# Classifying and analysing replication packages for software engineering experimentation

Martín Solari[1], Sira Vegas[2]

[1] Facultad de Ingeniería, Universidad ORT Uruguay
Cuareim 1451, 11100 Montevideo, Uruguay + 598 2 902 15 05
martin.solari@ort.edu.uy
[2] Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain +34 91 336 6929
svegas@fi.upm.es

**Abstract.** Replication packages are a means for researchers to share artefacts that improve experiment replication. There is, however, still not very much external replication of software engineering experiments. Researchers taking delivery of replication packages find it difficult to evaluate their content for both adaptation to the new replication setting and estimating the effort that such an adaptation will take. In response to this problem, we propose a framework for improving the evaluation of replication packages. Based on this framework we analyse several cases of available replication packages.

## 1 Introduction

Controlled experiments are one way of increasing and maturing scientific and technological knowledge. From experimentation, we can gather evidence to validate or refute our hypotheses about software development. Despite its importance, only a few of the scientific software engineering publications report controlled experiment [1]. An even smaller number report externally replicated experiments[1].

There are always some validity threats to experiment results. These threats stem from the actual design of the experiment and also the particular setting in which it was run [2]. External replication affords the opportunity to attenuate validity threats stemming from the particular context in which the experiment was run and also to examine other aspects and evolve the research [3].

Replication packages are a means of communication that improves the external replication of an experiment. The replication package is a documentary warehouse of the material needed to perform the replication. A good replication package is structured to improve the exchange of knowledge between researchers, minimizing the resources needed to understand and run the experiment [4].

The goal of our study is to evaluate the maturity of replication packages to find out if more work needs to be done on improving the contents of a package or whether this issue can be regarded as settled.

[1] The external replication of an experiment is an experiment repeated at another site by an independent group of researchers.

## 2 Replication package content types

Because of the diversity of research and the shortage of guidelines on structure, the content of different replication packages (RP) can vary enormously in terms of both quality and quantity. RPs can range from being a short summary of the experiment to containing artefacts and detailed instructions for its replication and even information on other replications run.

RPs generally contain the artefacts used in the experiment, for example, programs, requirements, failures, etc., for replication. Additionally, it describes the factors examined and the parameters used for the experiment setting[5]. RPs are not necessarily confined to exchanging artefacts and may also include descriptions and procedures that facilitate the experiment operation. We suggest the following categories as a means of defining the description of an experiment for the purposes of replication in varying detail:

- **Article** (summary of the experiment and description of major results)
  If the article is not accompanied by a RP, and is the only source of information for replication, whatever information it can provide for replicating the experiment is insufficient. Technical reports sometimes include more details, as they are not subject to the constraints journals or conferences place on space and format.
- **Operational RP** (materials for running the replication)
  The operational level is achieved when the RP includes the artefacts for running the experiment. These artefacts may be software production components, as well as items concerning the running of the experiment: instructions for subjects, forms and measurement instruments.
- **Descriptive RP** (experiment description)
  Even if an experimenter is in possession of all the artefacts, this does not guarantee that an exact replication can be made. For external replication, the researcher who takes delivery of the package needs a description of the items to be able to understand them. Also the design used for the experiment and the setting in which it was run should be described.
- **Procedural RP** (guide for the experimental process)
  To minimize the replication effort, the RP may provide direct step-by-step guidance for experimenters on how to do the replication. This is an improvement on the merely descriptive level in that it provides help on the procedure to be enacted. This level is referred to as a procedural RP.
- **Decision-making RP** (grounded decisions)
  The most detailed level occurs when the RP contains the reasons behind the decisions the original team made on how to run the experiment. The inclusion of such explanations gives the experimenters who intend to replicate an experiment an understanding of the decisions and a basis for their own decision-making.

It should be noted that the detail levels of a RP may vary from one part of the experiment to another. Therefore, a RP can be more thorough in some areas than in others. In the next section we analyse the phases into which the process of experimentation can be divided with a view to analysing the detail of the RP content by phases.

## 3 Experimentation process phases

Experimentation is a process divided into several phases. The key experimentation phases are: definition of the experiment, experimental design, experiment operation and results analysis [6, 7]. The resources required for the experiment are also managed in parallel to these phases.

A number of tasks are carried out in each phase to produce a number of outcomes for the experimental process. The thoroughness of the descriptions found in a RP can vary for each phase. For example, a RP can contain information on how to run the experiment, but provide no data on experimental design or data analysis. The package could also describe some phases in more detail than others.

