between variables in order to be able to have progress.

Experimenting is the only way to do this right!

Summary & Conclusions

- Interaction between theorists and experimentalists (academy & industry).
- The learning process is continuous and evolutionary.
- Must involve users and developers.
- CS Researchers must experiment, analyze, synthesize and package the knowledge for future development.
- Experimentation is crucial for any engineering or science discipline.

Summary & Conclusions

- Interaction between theorists and experimentalists (academy & industry).
- The learning process is continuous and evolutionary.
- Must involve users and developers.
- CS Researchers must experiment, analyze, synthesize and package the knowledge for future development.
- Experimentation is crucial for any engineering or science discipline.

וכבר אמרו חז"ל: "אין חכם כבעל ניסיון"

Thank you!

Questions?