SES '96

FORUM ON
SOFTWARE ENGINEERING STANDARDS ISSUES

HOTEL BONAVENTURE HILTON - MONTRÉAL, QUÉBEC, CANADA
OCTOBER 21-25, 1996

Position Paper

An Experimental Framework for Software Engineering Research

Pierre Bourque
Alain Abran
Software Engineering Management Research Laboratory
Université du Québec à Montréal
Département d'informatique, C.P. 8888, Succ. Centre-Ville
Montréal (Québec) Canada
Téléphone: (514) 987-3000 ext.0315
Fax: (514) 987-9477
email:bourque.pierre@uqam.ca
abran.alain@uqam.ca

1. Introduction

Fundamental principles of software engineering must be supported by theory and sound empirical research. This by itself should be a fundamental principle of software engineering. As in all areas of science and engineering, empirical research can only be considered sound when it is conducted using a valid experimental approach or protocol. This paper proposes a high-level and field tested experimental approach or framework for software engineering. This four-phase experimental framework begins in the definition phase by developing our intuitive understanding of the problem that we wish to solve into a precise specification of an experiment. The sampling design and the appropriate measures are selected in the planning phase. Following this, the actual execution of the experiment and the problems encountered are discussed in the execution phase. Statistical models are developed from the data collected. The results are interpreted in three widening contexts and extrapolation issues are explored in the interpretation phase.

This framework based on the scientific method was proposed by Basili *et al* [BASI86] "to help structure the experimental process and to provide a classification scheme for understanding and evaluating experimental studies" However, to our knowledge, no actual usage of the framework in experiments has been published by their team and no other experimental approach is available for software engineering. We therefore identified the need to field test this framework and to verify its applicability in software engineering experiments conducted in industrial settings.

It proved to be effective in three software engineering experiments [BOUR91, COTE96, MAYA96] and is currently being deployed to all research projects conducted at the Software Engineering Management Research Laboratory of the Université du Québec à Montréal. It is also currently being taught as an approach for conducting software engineering experiments in an industrial setting (CARD96).

We begin by describing the experimentation framework and then present some of the results of field testing. We conclude by identifying some opportunities for the software engineering standards community.

2. Description of the Experimentation Framework

The definition phase of the experiment is the first phase of this framework which is summarized in Table 1. This phase transforms our intuitive comprehension of the problem to be solved into a precise specification of an experiment that can contribute to the problem's solution. The definition of an experiment has six components, namely motivation, object, purpose, perspective, domain, and scope.

Table 1 Summary of the Experimentation Framework

I Definition				
Motivation	**Object**	**Perspective**	**Domain**	**Scope**
Understand	Product	Developer	Software team	1 project 1 team
Assess	Process	Maintainer	Software project	*x* projects 1 team
Manage	Model	Project manager		1 project *y* teams
Learn	Metric	Corporate manager		*x* projects *y* teams
Improve	Theory	Customer		
		User		
		Researcher		
II Planning				
Design		**Direct criteria**	**Measurement**	
Experimental design		Direct criteria	Measure selection and definition	
Statistical analysis methods		Indirect criteria	Data collection methodology	
III Operation				
Preparation		**Execution**	**Data analysis**	
Pilot study		Data collection	Preliminary analysis	
Participant training		Data validation	Formal analysis	
IV Interpretation				
Interpretation context		**Extrapolation**	**Further Research**	
Statistical framework Study purpose Field of research		Sample representativeness	Issues and problems which require further investigation	

Motivation is the reason for tackling the experiment. Motivation identifies what high-level or general issue we are trying to address. Object defines the principal entity being studied. The purpose is the precise objective sought during the experiment. Purpose is the explicit problem that we wish to resolve. Perspective specifies from what point of view the explicit problem will be attacked.

An experiment in software engineering has two domains: teams and deliverables. The project teams (one or more persons) work on projects that attempt to resolve an issue, in terms of a software engineering deliverable (manual, program, specification, test script...). Scope is defined according to the type of sample measured. It affects the general nature of the experiment's conclusion. There are four types: a project carried out by one team only (1-1); a project undertaken by several teams (1-y); several projects, each carried out by one team (x-1); and finally, several projects, each worked on by several teams (x-y).

