Panel: Empirical Validation—What, Why, When, and How

Robert J. Walker (Chair)
University of Calgary
Calgary, Canada
rwalker@cpsc.ucalgary.ca

Lionel C. Briand
Carleton University
Ottawa, Canada
briand@sce.carleton.ca

David Notkin
University of Washington
Seattle, USA
notkin@cs.washington.edu

Carolyn B. Seaman
University of Maryland, Baltimore County
Baltimore, USA
cseaman@umbc.edu

Walter F. Tichy
Universität Karlsruhe
Karlsruhe, Germany
tichy@ira.uka.de

All research requires validation, and where analytic solutions are not possible or not practical, empirical validation is necessary. Software engineering research covers a vast array of problems over which the level of maturity in our knowledge varies markedly.

Opinions as to the methodology to apply to SE research appear to be in disagreement, as a large variety of possibilities exist [7]. Tichy promotes quantitative, controlled, statistically-analyzable experimentation [6]. Kitchenham *et al.* recognize the value of "observational studies" in addition to formal experimentation [3], but emphasize industrial-context, quantitative evaluation, and statistics. Seaman promotes the value of qualitative evaluation [5]. Murphy *et al.* suggest that a different treatment is needed for emerging technologies than for more mature ones [4]. Briand *et al.* state that "each discipline needs to develop its own body of experience and strategies to answer its most pressing research questions" [1, p. 398].

Without some consensus, an SE researcher is faced with a difficult task of convincing their peers that their selected methodology is appropriate, let alone the details of their validation. This panel session strives to address these issues in order to determine where a consensus does and does not exist. Brief synopses of each panelist's thoughts follow.

Lionel C. Briand

Quantitative and qualitative evaluation are not really alternatives but are rather complementary. For example, when assessing the impact of a new technology in a controlled experiment, it may not be a good idea to assume that your research hypothesis is the unique reason for the results you observe. Questionnaires are commonly used to obtain qualitative but structured feedback from the experimental subjects for additional evidence of causation.

We need to study both current practices and new technologies. The former is needed to better understand what issues people are actually facing when developing and maintaining large software systems. The latter is needed to improve software quality and productivity.

Level of control is usually associated with internal and external validity. Fully controlled experiments usually take place in an artificial setting, for obvious practical reasons. In this case a very good internal validity is usually obtained (i.e., the treatment is likely to cause the effect) but external validity suffers from the fact that the setting may not reflect actual project conditions. When performing field studies, collecting data on actual projects and interviewing project participants, you then find yourself in the opposite situation.

In the end, when an important question arises regarding, for example, the evaluation of a new technology, both controlled experiments and field studies are necessary to build a credible body of evidence.

David Notkin

"Software engineering research is poorly evaluated and assessed" is a statement made by many in the community in print and perhaps more so in private. Although I believe that we have to make serious and significant improvements, I reject this statement as inaccurate and dangerous.

The statement is inaccurate because it makes assumptions that are not true; I address two of these assumptions. First, there is a strong implication that to get industrial adoption of a result, one must have empirical evidence for its utility. Empirical evidence may be material, and in some cases essential, but the general issue of technology transfer is extremely complicated. The Brooks–Sutherland Report [2] discussed how government, industry, and academic institutions play key roles in the development and transfer of influential technologies, with the process being an

Proceedings of the 25th International Conference on Software Engineering (ICSE'03)
0270-5257/03 $17.00 © 2003 IEEE

extremely dynamic one. Second, it generally appeals to "real science" as an example of what we should become. There are a number of reasons why this is inappropriate, not the least being an idealized but imprecise understanding of what research in these other fields looks like. Furthermore, the combination of heavily different environments (in terms of the kinds of software produced, the educational and experience backgrounds of the software engineers, etc.) with the uniquely rapid rate of change in our field makes the situation different from other "real science" fields.

