Techniques for Empirical Validation

Marvin V. Zelkowitz

Abstract. In 1998 a survey was published on the extent to which software engineering papers validate the claims made in those papers. The survey looked at publications in 1985, 1990 and 1995. This current paper updates that survey with data from 2000 and 2005. The basic conclusion is that the situation is improving. One earlier complaint that access to data repositories was difficult is becoming less prevalent and the percentage of papers including validation is increasing.

1 Introduction

Any science advances by the process of developing new abstract models and then a series of experiments to test those models against reality. However, all too often in the software engineering domain, models (e.g., programs, theories) are described without any corresponding validation that those models have any basis in reality.

In order to determine the status of experimental validation in software engineering, in 1998 a paper by Zelkowitz and Wallace [4] surveyed the research literature in order to classify the experimental methods used by authors to validate any technical claims made in those papers. A total of 612 papers, published in 1985, 1990 and 1995, were studied. Of these, 62 were deemed not applicable, leaving 560 research papers. The 3 data sources used for this survey were:

- ICSE – Proceedings of the International Conference on Software Engineering
- TSE – IEEE Transactions on Software Engineering
- SW – IEEE Software Magazine

Each of the research papers was classified according to a 14-scale taxonomy:

1. Project monitoring. Collect the usual accounting data from a project and then study it.
2. Case study. Collect detailed project data to determine if the developed product is easier to produce than similar projects in the past.
3. Field study. Monitor several projects to collect data on impact of the technology (e.g., survey).
4. Literature search. Evaluate published studies that analyze the behavior of similar tools.
5. Legacy data. Evaluate data from a previously-completed project to see if technology was effective.
6. Lessons learned. Perform a qualitative analysis on a completed project to see if technology had an impact on the project.

V. Basili et al. (Eds.): Empirical Software Engineering Issues, LNCS 4336, pp. 4 – 9, 2007.

7. Static analysis. Use a control flow analysis tool on the completed project or tool.
8. Replicated experiment. Develop multiple instances of a project in order to measure differences.
9. Synthetic. Replicate a simpler version of the technology in a laboratory to see its effect.
10. Dynamic analysis. Execute a program using actual data to compare performance with other solutions to the problem.
11. Simulation. Generate data randomly according to a theoretical distribution to determine effectiveness of the technology.
12. Theoretical. Formal axiom-proof style paper describing a new theory.
13. Assertion. Informal feasibility study of the technology. (More of an existence proof rather than an evaluation of the claims of the technology).
14. No experimentation. The default classification if a paper fails to fall into any of the preceding classification.

The first eleven categories represented various empirical validation methods. Method 12 (Theoretical) indicated that the paper was a formal model of some property. (The original 1998 paper did not include a separate theoretical category, as methods 12 and 14 were combined as one category.) There was a thirteenth quasi-validation method, called an assertion. Assertion papers were those where the author knew that an experimental validation would be appropriate, but only a weak form of validation was applied. (For example, a paper describing a new programming language might only show that it was feasible to write programs in that language, not whether the programming language solved any underlying problem that needed to be solved.) All other papers were characterized as “No experimentation,” indicating that some form of validation was appropriate, but was lacking.

The basic conclusion was that approximately half of the papers had an inadequate level of validation. Similarly, Walter Tichy in 1994 [2] did his own literature search using a different protocol, yet came up with a similar conclusion. The general result was that the software engineering community was not doing a good job in developing a science of software development.

It is now ten years after these two surveys, so it seemed appropriate to redo the 1998 study in order to see if the situation had changed. One of the conclusions in the Zelkowitz and Wallace paper was that the situation seemed to be improving. Since two more 5-year milestones have since passed, it is worthwhile to revisit that initial survey to see how the research world has changed in the approximately 10 years since the original survey was conducted.

