Knowledge Acquisition in Software Engineering Requires Sharing of Data and Artifacts

Dag I.K. Sjøberg

Simula Research Laboratory
P.O. Box 134, NO-1325 Lysaker
NORWAY
dagsj@simula.no

Abstract. An important goal of empirical software engineering research is to cumulatively build up knowledge on the basis of our empirical studies, for example, in the form of theories and models (conceptual frameworks). Building useful bodies of knowledge will in general require the combined effort by several research groups over time. To achieve this goal, data, testbeds and artifacts should be shared in the community in an efficient way. There are basically two challenges: (1) How do we encourage researchers to use material provided by others? (2) How do we encourage researchers to make material available to others in an appropriate form? Making material accessible to others may require substantial effort by the creator. How should he or she benefit from such an effort, and how should the likelihood of misuse be reduced to a minimum? At the least, the requester should officially request *permission* to use the material, *credit* the original developer with the work involved, and provide *feedback* on the results of use as well as problems with using the material. There are also issues concerning the *protection* of data, *maintenance* of artifacts and *collaboration* among creators and requestors, etc. A template for a data sharing agreement between the creator and requestor that addresses these issues has been proposed.

Introduction

A prerequisite for the evolution of most sciences is that researchers build on the work of other researchers. In empirical sciences, this includes the sharing of data and experimental material. For example, to evaluate, compare and generalize results from empirical studies, one should replicate them, and preferably develop theories or models that represent the current knowledge in the field. If replication of studies, meta-analysis, theory development and other research that builds on others' work is stimulated by editors, program chairs and reviewers of journals and conferences, this would be an incitement for individuals to reuse material produced by others.

This chapter is organised as follows. The next section motivates, within the context of empirical software engineering, the need for more replication of studies conducted by other than the original researcher, and the need for more theory building. To support rapid progress in these areas, an increase in the sharing of data and artifacts among software engineering researchers would be required. However, there are several challenges to such a sharing, which is the topic of the subsequent section. Then follow a section that describes a concrete proposal for a template for a data sharing agreement that may help ensure that the reuse of data and material is performed in a way that is satisfactory for both creators and users. The conclusion section ends the chapter.

Motivation

The purpose of this section is to demonstrate two important areas in empirical software engineering, replication and theory building, whose progress is dependent on the sharing of data and artifacts among several researchers.

Replication of Studies

"Methodological authorities generally regard replication, or what is also referred to as "repeating a study," to be a crucial aspect of the scientific method" [1]. In a literature survey, 113 experiments were identified in 103 articles extracted from of a total of 5,453 articles published in major software engineering journals and conferences in the decade 1993-2002 [2]. Only 18 percent of the surveyed experiments were replications. Moreover, of the 20 replications, five can be considered as close replications in the terminology of Lindsay and Ehrenberg [1], i.e., one attempts to retain, as much as is possible, most of the known conditions of the original experiment. The other replications are considered to be differentiated replications, i.e., they involve variations in essential aspects of the experimental conditions. One prominent variation involves conducting the experiment with other kinds of subject (for example, professionals instead of students, undergraduates instead of graduates, etc.), application system, task, etc. Table 1 shows that experiments that are replicated by the same authors tend to confirm the results of the original experiments, and experiments that are replicated by others tend to have different results. This may indicate that when you replicate your own experiments, it is difficult not to be biased. Consequently, replications should preferably be conducted by others.

Table 1. Proportion of differentiated replicated studies that confirm result of the original study

Result	Same authors	Other authors	Total
Confirmation	7	1	8
Different results	1	6	7
Total	**8**	**7**	**15**

Theory and Model Building

There are many arguments in favour of theory use, such as structuring, conciseness, precision, parsimony, abstraction, generalisation, conceptualisation and communication [3,4,5]. Such arguments have been voiced in the software engineering community as well [6,7,8,9]. Theory provides explanations and understanding in terms of basic concepts and underlying mechanisms, which constitute an important counterpart to knowledge of passing trends and their manifestations. When developing better software engineering technology for long-lived industrial needs, building theory is a means to go beyond the mere observation of phenomena, and to try to understand *why* and *how* these phenomena occur. In general, hypotheses in software engineering are often isolated and have generally a much simpler form than has a theory:

Hypothesis: Technology (process, method, technique, tool language) *A* is better than technology *B*

Theory: When and why is *A* better than *B,* and how much? Depending on category of developers, tasks, systems, support materials and technology, company culture and other environmental factors, *A* is *X* percent better than *B*, because … etc.

