Introduction to Experimentation in Software Engineering

OHJ-1860 Software Systems Seminar
Empirical Software Engineering Research

Petri Selonen

Senior Researcher
Institute of Software Systems
Tampere University of Technology, Finland

27th February, 2007
petri.selonen@tut.fi
Outline

1. "Pre-scientific status" of software engineering
2. Notes on experimentation and empirical studies
3. Software engineering and experimentation
1. ”Pre-scientific status” of software engineering
Software engineering according to the IEEE

Software engineering is

(1) The application of a systematic, disciplined, quantifiable approach to the development, operations, and maintenance of software; that is, the application of engineering to software,

and

(2) the study of approaches as in (1).

[IEEE Standard 610.12-1990, pp. 67]
In the scientific method, experiment is a set of actions and observations, performed in the context of solving a particular problem or question, to support or falsify a hypothesis or research concerning phenomena. [Wikipedia]

- cornerstone in the empirical approach to acquiring deeper knowledge about the physical world
"Pre-scientific status” of software engineering

- Experimentation refers to matching with facts the assumptions, speculations and beliefs abound in software construction
- Which of the claims made by software development community are valid?
  - and under which circumstances?
"Pre-scientific status" of software engineering, cont’d

- **Natural selection of ideas through test of time**
  - certainty of an idea judged by amount of people using it
  - if few people use the idea, it is assumed to be false

- **Is this *modus operandi* more reminiscent of fashion than engineering?**
  - how long to wait until we can be sure an idea works?
  - and is it applicable to the settings we would like to use it?

- **Is software engineering really engineering?**
  - advocacy based research papers, anecdotes
Over 30% of software system papers have no experimental validation, and only 10% had some systematic experimentation [Zelkowich 1998]

Of 8% of papers with a "sizable" quantitative evaluation, none had formal evaluations (i.e., hypotheses, repeatable experiments) [Tichy 1993]

Of 400 research articles requiring experimental validation, 40% had none, compared to 15% in other disciplines [Tichy 1995]
Why do we not experiment in software engineering [Tichy 1998]?

1. Traditional scientific method is not applicable
   • To understand the information process, computer scientists must observe the phenomena and formulate and test explanations. This is the scientific method.

2. The current level of experimentation is good enough
   • Relative to other sciences, the data show that computer scientists validate a smaller percentage of their claims.

3. Experiments cost too much
   • Meaningful experiments can fit into small budgets; expensive experiments can be worth more than their cost.

4. Demonstrations will suffice
   • Demos can provide incentives to study a question further, but they often merely illustrate potential
Why do we not experiment in software engineering [Tichy 1998]?

7. There is too much noise in the way
   - Techniques can be used to simplify variables and answer questions

8. Experimentation slows progress
   - Increasing the ratio of papers with a meaningful validation has a good chance of actually accelerating the progress

9. Technology changes too fast
   - If a question becomes irrelevant quickly, it is too narrowly defined and not worth spending a lot of effort on

10. You will never get it published
    - Smaller steps are still worth publishing because they improve our understanding and raise new questions
2. Notes on experimentation and empirical studies
Knowledge

- Knowledge can attain the status of science
  - when it is verifiable and thus, valid
  - experimental results should be *reproducible* by an external agent

- Each new item of knowledge is confirmed by means of
  - fully defined experiments,
  - that can be replicated by other scientist,
  - who can verify the results
• Speculations are confronted with reality
  • experimentation is used to compare theoretical speculations and reality
  • experiments test whether the ideas are confirmed by events

• Experiments cannot prove a theory, they can only fail to falsify it (Popper 1935, 1959)
  • falsification instead of verifiability
  • empirical/scientific system must be capable of being tested by experience

• ”Scientific knowledge is not a system of true statements but claims that are provisionally true as long as they are not contradicted”
Kinds of empirical studies

- **Quantitative research** aims at numerical (quantifiable) relationship between several variables or alternatives under examination

- **Qualitative research** aims to examine objects in their natural settings
  - making sense and interpreting a phenomenon in terms of explanations people bring to them
  - explicate ways people in particular settings come to understand, account for, take action, and otherwise manage their day-to-day situations
  - gaining a holistic overview on the context under study
  - deals mostly with words
Kinds of empirical studies, cont’d

- **Example:**
  - a quantitative study can be used to determine that method A is more productive than method B, while
  - a qualitative study can be used to determine *why* method A is more productive than method B.