If we were to redo in total, a slightly different taxonomy would be chosen than the 14-point scale given above. However, one goal was to understand how the research world has changed since the 1990s, so the same classification model was used. One problem today is that we don’t have an agreed upon model for classifying software engineering research methods. Two other surveys compiled in the interim period [1] [3] use a different classification model for determining the experimental validation method used.

Table 1 presents the basic data from both the original and 2006 survey. In the 2006 survey, an additional 361 papers were evaluated, with 35 not applicable, leaving 326 additional research papers to classify.

2 Observations

The percentages (excluding the "not applicable" category) for each of the 13 validation methods are given in Figure 1. Case study remains the most popular method, increasing in each survey period from 8.3% of the papers in 1985 to 18.8% in 2005. The "classical" experimentation method of a controlled replicated study (represented as the sum of synthetic and replicated in Figure 1) grew slightly to 5.3% of the papers in 2005 from 2.6% in 1985. Dynamic analysis dominated the experimental methods in 2005 with 20% of the papers. A possible reason why this is so is given later.

Table 1. Classification data from 973 papers: 1985-2005

		Project monitoring	Case study	Field study	Literature search	Legacy	Lessons learned	Static analysis	Replicated	Synthetic	Dynamic analysis	Simulation	Assertion	Theoretical	No experimentation	Not applicable	Total
	ICSE	0	5	1	1	1	7	1	1	3	0	2	12	3	13	6	56
	TSE	0	12	1	3	2	4	1	0	1	0	10	54	18	38	3	147
	SW	0	2	0	1	1	5	0	0	1	0	0	13	1	10	6	40
1985 Total		0	19	2	5	4	16	2	1	5	0	12	79	22	61	15	243
	ICSE	0	7	0	1	2	1	0	0	0	0	0	12	1	7	4	35
	TSE	0	6	1	1	2	8	0	1	4	3	11	42	19	22	2	122
	SW	1	6	0	5	0	4	0	0	1	0	0	19	0	8	16	60
1990 Total		1	19	1	7	4	13	0	1	5	3	11	73	20	37	22	217
	ICSE	0	4	1	0	1	5	0	1	0	0	1	4	3	7	5	32
	TSE	0	10	2	2	1	8	2	3	2	4	6	22	7	7	1	77
	SW	0	6	1	3	1	7	0	0	0	0	1	14	0	3	7	43
1995 Total		0	20	4	5	3	20	2	4	2	4	8	40	10	17	13	152
	ICSE	0	10	0	0	1	4	0	2	2	4	1	11	3	20	10	68
	TSE	0	9	3	1	0	0	0	0	4	4	7	11	10	15	2	66
	SW	0	7	3	1	1	3	0	0	3	0	0	4	1	11	19	53
2000 Total		0	26	6	2	2	7	0	2	9	8	8	26	14	46	31	187
	ICSE	0	14	1	0	1	0	0	0	3	8	1	10	1	3	0	42
	TSE	0	9	4	1	5	0	2	1	2	13	5	13	1	8	2	66
	SW	0	9	4	1	5	0	2	1	2	13	5	13	1	8	2	66
2005 Total		0	32	9	2	11	0	4	2	7	34	11	36	3	19	4	174

More important than individual methods is the general "health" of the software engineering research field. This is summarized by Figure 2. Except for 2000, the percent of "No experimentation" papers dropped from 26.8% in 1985 to only 10.9% in 2005. Assertions dropped from 34.6% in 1985 to 21.2% in 2005. The percent of papers that used one of the 11 validation methods rose from 29% to 66% in 2005. (The percentage rose from 39% to 68% when theoretical papers were also included.) Clearly the situation is improving. This is consistent with an alternative study of the International Software Engineering conferences (ICSE) [3]. Using a sampling technique over all 29 ICSE proceedings, they found that 19 of 63 papers included no empirical study (30%). This present study indicates that 50 out of 208 ICSE papers

