

Case Studies for Software Engineers

Dewayne E. Perry
The University of Texas at Austin
perry@ece.utexas.edu

Susan Elliott Sim
University of California, Irvine
ses@ics.uci.edu

Steve M. Easterbrook
University of Toronto
sme@cs.toronto.edu

Abstract

The topic of this full-day tutorial was the correct use and interpretation of case studies as an empirical research method. Using an equal blend of lecture and discussion, it gave attendees a foundation for conducting, reviewing, and reading case studies. There were lessons for software engineers as researchers who conduct and report case studies, reviewers who evaluate papers, and practitioners who are attempting to apply results from papers. The main resource for the course was the book Case Study Research: Design and Methods by Robert K. Yin. This text was supplemented with positive and negative examples from the literature.

1. Introduction

Case studies are a powerful and flexible empirical method. They are used for primarily for exploratory investigations, both prospectively and retrospectively, that attempt to understand and explain phenomenon or construct a theory. They are generally observational or descriptive in nature, though they can be relational as well. They can also be used in the validation of research results. Due to this dexterity, they have become popular in software engineering and are frequently used in papers to understand, to understand explain or to demonstrate the capabilities of a new technique, method, tool, process, technology or organizational structure. Unfortunately, they are usually not used to their full potential, and often not used correctly. The aim of this full-day tutorial was to teach software engineering researchers and professionals how to effectively design, conduct, evaluate and read case studies.

2. Characteristics of Case Studies

A case study is an empirical method. By this we mean a defined, scientific, method for posing research questions, collecting data, analyzing the data, and presenting the results. Each of these steps is planned from the outset of the study and do not come about through serendipity. Case studies are well-suited to “how” and “why” questions in settings where the researcher does not have control over variables and there is a focus on contemporary events.

Unfortunately, there is a great deal of confusion regarding the term “case study” within software engineering. Some of these misuses of the term are understandable because it has different meanings in different settings or disciplines. For the remainder of this Section, we will clarify what case studies are *not*.

A case study is not an exemplar or case history. The term case study is frequently used in medicine and law. Patients or clients are referred to as “cases,” so a study of interesting instances of these are sometimes called case studies [1]. However, empirical studies conducted using a case study method are very different from the interesting examples that practitioner-researchers encounter. In addition, a report on something interesting that was attempted by researchers on a toy problem is not a case study.

A case study is not an experience report. The latter is a retrospective report on an experience that was particularly illuminating and best examples of these include lessons learned. However, even exploratory case studies need to start out with a research question and systematically collect and analyze data to answer the initial question. This confusion is very common as the Experience Reports track of ICSE 2003 had a session on Case Studies.

A case study is not a quasi-experimental design with $n=1$. While some quasi-experimental studies are conducted in the field, they still retain control over some independent variables, so that time series designs, non-equivalent before-after designs, and *ex post facto* designs can be brought to bear on the research question [2]. Finally, a case study is equivalent in scope to a single experiment, and both need a series of studies to fully understand a phenomenon and produce results that generalize.

3. Goals of the Tutorial

The purpose of this tutorial was to help software engineers understand and avoid and identify common mistakes with case studies by giving them a solid grounding in the fundamentals and principles of case studies as a research method. For researchers, our goal was to provide them a starting point for learning how to conduct case studies. When they return to their home

institutions, they would be able to find, assess, and apply appropriate resources in designing their studies. For reviewers, our goal was to provide them with guidance on how to judge the quality and validity of reported case studies. Reviewers They would be able to use the criteria presented in this tutorial to assess whether research papers based on case studies are suitable for publication, allowing them to raise the quality of publications and give appropriate feedback to authors. For practitioners, our goal was to provide a better awareness of how to interpret the claims made by researchers about new software engineering methods and tools. We also aimed to offer practitioners deeper insights into the roles they can play in designing and conducting case studies in collaborative research projects, and the ability to read case studies more effectively and be better able to identify results suitable for use in their workplace.

4. Format and Curriculum

During this full-day tutorial, time was divided evenly between lecture and discussion. The lectures drew on our experience with empirical studies, research methodology texts, and papers from the software engineering literature.

The tutorial covered a range of topics on the design and implementation of case studies. It started with basic issues in common to all empirical studies, moved on to issues ones particular to case studies, and concluded with an examination of practical issues. Students will gain experience designing and evaluating case studies.

The curriculum included the following topics.

- Research Methodology
 - Strategies for Software Engineering Research
 - Approaches for Empirical Studies
- Case Study Fundamentals
 - Exploratory Questions
 - Validation
- Designing Case Studies
 - Research Context
 - Validity
 - Ethical Issues
 - Data Gathering and Analysis
- Publishing Case Studies
 - Preparing Evidence
 - Elements of the Report
- Reviewing Case Studies
 - Replication

The primary text text used for the course tutorial was *Case Study Methods 3/e*, by Robert K. Yin [3]. This book is a respected resource on case studies and is widely cited both inside and outside software engineering.

The lessons were reinforced by small group sessions where participants examined and discussed case studies that have been published in software engineering conferences and journals.

The following papers, in our opinion, are exemplary research case studies:

Matthias M. Müller and Walter F. Tichy, “Case Study: Extreme Programming in a University Environment,” presented at Twenty-third International Conference on Software Engineering, Toronto, Canada, pp. 537-544, 12-19 May 2001.

Carolyn B. Seaman and Victor R. Basili, “An Empirical Study of Communication in Code Inspections,” presented at Nineteenth International Conference on Software Engineering, Boston, MA, pp. 96-106, 17-23 May 1997.

D.N. Card, V.E. Church, and W.W. Agresti, “An Empirical Study of Software Design Practices,” *IEEE Transactions on Software Engineering*, vol. 12, no. 2, pp. 264-271, 1986.

Sallie M. Henry and Dennis G. Kafura, “Software Structure Metric Based on Information Flow,” *IEEE Transactions on Software Engineering*, vol. 7, no. 5, pp. 545-522, September, 1981.

During the break-out sessions, theThe tutorial was divided into three discussion groups, each led by one of the instructors. These smaller groups increased the amount of interaction and allowed the material to be tailored to the students. [At time of writing, we were planning to have tracks for investigators, reviewers, and practitioners, however, this may change depending on the demographics of the tutorial attendees].

5. Conclusion

Case studies are an empirical method in their own right, with their own established internal logic, and design principles. Even for studies that are properly called case studies, there are often problems with selecting a unit of analysis, validity of results, data observation and collection. This tutorial sought to address these issues, because case study is a method that is well-suited to software engineering. It is particularly appropriate when we seek to understand how and why technology is used or not used, functions or does not function in contemporary settings, and where we have little or no control over the variables. Our discipline can only be improved by the addition of high-quality, published case studies that employ solid methods and produce innovative results.

6. References

- [1] Blanche Geer, Everett C. Hughes, Anselm L. Strauss, and Howard Saul Becker, *Boys in White: Student Culture in Medical School*: Transaction Publications, 1991.
- [2] William J. Ray, *Methods Toward a Science of Behavior and Experience, Fourth Edition*. Pacific Grove, CA: Brooks/Cole Publishing Company, 1993.

[3] Robert K. Yin, *Case Study Research: Design and Methods*, 3/e. Thousand Oaks, CA: Sage Publications, 2002.