- **Quantitative and qualitative methods should be considered complementary**
  - often, qualitative studies are used to establish a hypothesis while quantitative studies are used to test them
First level of experimentation

- First level of experimentation is done *in vitro* in laboratories with strictly controlled conditions
- "Laboratory conditions" in software engineering can be a project free of marketing pressure
  - with control over the development process, background of the developers, etc.
- Researchers should publish original experiments and results so given theory can be considered to work in the laboratory settings at least
  - the results should be replicable
Second level of experimentation

- Second level of experiments should be carried out in real projects whose developers are prepared to run risks
  - for learning about latest technological innovations
  - early adopters and visionaries
  - tolerate shortcomings and failures
- Reporting under which conditions the approach worked and/or did not work
- The results of such experimental projects should be reported
  - limits and boundaries of the proposal should be defined
Third level of experimentation

• Ideally, only after laboratory experimentation and pilot projects should the approach be tested on "genuine real-world" projects
  • pragmatic practitioners

• Even then, the community should be responsible for collecting and publishing data on its own performance

• These three levels of experiments are sometimes called *laboratory experiments*, *quasi-experiments* and *surveys*, respectively
• Research is a process of *directed learning*

• Three models of reasoning for arriving at a hypothesis
  • induction: the premises of an argument are believed to support the conclusion but do not ensure it
  • abduction: starts from a set of accepted facts and infers to their most likely, or best, explanations
  • deduction: the conclusion is necessitated by, or reached from, previously known facts;

• **By deduction, a preliminary hypothesis should lead to particular consequences that can be compared against data**
  • when theory and data do not coincide, the hypothesis can be modified through induction and abduction
• Convergence towards results will be quicker and more certain if they are supported by proper experimental design and data analysis
  • efficient experimental design methods are as unambiguous and unaffected as possible by experimental errors
  • analysis of the collected data, as a result of experiments, specifies what can be reasonably deducted from the hypotheses and produces new ideas for consideration

• Good experimental design is even more important than data analysis
Experimentation/learning cycle, cont’d

- **Interacting with reality**
  - by *observation*: researchers perceive facts from outside
  - by *experimentation*: researchers enter into a dialogue with objects under study, interact with the outside world and have control over experiments

- **Speculation of reality**
  - hypothesizing about perception of the outside world
  - either mere description or induction
    - formulation of a "general law" that establishes relationships
    - in software engineering, relationships between development variables to predict implications to process itself and its output
Empirical knowledge vs. theoretical knowledge

- Often mechanism covering a process is not well enough known or it is too complex for an exact model to be constructed

- Experimental research aims to elucidate certain points of relationships among variables
  - survey inquiries: distinguish which of many variables affect other variables
  - empirical inquiries: discover empirical model that describes how certain variables affect others
  - mechanistic inquiries: produce a theoretical model that explains why variables affect response in observed manner
A discipline is formed as the body of validated knowledge grows

We look to discover relationships among the variables involved in a phenomenon

- questions like "how will the use of programming language X affect system reliability?" and "if project analysis stage has taken 50% longer than expected, what effect will this have on the remainder of the project?"

Several levels of relationships
- descriptive relationship
- correlation
- (probabilistic) causality
3. Software engineering and experimentation
Why do we not experiment in software engineering?

- Lack of training in the scientific method, experimentation in validating theories, and analysis of data
- Lack of experimental design and analysis books targeting software engineering
- Problems in publishing empirical studies
  - in other scientific and engineering disciplines, repetitions of experiments performed by others (to check their validity)
- Complexity and number of variables affecting software development
Why do we not experiment in software engineering, cont’d?

- **Huge money moved by software market**
  - constant development of new, increasingly complex systems
  - the market is used as a medium for performing the experiments

- **Problems in obtaining global, generalizable results**
  - however, it should still be possible to determine the circumstances under which one option is better than another
The human factor

- Software engineering results are not independent of practitioners
- Substantial obstacle in generalizing results yielded by empirical testing
- This is probably one of the major issues about software engineering research
  - software engineering research is as much about social sciences as it is about engineering (or "hard") sciences
  - traditional culture (e.g. at TampereUTech) does not take this into account all that well!
Software engineering can be considered as a social process

- its artefacts (methods/tools/paradigms) are affected by the experience, knowledge and capability of the user

The importance of the human element is an important difference between software engineering and other engineering disciplines

It takes place in a social context

- influenced by relationships among people (project teams, managers, users) and the social context (corporate culture, organizational procedures)
Repeatability in software engineering

- It is very difficult to repeat identical experiments for software development projects
  - with a fixed problem, team of developers, process and product, the development conditions are not the same for successive experimentation

- Phenomenon under investigation is very complex
  - identical parametric values are out of the question
  - replicability in software engineering has to be based on similarity rather than equality
  - basic characteristics of project must be defined to speak of replication based on similarity
    - human factor makes software engineering resemble cognitive psychology
Social aspect of experiments

- **Working with human beings makes experiments more complex than in natural sciences**
  - this is often used by software engineers as an excuse not to experiment

- **The social aspect must be taken into account when running experiments and generalizing results**
  - techniques like block design, randomization, different levels of experimentation, iterative procedures, etc.
The goal should be to extract useful principles of software construction

- through empirical investigation of successful projects
- validate design principles developed in research literature
- advance understanding of software engineering process by experimenting with new approaches

Yet, the software community does not take benefits of empirical investigation seriously

It seems that very few ideas in software engineering are matched with empirical data
Conclusions and discussion

- Juristo and Moreino book claims to tackle the above limitations in software engineering context

- Is software engineering so much different from other engineering disciplines?

- Can we generalize the results?

- Should we start with qualitative studies?