A systematic review of the explicit use of theory in the set of 103 articles reporting controlled experiments (described above), was conducted by Hannay *et al.* [10]. Of the 103 articles, 24 use a total of 40 theories in various ways to explain the cause-effect relationship(s) under investigation. Only two of the extracted theories are used in more than one article, and only one of these is used in articles written by different authors. Hence, there is little sharing of empirically-based theories within the software engineering community, even within topics.

Challenges of Sharing Data and Artifacts

We have identified several challenges related to the sharing of artifacts [11]. They concern the *permission* to use the items, the *credit* that should be given to the original creator, the opportunities for *collaboration* between the original creator and the requestor, the kind of *feedback* on the results of use, as well as problems with using the artifact or data that should be reported to the original creator, the *protection* of the data and artifacts, and the *maintenance* of these artifacts:

Permission: Does one have to request permission to use the material? Is it simply publicly available? What should be the rules? If publicly available, how (or should) one provide some form of controlled access to the artifacts? There might be a request to use the artifact with a commitment to provide feedback after or during use (method, results, other data) and reference the items in all work using them. A mechanism that could effectively restrict access would be to require that the requestor write a short proposal to the data owner. Then the item can be used:

- freely, in the public domain,
- with a data sharing agreement or license, or
- with a service fee for use (by industry) to help maintain the data.

Credit*:* How should the original group gathering the data or developing the artifact be given credit? What would be the rewards for the artifact or data owner? The type of credit is related to the amount of interaction. If there is an interaction, depending on the level, co-authorship may be of value. If it is used without the support of the data owner, some credit should still be given, e.g., acknowledge and reference the data owner. Thus, if the requestor uses it but the owner is not interested in working on the project, the minimal expectation is a reference or an acknowledgement. (There are various possibilities for how that reference should be made, e.g., the paper that first used the artifact, a citation provided by the artifact or data developer, or some independent item where the artifact itself exists a reference.) It is also possible that some form of "associated" co-authorship might be appropriate.

Collaboration*:* In general, it has been suggested that the requestor keep the option open of collaboration on the work. Funding agencies are often looking for "new" ideas, so it is often difficult to be funded for a continuing operation. What options are there for funding collaborations? If collaboration is not desired by the owner of the artifacts, what are the rights of the requestor? It is probably too strong to require collaboration as a requirement for any requestor.

Feedback*:* By requiring permission, there is a sense that the originators of the material know that someone is using their materials. However, some form of feedback can act as payment, i.e., updated versions of artifacts, data so it can be used in some form of meta-analysis, some indication of the effectiveness of technology on the experimental environment. A related issue is assuring that the quality of the data, analysis, and new knowledge being returned to the originator is acceptable and consistent within the context of the original experiment.

Protection*:* There are a large number of issues here. How does one limit potential misuse? How does one support potential aggregation and assure it is a valid aggregation. How does one deal with proprietary data? What about confidentiality? What is required of the originators? Should they be allowed to review results before a paper is submitted for external publication? Does the artifact owner have any rights to stop publication of a paper with invalid results based upon the original artifacts or is the "marketplace of ideas" open to badly written papers? Should there be some form of permission required by reviewers? Who has the rights to analyze and synthesize and create new knowledge based upon the combined results of multiple studies? Again here, how is credit given, authorship determined? How does one limit potential misuse? On the other hand, how do we protect scientific integrity? If users of data find gross negligence on the part of those who created it, what are their obligations to reveal those issues (e.g., the South Korean scandal over stem cell research[1])? Can licensing requirements be an impediment imposed by the guilty to hide their actions?

Maintenance*:* A large physical device (e.g., particle accelerator) generally is built and supported over the long term. But the same has not been true of computer software, which has an ethereal quality of simple residing hidden in a computer file system. Who pays for the cost of maintaining the experience base? There are only three possibilities here toward maintenance: (1) Owner of the data, (2) Users via a

[1] http://en.wikipedia.org/wiki/Hwang_Woo-Suk

licensing fee, (3) Everyone via an open source arrangement. “Owner of the data” generally will not work since few have such resources, ”Licensing fee” may work, but costs will limit use; researchers will not generally pay for something they view as a “free resource.” “Open source” is a possibility.

A proposal for a Data and Artifact Sharing Agreement

To help address the challenges described above, a template for a data and artifact sharing agreement has been proposed [11]. The template is shown in Table 2. It has been developed on the basis of experiences from several projects in which data and artifact sharing has been undertaken [11]. I have used it successfully myself in two recent projects: one small project where most of the attributes were not considered relevant, and another, larger project in which we had to include many details that were not in the template. Note that the purpose of the template is to provide a general framework that in most cases would need adaptations depending on the actual project.