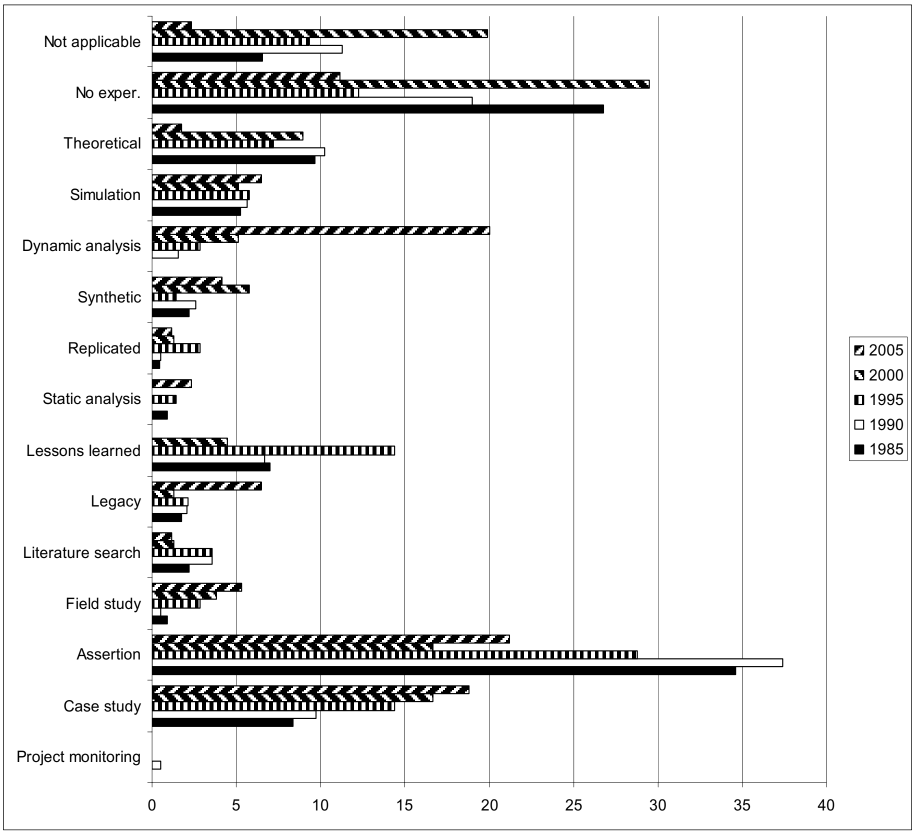

Fig. 1. Percentages of each validation method

(26%) had no experimentation. They also found a statistically significant increase in evaluation papers between the conferences prior to 1990 and those since then.

Unlike in the Zannier et al study [3], no attempt was made to evaluate the quality of the validation presented in those papers. (It was beyond our knowledge to understand and evaluate all 886 papers, but it was fairly easy to understand hoe the authors proposed to evaluate that technology.) If the paper stated an hypothesis about the technology described in the paper (even if stated indirectly) and then proceeded to describe a validation method for that hypothesis, we considered it as validated. Perhaps the hardest part of the study was trying to understand what the underlying hypothesis really was and how the authors would proceed to evaluate it. As stated earlier, we need a common terminology in which to describe validation methods. Many of the authors used terms like "experiment," "case study," "simulation," "controlled," etc. in very different ways.

Several anecdotal observations are buried in the data. A common complaint 20 years ago was the lack of published data sources that others could have access to. That seems to be changing. Many of the papers used the various open source