Therefore, we have decomposed RP content by experimentation phase for classification purposes. We will analyse how thorough the package descriptions are in each phase. Based on the products to be output in each phase, the information we evaluate is:

- Definition: Goal; Research setting; Techniques examined; Constraints
- Design: Factors and parameters; Subject profile (if applicable); Analysis of validity threats
- Operation: Objects (specifications, code, faults, etc.); Instruments; Description of setting; Training material (if applicable); Operating procedure
- Results Analysis: Definition of response variables; Data collection forms; Analysis tools; Guides for interpretation

## 4 Evolution of experimental research

Another point worth considering for evaluating RPs is that an experiment is not an end itself but a means to gain more evidence about a particular subject. Therefore, families of experiments [8], grouped around a research subject, are defined. As more replications are done within a family of experiments, the body of knowledge about the subject under research evolves.

Some phases are repeated every time the experiment is replicated, but there are others that are related to research on a larger scale, of which the replication is just a part. There are then two identifiable phase levels, one for the *experimental research process* and another for the *replication process*. Some of the phases that are part of the large-scale experimental research process are: definition (and retargeting) of research, experiment replication, results aggregation, package improvement. These research-level phases can also be part of the package structure and improved by providing descriptions, tools and procedures.

Assuming that the resources available for and the setting of the replication are unlikely to be the same, the experiment will quite possibly be changed when it is externally replicated. The RP can be built to facilitate the changes and set out the information resulting from these changes.

When an experiment is replicated externally, we assume that this is part of a broader experimental investigation. Therefore, the RP structure should accommodate elements to improve this process. With respect to the support for the experimental research process, we will use the following factors for analysis:

- Support for logging replications: Identification of replications; New factors researched; Reasons behind design changes
- Support for knowledge aggregation: raw data; analysis tools
- Support for package evolution: Package updates; Versioning

## 5 Analysis of replication packages

Based on the proposed structure, we have analysed several software engineering RPs. The only criterion used to select these packages was that they should be published on the web[2].

Table 1 contains a comparison of the analysed packages. It should be noted that, in most cases, there is information published in articles that is omitted from the main body of the package. As it is our intention to focus on the analysis of the RPs as a self-contained tool separate from other documents, we only consider the information that is part of the RP. The blank cells in the table mean that the package does not contain information about the point or that this information is in an article or another document.

**Table 1.** Comparison of replication packages

| Authors | **Lott et al.** | **Basili et al.** | **Thelin et al.** | **Dunsmore** | **Do et al.** |
|---|---|---|---|---|---|
| Research subject | Functional testing, structural testing, static review | PBR (Perspective Based Review) | Usage-based / checklist-based reading | Object-oriented inspection | Regression testing |
| Known replications | U. Kaiserslautern U. Strathclyde | U. Maryland NASA SEL U. Kaiserslautern | U. Lund | U. Strathclyde | U. Nebraska |
| **Support for replication phases** | | | | | |
| Definition | | | | | |
| Goal | | Yes | Yes | | |
| Research setting | | Yes | | | |
| Researched techniques | | Yes | Partial | | |
| Constraints | | | | | |
| Design | | | | | |
| Factors and parameters | | Yes | | | |
| Subject profile (if applicable) | | Partial | Yes | | |
| Validity threats analysis | | | | | |
| Operation | | | | | |
| Objects | Yes | Yes | Yes | Yes | Yes |
| Instruments | Yes | Yes | Yes | Yes | Yes |
| Environment description | | Partial | | | |

[2] Key words used in the search were: *software engineering replication* (alternative *laboratory*) *package*. Then to get the primary results we manually checked to see which actually did contain replication packages. References to replication packages with which we are acquainted were also used.

