

## Towards Standard for Experiments in Program Comprehension.

Václav Rajlich, George S. Cowan  
Department of Computer Science  
Wayne State University  
Detroit, MI 48202  
rajlich@cs.wayne.edu, cowan@acm.org

### Abstract

*Program comprehension can make a unique contribution to the field of software engineering because it is feasible to validate its claims with inexpensive experiments. To fully realize this unique position, program comprehension researchers need to develop standards that will guide them in designing experiments and allow them to judge the strength of an experiment in supporting a claim. To begin the discussion leading to such standards, we propose that program comprehension experiments always measure and interpret the following dependent variables: accuracy, accurate response time, and inaccurate response time.*

### 1. Introduction.

Experimental methods in software engineering validate the tools and processes of software engineering as well as its theories. The tools and methods are used by programmers to work on programs, and the experiments are the only validation which takes into account all the variability found among the programmers and programs. This situation is in many respects similar to the situation in other fields dealing with human subjects, where the products are validated by experiments before being applied in general use. For example, before new drugs are introduced into general use, they first must be validated in a series of experiments which measure both their effects and side effects.

At this point, the software engineering field does not expect the same level of validation as is common in other fields. One reason is the prohibitive cost of most experiments. For example, experimental validation of a new software development process or a new software technology would require development of many programs, where each program development would be just one data point. The cost of such an experiment would be enormous, and therefore the complete experimental validation is unlikely. Contrast this with the situation in the area of program comprehension. Program comprehension is one of the areas of software engineering where the cost of the experiment is relatively low. One data point in these experiments is one answer to one question from one person, and its cost is much lower than that in other areas of software engineering. Hence experiments in program comprehension are feasible, even for large software systems. As well as directly answering important questions, they can be used to provide valuable insight into software engineering issues in those situations where other approaches cannot be applied because of costs. However, to make the reliable use of experimental results more widespread, the program comprehension community should develop a standard for such experiments, deciding what does and what does not constitute an acceptable experiment. The purpose of the standard would be to guide software engineers in designing experiments and to allow them to judge the strength of an experiment in supporting a claim. In this brief statement, we discuss one aspect of such a standard, the dependent variables of comprehension experiments.

## 2. The dependent variables.

In a software comprehension experiment, there are three dependent variables. How accurate are the answers that programmers give to questions about the program? What is the response time for the accurate answers? What is the response time for inaccurate answers? McLeod and Nelson [3] give experimental evidence that these three variables measure different underlying phenomena. They suggest that accuracy measures difficulty, accurate response time measures the number of steps in the process as well as sum of the difficulty of the steps, and inaccurate response time measures willingness to keep trying versus giving up. However the experiments of McLeod and Nelson deal with the psychology of memory, where the objects are simple items, and the underlying theory may not apply to the comprehension of complicated software systems. Hence the three different dependent variables need a new interpretation for the program comprehension.

The current theories of program comprehension [1,4] do not provide a direct interpretation of the three dependent variables of the program comprehension experiment, and hence the interpretation of the three dependent variables is a task which needs to be resolved. In [2], the issue of the three dependent variables was resolved by an experimental design which had a ceiling effect on accuracy, and the (mostly accurate) response time was the only dependent variable which was used to reach the conclusions. Is that an acceptable practice? What happens if there is no ceiling effect on the accuracy? Note that accuracy and response time measure different things. For example, if one method of documentation enabled programmers to work more quickly while another one enabled them to work more accurately, that would be an interesting result in itself. Therefore we believe that a full standard for comprehension experiments will require the measurement and interpretation of all three dependent variables.

The three dependent variables are only one facet of experimental design. Other issues which need to be resolved include independent variables, populations of programmers, and scaling up the results of the experiments to large systems. It is important for the program comprehension community to work on these issues, so that the full potential of the comprehension experiments can be realized, for the benefit of the entire software engineering field.

## References.

- [1] Brooks, Ruvim; "Towards a Theory of the Comprehension of Computer Programs", *International Journal of Man-Machine Studies*, Vol. 18, pp. 543-554, 1983
- [2] Curtis, Bill; Sheppard, Sylvia; Kreusi-Bailey, Elizabeth; Bailey, John; and Boehm-Davis, Deborah A.; "Experimental Evaluation of Software Documentation Formats", *Journal of Systems and Software*, vol. 9, pp. 167-207, 1989
- [3] MacLeod, Colin M.; and Nelson, Thomas O., "Response Latency and Response Accuracy as Measures of Memory", *Acta Psychologica*, vol. 57, pp. 215-235, November, 1984
- [4] Pennington, N., "Comprehension Strategies in Programming", *Empirical Studies of Programmers: Second Workshop*, Olson, Sheppard, and Soloway, eds., Ablex Publishing Corporation, pp. 100-112, 1987
- [5] Basili, Victor R., The Role of Experimentation in Software Engineering: Past, Current, and Future, Proc. 18. International Conference on Software Engineering, IEEE Computer Society Press, 1996, pp. 442-449.