Table 2. Data/artifact sharing agreement taxonomy

Attribute	Property	Value	Definition
Lifetime	Permission	Single use	Can use artifact only for one application
		Limited	Can use artifact repeatedly for a set period of time
		Unlimited	Unlimited use of the artifact
Area	Permission	Specific project	Can use artifact only for one project
		Specific research	Can use artifact within one research area
		Unlimited	Unlimited use of the artifact
Data	Protection	Sanitized	No personal information contained
		Proprietary	Data contains information that uniquely identifies individuals of specific organizations
Transfer to 3rd party	Permission	No	Only signer of agreement can use artifact
		Yes	Signer of agreement can pass on artifact under the same agreement conditions to another. This may require a non-disclosure agreement with either this signer or owner of artifact.
		Yes after period	Signer of agreement can pass on artifact after a period of time (e.g., restricted for 3 years then available to anyone)
Publication	Credit, Feedback	None	Signer of agreement is free to use artifact in any way.
		Prior results	Signer of agreement has to send results of using artifact to owner of artifact prior to writing a paper on the topic
		Acknow-ledge	Signer of agreement has to acknowledge creator of artifact in publication. Agreement will state how this acknowledgement will occur.
		Review	Artifact owner has rights to review paper based on artifact prior to publication submission
Help	Collabo-ration	Data only	Signer of agreement obtains the data “as is.” No help is provided from artifact owner.
		Limited	Artifact owner is willing to provide limited help to signer of agreement to use artifact.
		Extensive	Artifact owner is willing to provide significant collaboration and may want to be co-author on publications.
Costs	Maintenance	None	Artifact is free to signer of agreement, with perhaps a minimal cost for a tape or CD of data
		Payment	A set amount is specified to obtain artifact. If successful, this may help provide funding for maintenance of artifact repository.
Derivatives	Permission, Feedback, Protection, Maintenance	None	Derived artifact is owned by signer of agreement. (May be separate clauses covering derived software and related artifacts or derived data using meta-analysis)
		Creator	Derived artifact is owned by original artifact creator and creator must get a copy of the derived artifact.
		Open-source	An agreement such as used by the open source community from the Free Software Foundation. Any derived work has the same usage requirements as the original artifact.

Conclusions

To make progress in empirical software engineering, we need to build on the work of each other; we need to share data, testbeds and other artifacts. The proposed basic data sharing agreement must evolve based on the feedback from actual use. Hence, we hope that as many as possible will start using this template and report experiences and suggestions for change to the author of this chapter or one of the authors of [11].

References

1. R.M. Lindsay and A.S.C. Ehrenberg, The Design of Replicated Studies, The American Statistician, vol. 47, pp. 217-228, Aug. 1993.
2. D.I.K. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanović, N.-K. Liborg, and A.C. Rekdal. A survey of controlled experiments in software engineering. IEEE Transaction on Software Engineering, 31:733–753, September 2005.
3. S.B. Bacharach. Organizational theories: Some criteria for evaluation. Academy of Management Review, 14(4):496–515, 1989.
4. J.W. Lucas. Theory-testing, generalization, and the problem of external validity. Sociological Theory, 21:236–253, 2003.
5. D.G. Wagner. The growth of theories. In M. Foschi and E.J. Lawler, editors, Group Processes, pages 25–42. Nelson–Hall Publishers, Chicago, 1994.
6. V.R. Basili. Editorial. Empirical Software Engineering, 1(2), 1996.
7. A. Endres and D. Rombach. A Handbook of Software and Systems Engineering. Empirical Observations, Laws and Theories. Fraunhofer IESE Series on Software Engineering. Pearson Education Limited, 2003.
8. 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 Transaction on Software Engineering, 28(8):721–734, August 2002.
9. W.F. Tichy. Should computer scientist experiment more? 16 excuses to avoid experimentation. IEEE Computer, 31(5):32–40, May 1998.
10. J.E. Hannay, D.I.K. Sjøberg, T. Dybå, A Systematic Review of Theory Use in Software Engineering Experiments, accepted for publication in IEEE Transaction on Software Engineering, 2006.
11. V. Basili, M. Zelkowitz, D. I.K. Sjøberg, P. Johnson and T. Cowling, Protocols in the use of Empirical Software Engineering Artifacts, accepted for publication in Empirical Software Engineering. 2006.