Position paper for 'Workshop on Empirical Studies of Software Maintenance: How to wade through the mire of evidence?' (WESS99)
Session 3: How should we present empirical research?

Raise Public Scrutiny

Lutz Prechelt (prechelt@ira.uka.de)
Fakultät für Informatik, Universität Karlsruhe
D-76128 Karlsruhe, Germany
+49/721/608-4068, Fax: +49/721/608-7343

April 29, 1999

Abstract:

This position paper lists reasons why the results in software maintenance are so difficult to assess and use, proposes three possible countermeasures (what and why), and shortly discusses the roadblocks.

Note: This paper is written in a rather apodictic tone, but that is for brevity only.

Why we have a ***mire*** of evidence

When one dives into the literature on software maintenance, the amount of suggestions and experience reports is overwhelming, the validity of the statements is often dubious (but hard to assess), commonalities between results are difficult to understand and characterize, and contradictions between different reports (which occur frequently) are difficult to resolve. The collective evidence is a morass; it is no wonder that so few organizations exploit research results successfully.

These effects are due to the following problems:

• **Overall volume.** The total number of reports on maintenance is quite large. In principle this is positive, but it also has two drawbacks: Gaining and maintaining a representative overview of the literature is difficult and the quality of some of the contributions is low.

• **Variability.** There is a multitude of rather different maintenance contexts and tasks, which makes understanding commonalities difficult. This is an intrinsic problem, which can possibly be controlled but not be resolved.

• **Incompleteness.** Important information is often missing in reports, in particular with respect to the context of the results, but also about the results themselves or about their meaning.

• **Incomparability.** As a result of the lack of information it is often almost impossible to relate the results from different reports even if they could be comparable in principle.

• **Missing baselines.** It is frequently impossible to understand data in reports because it is free-floating: without any reliable baseline or control group against which to judge it.

• **Ambiguity.** Reports are often ambiguous because some of their important terms have multiple possible definitions (e.g. LOC, module, hour, design, development, testing) but no definition is provided.

• **Overall incompleteness.** Even if one had a complete overview of the literature, the resulting picture would be distorted, because reports on failures are misleadingly rare.

• **Mixing report and interpretation.** In many reports it is not at all clear which parts refer to objective facts (measured, counted, etc.) and which refer to estimations or subjective judgements.

Possible solutions

I conjecture that the following three countermeasures, if implemented to a sufficient degree, would much reduce the problems mentioned above.

• **Clarified terminology.** We need to use well-defined terminology in our reports. Such terminology must describe organizational context, structure of the existing software, the maintenance tasks, and the observed variables (such as cost, errors, etc.) Such terminology could make reports more complete and precise, less ambiguous, and easier to compare and would make it easier to separate interpretation from facts. Due to the high variability of maintenance situations, defining a terminology that is always appropriate is quite difficult. Furthermore, the task here is not only defining such a terminology — partial suggestions do exist —, but also convincing people to use it (and use it correctly). One way of doing this might be to popularize

a restricted language, for instance as provided by the schema of a database of maintenance results. Hence, the next solution proposed below might imply solving the terminology problem as well.

• **Global results database.** Most studies of maintenance produce quantitative (or at least ordinal qualitative) results of one form or another. It would be extremely useful if the researchers collected such results and represented them in a common format in a single database: due to automatic processing capabilities, the large number of reports will then become manageable; the reporters can more easily avoid leaving out information accidentally; the terminology is implicitly unified and ambiguity is reduced through the database schema; and interpretation becomes more clearly separated from raw result reporting.
As a result, large-scale meta-analyses become feasible that will allow for building complex quantitative statistical models for understanding the maintenance process and will also indicate areas where more research is needed.
One remaining problem would be missing baselines, but the database would make this problem clearly visible and would encourage research in appropriate formats such as comparative case studies or even controlled experiments.
The second remaining problem would be the distortion due to missing reports of failures.

• **Encouraging reports of failures.** We need to explicitly call for reports on failed improvements attempts, counter-intuitive results, and results contradicting "common wisdom". Only if we get to read these "odd" results, we will be able to understand our results globally, to form useful predictive models of the maintenance process, and to validate and improve our research methods.

Obstacles

Unfortunately, there are strong roadblocks opposing the implementation of the above-mentioned remedies:

• **Different backgrounds.** The different contexts, approaches, and educational/professional backgrounds of those working in maintenance research make it difficult to find terminology that "feels familiar" and is acceptable for all of them. Worse, the inhabitants of the software field apparently love creating new terms (even unnecessarily) but rarely include proper definitions for them.

• **Lack of resources.** Building and maintaining a results database is a major effort and makes sense only if it can be kept up for many years. It is unclear which organization has sufficient resources. Validating submitted data for ensuring the data quality of the database is technically difficult and politically delicate. Furthermore, the database will only have sufficient submissions if the users trust the competence and endurance of its hosting organization.

• **Lack of individual benefit.** Users will not submit to the database unless they get some noticeable benefit back. It is unclear how to provide such benefit.

• **Inherent difficulty.** Due to the multitude of different characteristics of maintenance situations and maintenance research approaches it will be difficult to define a database schema that can represent most or all result data adequately and unambiguously and that can evolve over time as required by shifting maintenance technology, methods, and circumstances.

• **Trade secrets.** Even if a database was created, most companies would be reluctant to allow sufficiently detailed information be reported. It is unclear to what degree this problem could be overcome if anonymity was guaranteed (which is technically feasible).

• **Short-sighted success orientation.** A strong preference for success, deeply rooted in most cultures, makes it difficult to "admit" and report a failure, even if that will clearly contribute to overall research progress and success in the long run.

Conclusion

Obviously, the situation is not an easy one, because the roadblocks to implementing the solutions are severe. However, perhaps it is time that we direct our frustration about the bad shape of much of our evidence into a serious attempt at designing and deploying a central maintenance research results database — with the side-effect of ameliorating the terminology problem.

Making failures more visible is partly in the hands of editors and conference organizers. A reviewed publication that invites reports on failures (from all parts of computer science and software engineering) already exists: the *Forum for Negative Results* [1].

Acknowledgements

Thanks to Walter Tichy for providing very useful comments on a draft of this position paper.

References

[1] The Forum for Negative Results (FNR). A permanent special section of the Journal of Universal Computer Science (J.UCS). http://wwwipd.ira.uka.de/fnr.