The statement is dangerous because it encourages researchers to focus on problems that can evaluated in particular ways rather than on problems that need to be solved. We need to have strong demands about evaluation and assessment, but realize that one size does not fit all. Not all ideas nor claims are the same, and they should not generally be placed into pigeonholes. What is fundamentally important is for the claims to be appropriately justified: heavy claims demand heavy evaluation. Additionally, we need to have a sense of the lifetime of different ideas. Ideas in their early stages need to be given some room to grow, without overly restrictive expectations about their evaluation. More mature ideas need more mature evaluation.

Let's not strangle ourselves, and our chances of having an influence, by hiding behind a few specific approaches to evaluation of our work. Rather let's be demanding but flexible in this dimension.

Carolyn B. Seaman

One of the principles of empirical investigation in any field is triangulation. This is the premise that any stated finding should be supported by more than one piece of "evidence". Another way of saying this is that any data generated by an empirical study cannot be considered valid unless the same information has been generated from two different data sources, by using two different empirical methods, or by using two different measurement instruments.

Taking triangulation seriously in empirical studies of software engineering means that software engineering empirical researchers must broaden their repertoire of empirical methods. Qualitative research methods constitute a large body of well-defined, mature empirical methods that have not been fully employed in the study of software engineering. Qualitative research is empirical research in which the data is expressed textually or in images, rather than in quantifiable terms. Qualitative methods were originally developed principally for the study of human behavior, under the premise that such a subject is too complex to be captured adequately quantitatively. Thus, the qualitative approach allows the researcher to capture the full complexity of a phenomenon without having to abstract away important details that do not fit into a quantitative model.

Much of the complexity of many research questions in software engineering comes from the role of humans in the development and maintenance of software, and the fact that software itself is a product of human intellect. Thus, the use of qualitative methods in software engineering research provides a way to deal with the complexities of these questions, as well as providing excellent opportunities for triangulation, when used in conjunction with quantitative methods.

Walter F. Tichy

The quantity of empirical research in SE has increased dramatically. A sizeable proportion of papers in most major SE publications and conferences now presents empirical results. Now that quantity is up, quality must be improved as well. Some of the empirical work is unsatisfactory, for a variety of methodological reasons. I think reviewers have to be both more careful and more understanding: more careful in checking the validity of the results, and more understanding regarding what can be achieved in a single paper. There is no "silver bullet experiment" that will provide the final answer for a research question, nor can an empirical paper be rejected solely on the grounds of the results appearing "obvious" to the reviewer.

I think we should give up on correlating volume metrics (counting attributes of various documents in the life cycle) with cost estimates. Experiments have shown that the predictive power of volume metrics is poor and "tuning" them does not carry over from one project to another. Instead, we need to figure out better models of software development processes to achieve better estimates.

Besides metrics, there are many other pressing questions regarding the effectiveness of software tools and methods that can only be answered empirically.

References

[1] L. Briand et al. Empirical studies of object-oriented artifacts, methods, and processes. *Empiric. Softw. Eng.*, 4(4):387–404, 1999.

[2] F. P. Brooks, Jr., I. E. Sutherland, et al. *Evolving the High Performance Computing and Communications Initiative to Support the Nation's Information Infrastructure*. The National Academies Press, 1995.

[3] B. Kitchenham et al. Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Softw. Eng.*, 28(8):721–734, 2002.

[4] G. Murphy, R. Walker, and E. Baniassad. Evaluating emerging software development techniques. *IEEE Trans. Softw. Eng.*, 25(4):438–455, 1999.

[5] C. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Trans. Softw. Eng.*, 25(4):557–572, 1999.

[6] W. Tichy. Should computer scientists experiment more? *Computer*, 31(5):32–40, 1998.

[7] M. Zelkowitz and D. Wallace. Experimental models for validating computer technology. *Computer*, 31(5):23–31, 1998.

Proceedings of the 25th International Conference on Software Engineering (ICSE'03)
0270-5257/03 $17.00 © 2003 IEEE