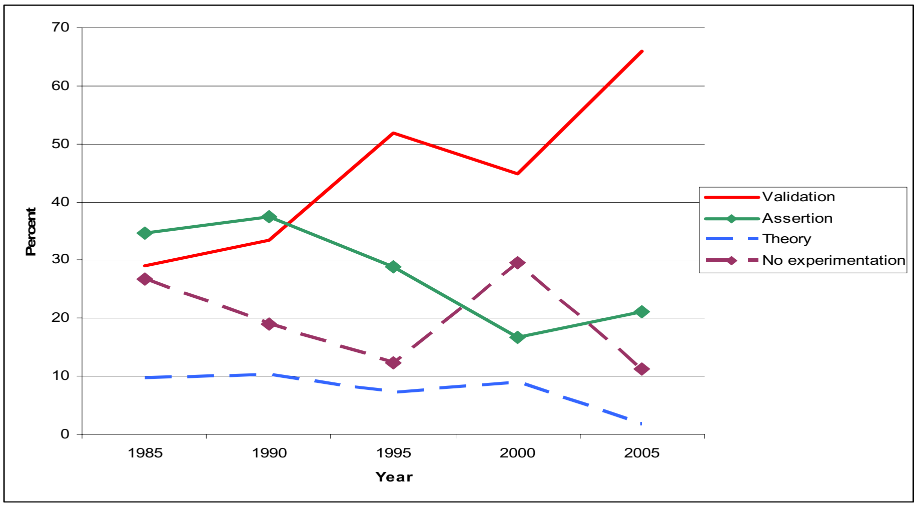

Fig. 2. Changes over time in validated papers

repositories, looking at the development history of products such as the Apache web server or Mozilla, as sources for data. This use of historical data using open source and other data repositories was one of the reasons for the rise in the dynamic analysis category in Figure 1. Similarly, data mining through theses sources led to the rise in the legacy data category.

3 Conclusions

There are several threats to the validity of this study.

1. The 2006 classification was performed about 9 years after the earlier study. While the same classification process was used to classify the papers according to the 14-point taxonomy, undoubtedly the intervening years may have changed our views of some of the validation methods. Consistency of this somewhat subjective classification method is a problem. For example, in [1], they report 0 and 3 controlled studies in ICSE for 1995 and 2000, respectively, while Table 1 shows 1 and 4, respectively, for those years in our classification). While this may have affected individual percentages in Figure 1, it should not have had much of an impact on the overall results as given by Figure 2.

2. As with the earlier 1998 study, each paper source for each year was managed by a different editor or conference chair. This has an effect on the overall acceptance rate of various papers submitted to that source. For example, the rise in “No experimentation” in 2000 was partially due to the largest number of ICSE papers (68) in the entire survey and the relatively large number of “No experimentation” papers (20) in those proceedings. Although such variances affect individual sources in a given year, the overall trends seem consistent.

3. There was a change in the scope of IEEE Software between 1995 and 2000. In the earlier survey, this magazine often published longer articles that had a research component. However, more recently the papers have been shorter with more regular columns appearing in each issue. Regular columns were not included in this survey and a value judgment was made on the remainder of the papers. If a paper discussed many solutions to a given problem, the paper was considered a tutorial or survey and listed as "Not applicable," but if the paper focused on a particular technique (often the author's), then it was considered a research paper.

The greatest limitation to this study, however, was mentioned earlier – the quality of the evaluation was not considered in classifying a paper. If the field is to mature as a scientific discipline, not only do we need empirical validation of new technology, we also need quality evaluations. However, that study still needs to be done.

In spite of these limitations, the results should prove of interest to the community. It provides a general overview of the forms of validation generally used by the computer science community to validate the various research results that are published and it does show that the field is maturing. Computer science seems to be developing an empirical culture so necessary to allow it to mature as a scientific discipline.

References

1. Sjøberg D. I. K., J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N-K Liborg and A. C. Rekdal, A survey of controlled experiments in software engineering, IEEE Trans. on Soft. Eng. 31, 9 (2005) 733-753.
2. Tichy W. F., P. Lukowicz, L. Prechelt, and E. A. Heinz, Experimental evaluation in computer science: A quantitative study, J. of Systems and Software 28, 1 (1995) 9-18.
3. Zannier C., G. Melnik and F. Maurer, On the success of empirical studies in the International Conference on Software Engineering, Inter. Conf. on Software Eng., Shanghai, China (2006) 341-350.
4. Zelkowitz M. V. and D. Wallace, Experimental models for validating computer technology, IEEE Computer 31, 5 (May, 1998) 23-31.