The experiment is planned in detail in the second step of the framework. During the design step, we select what kind of projects and teams will be included in our samples, then choose the statistical techniques that will be used to analyze our data. We identify the direct and indirect criteria or factors that we believe are related to the study's purpose. We then select measures to quantify these direct and indirect criteria. Finally, the data collection procedures and tools must be established.

The experiment is actually carried out during the third phase of the framework. To reduce the risk, we might want to begin with a pilot project.

Training might be required for the personnel who will be taking the measurements. In some cases, training will also be necessary for the project teams participating in the experiment. This data is first analyzed qualitatively using scatter plots, then more formal analysis is completed using the statistical techniques chosen during the design step.

Within this framework, the interpretation of the results is undertaken in three stages. First we interpret these results on purely statistical grounds. We then go on interpret them in the context of the study purpose and then compare these results to other research in this area. The representativeness of our sample will be the most factor in extrapolating the results to different environmnents and contexts. Lastly, further issues requiring more investigation and new research opportunities are identified.

3. Field Testing of the Framework

This framework has been field tested in three diverse software engineering experiments in industrial settings. These experiments are in the areas of software size estimation [BOUR91], software sizing for small adaptive maintenance requests [MAYA96], and a third experiment which attempts to identify some of the fundamental characteristics of business software using some multidimensional statistical techniques [COTE96].

The framework has shown to be simple to use and understand by all involved parties. We have seen that by encouraging you to properly define what you are trying to attain and to plan it carefully, it certainly has helped us avoid some of the too often seen experiment pitfalls. For example, such pitfalls are jumping into data collection without properly defining what we are trying to measure or collecting data for a certain period of time and then realizing that the right measures have not been collected. It also has encouraged us to interpret the results of our experiments to their maximum and to properly position these results in the context of other research results. It has shown to be an excellent communication and teaching tool when presenting a completed or ongoing experiment. It has also shown to be an excellent tool even though the software measurements were collected prior to our involvement. We are currently conducting all our experimental research following this appraoch.

4. Conclusion

Fundamental principles of software engineering must be supported by theory and sound empirical research. This by itself should be a fundamental principle of software engineering. Sound experimental research must be conducted scientifically by following correct experimental practices. To help us ensure that we are following correct experimental practices, this paper proposes a simple, field tested and easy to use experimental framework for software engineering.

Basili *et al* proposed this framework but to our knowledge no actual usage has been published and no feedback is available on its applicability. We are also not aware of any other experimental framework for software engineering. We have field tested this framework and found it very beneficial. Therefore we believe this framework could form the basis for a standard on software engineering experimentation.

References

[BASI86] V. R. Basili, R. W. Selby and D. H. Hutchens, "Experimentation in software engineering", *IEEE Transactions on software engineering*, Vol. SE-12, No. 7 (1986) pp. 733-743.

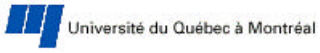
[BOUR91] P. Bourque and V. Côté, "An experiment in software sizing with structured analysis metrics", *Journal of Systems and Software*, Vol. 15 (1991) pp. 159-172.

[CARD96] J. Cardascia, Learning while doing: Research design in a systems engineering environment, *International Function Point Users Group Fall Conference Proceedings*, Dallas (1996).

[COTE96] V. Côté, G. Métivier, P. Bourque, J.-P. Jacquet, Caractérisation des logiciels industriels de gestion, *Actes des neuvièmes journées internationales - Le Génie logiciel et ses applications*, Paris (1996) (To be published).

[MAYA96] P. Bourque, M. Maya, A. Abran, A Sizing Measure for Adaptive Maintenance Work Products, *International Function Point Users Group Spring Conference Proceedings*, Atlanta (1996).

SES'96 HOME AGENDA PROGRAM



Software Engineering Management Research Laboratory