| | | | | | |
|---|---|---|---|---|---|
| Training material (if applicable) | Yes | Yes | Yes | Yes | |
| Operating procedure | | Partial | Partial | Partial | Partial |
| Results analysis | | | | | |
| Definition of response variables | | Yes | | | |
| Data collection forms | | | | | |
| Analysis tools | | | | | |
| Guidance for interpretation | | Yes | | | |
| **Support for the experimental research process** | | | | | |
| Logging replications | | | | | |
| Identification of replications | | Yes | | | |
| Changes in design | | Yes | | | |
| Support for aggregating knowledge | | | | | |
| Publication of raw data | | | | | |
| Meta-analysis techniques | | | | | |
| Support for evolving the package | | | | | |
| Packages updates | Yes | Yes | | Yes | |
| Versioning | | | | Yes | |
| Packaging and presentation | | | | | |
| Format | tar pdf | html ps | zip pdf fm | zip html doc | html gz |
| Summary and table of contents | Web summary | Web hypertext | Table of contents | Directory structure | Web and directory structure |
| Unified structure | | Yes | | | Yes |
| Availability | Web | Web | Web | Web | Web |

From an overall analysis of the table we can gain an idea of the state of practice:

- Most RPs focus on artefacts for running the experiment.
- Descriptions and guidance for the replication process are less common.
- Information about the replications run and the evolution of the experimental research is seldom integrated into the RP.
- There is no standard structure and a number of different formats are used to organize content.

## 7 Challenges for replication packages

Owing to their importance as instruments of communication between researchers, RPs play a key role in experimental research. We believe that to achieve their goal, RPs face a number of challenges that go beyond just transferring information:

- **Transferring experimental knowledge.** A mature RP should not be a mere collection of artefacts used in the experiment. The descriptions and procedures it contains can promote an effective transmission of knowledge about experimentation and the SE subject under research. Confirmed and refuted hypotheses, analysed SE techniques, together with the design and construction of experimental artefacts, are a way of transmitting very thorough knowledge.
- **Improving experiment replication.** An effective RP is one that minimises the effort it takes to do a replication. The shortage of resources available to experimenters

means that the RP should be designed not just to transfer knowledge but to do so expending the bare minimum of resources. Two factors that improve replication are a good organization and the inclusion of procedures for the different experiment phases.

- **Improving the adaptation of the experiment to the context.** When an experiment is replicated there are often constraints forcing its design to be adapted to the new setting. Exact external replications of experiments in the strictest sense are few and far between. If the setting is different, the factors, levels or parameters of the original experiment may need to be altered. The RP may include guidance for experimenters on how to go about this adaptation. Whoever builds the package should bear in mind that constraints vary from setting to setting and are difficult to foresee.
- **Enabling results aggregation.** The goal of controlled experiments is to find evidence to verify items of knowledge. External replications are a way of raising the confidence in this evidence, lifting validity threats. For this process to be possible, the replication process should be followed by a results aggregation process. The way in which RPs present the information on each replication and particularly the format of results affects the chances of integration.
- **Supporting the evolution of experimental research.** Experimental research is not an end in itself but a means for evolving knowledge. Therefore, an experiment normally changes as research moves forward. When the experiment changes the RP should reflect this evolution. Also it should be capable of capturing the motivation of those changes and the evolution of the experiment across a number of replications.

In other words, RPs should not be considered as static elements that are unchanged after being written. On the contrary, they should be dynamic containers that reflect the evolution of an experimental investigation. The evolutionary dynamics of the experiment also captures knowledge and evidence on the subject under investigation.

## References


[1] D. I. K. Sjoeberg, 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 Transactions on Software Engineering, vol. 31, pp. 733-753, 2005.

[2] S. L. Pfleeger, "Soup or Art? The role of evidential force in empirical software engineering," IEEE Software, vol. 22, pp. 66-73, 2005.

[3] A. Brooks, J. Daly, J. Miller, M. Roper, and M. Wood, "Replication of experimental results in software engineering " ISERN-96-10, 1996.

[4] F. Shull, M. G. Mendonça, V. R. Basili, J. Carver, J. C. Maldonado, S. C. P. F. Fabbri, G. H. Travassos, and M. C. F. d. Oliveira:, "Knowledge-Sharing Issues in Experimental Software Engineering," Empirical Software Engineering, vol. 9, pp. 111-137 2004.

[5] A. Jedlitschka and D. Pfahl, "Reporting guidelines for controlled experiments in software engineering," ISERN-05-01, 2005.

[6] N. Juristo and A. M. Moreno, Basics of Software Engineerig Experimentation: Kluwer, 2001.

[7] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," IEEE Transactions on Software Engineering, vol. 28, pp. 721-734, 2002.

[8] V. R. Basili, F. Shull, and F. Lanubile, "Building knowledge through families of experiments," IEEE Transactions on Software Engineering, vol. 25, pp. 456-473, 1999.