Principles of Survey Research
Part 1: Turning Lemons into Lemonade

Shari Lawrence Pfleeger
Systems/Software, Inc.
Washington, DC, USA
s.pfleeger@ieee.org

Barbara A. Kitchenham
Department of Computer Science
Keele University, Staffordshire, UK
barbara@cs.keele.ac.uk

Setting the scene

Surveys are probably the most commonly-used research method world-wide. Survey work is visible not only because we see many examples of it in software engineering research, but also because we are often asked to participate in surveys in our private capacity, as electors, consumers, or service users. This widespread use of surveys may give us the impression that survey-based research is straightforward, an easy option for researchers to gather important information about products, context, processes, workers and more. In our personal experience with applying and evaluating research methods and their results, we certainly did not expect to encounter major problems with a survey that we planned, to investigate issues associated with technology adoption. This article and subsequent ones in this series describe how wrong we were. We do not want to give the impression that there is any way of turning a bad survey into a good one; if a survey is a lemon, it stays a lemon. However, we believe that learning from our mistakes is the way to make lemonade from lemons. So this series of articles shares with you our lessons learned, in the hope of improving survey research in software engineering.

We began our investigation by reviewing what is known about technology transfer. We found that, a few years ago, Zelkowitz, Wallace and Binkley (1998) surveyed practitioners to determine their confidence in different types of empirical evaluations as the basis for technology adoption decisions. Their findings confirmed what we have long suspected: the evidence produced by the research community to support technology adoption is not the kind of evidence being sought by practitioners. To build on Zelkowitz et al.'s work, we contacted the publisher of Applied Software Development, a newsletter whose readership included primarily software project managers. We wanted to do a follow-up survey of managers, to find out what kinds of evaluations they make of proposed technologies, and what kinds of evidence they rely on for their technology decisions. However, when we administered our survey

and analyzed the results, we realized that we had made errors in survey design, construction, administration and analysis that rendered our results inconclusive at best.

Nevertheless. our lemon of a survey gave us some useful insights into the right and wrong ways to do survey research. This article is the first of six parts about how to do useful survey research. We use our own work, plus the work of several other researchers, to illustrate the attractions and pitfalls of the survey technique.

To understand how surveys fit in the larger scheme of empirical investigation, we invite you to read the series we wrote, starting in December 1995, in Software Engineering Notes. There, we described several empirical methods, including case studies, factor analyses, experiments and surveys, to help you decide which technique is appropriate in which situation. Here, we assume you have decided to do a survey, and we focus on how to organize, administer and analyze one so that you get useful, meaningful results.

What our series will discuss

This first installment of our series describes three surveys: two reported in the literature and the one we attempted recently. We provide you with a general overview of each survey; we will use particulars of each survey in subsequent installments as examples of the state of the practice and how it might be improved. We also suggest general reference material for studying survey techniques.

Subsequent installments will focus on these issues:

- Survey design, including discussions of sample size and response rate
- Questionnaire design and construction
- Issues in survey administration, including motivating respondents, surveyor bias, pre- and pilot tests, survey documentation, and survey reliability and validity
- Population and sampling
- Data analysis.

What is a survey?

To begin, let us review exactly what a survey is. A survey is not just the instrument (the questionnaire or checklist) for gathering information. It is a comprehensive system for collecting information to describe, compare or explain knowledge, attitudes and behavior. Thus, the survey instrument is part of a larger survey process with clearly-defined activities:

1. Setting specific, measurable objectives
2. Planning and scheduling the survey
3. Ensuring that appropriate resources are available
4. Designing the survey
5. Preparing the data collection instrument

6. Validating the instrument
7. Selecting participants
8. Administering and scoring the instrument
9. Analyzing the data
10. Reporting the results.

There are several different types of surveys. Surveys can be either supervised or not, depending on the objectives and the resources available. If supervised, we can assign one survey researcher to each respondent, to ensure that the respondent understands each question and provides an answer. Telephone interviews are often of this type, where a questioner works one-on-one with a respondent to elicit answers. Or a survey can be administered to a group, with a survey researcher available to clarify and elaborate on the instructions in the survey instrument. Some surveys are semi-supervised, where a researcher explains the objectives and format, perhaps working through some sample questions, but then leaves the respondents to provide information on their own.

We are all familiar with unsupervised surveys: the automated-voice telephone calls in the middle of dinner, or the mailed questionnaire (sometimes with a prize or a pound coin) requesting responses about life style, travel habits, or food preferences, for example.

As noted above, the first step in beginning any survey research (or any research, for that matter!) is setting objectives. Each objective is simply a statement of the survey's expected outcomes. For instance, a survey may hope to identify the most useful features of a front-end development tool, or the most common training needs for new hires. It is very important that the statement of objectives include definitions of all potentially ambiguous terms. These definitions drive not only the instrument's design and development but also the respondents' understanding of the terminology.

Where do these objectives come from? Sometimes they are derived from a perceived need. Someone in your organization wants to know something about the organization, the state of the practice, the effectiveness of a new technology, or the experience base of its members, for example. At other times, an objective may be determined by a literature search to find out who is writing about which new technologies and where there are gaps in current knowledge. Experts can also provide objectives. For example, researchers planning a survey about software testing technologies may initially canvass testing experts to determine what information is needed about which technologies. This kind of solicitation is sometimes done in a focus group, where participants brainstorm about future possibilities or different outcomes. When there is disagreement about what the objectives should be, sometimes we form a consensus panel to encourage people to converge on one or two ideas.

It is essential to devote enough time to the objectives so that they are clear and measurable, because the objectives are essential for all subsequent survey process activities. First, the objectives determine what the survey will ask, of what population, and what information will be collected. Second, the objectives are usually rephrased as research questions or hypotheses, and they help to suggest which are the dependent and independent variables of the investigation. Third, the objectives confirm whether a survey is the appropriate type of empirical investigation. That is, if the objectives are fuzzy, then we may require a different research method.

Description of our survey

To see how easy it can be to fall into several survey traps, it is useful to understand the evolution of our own survey. We had noticed that many newsletters often include reader survey forms, some of whose questions and answers could provide useful insight into managers' decision-making processes. We approached the publisher of *Applied Software Development*; he was eager to cooperate with the research community, and he agreed to insert a one-page survey in the newsletter and gather the responses for us. As a result, we took the following steps:

1. We designed a survey form and asked several of our colleagues to critique it. The survey asked respondents to examine a list of technologies and tell us if the technology had been evaluated and if it had been used. If it had been evaluated, we asked the respondents to distinguish between a "soft" evaluation, such as a survey or feature analysis, and a "hard" evaluation, such as formal experiment or case study.
2. We "tested" the resulting survey form on a colleague at Lucent Technologies. We asked him to fill out the survey form and give us feedback on the clarity of the questions and responses, and on the time it took him to complete the form. Based on his very positive reaction to our questionnaire, we submitted a slightly revised survey to the newsletter publisher.
3. The publisher then revised our survey, subject to our approval, so that it would fit on one page of his newsletter.
4. The survey form was included in all copies of a summer 1999 issue of *Applied Software Development*.

Of the several thousand possible recipients of *Applied Software Development*, only 171 responded by sending their survey form back; thus, our response rate was low, which is typical in this type of survey. The staff at *Applied Software Development* transferred the data from the survey sheets to a spreadsheet, which became the basis for our analysis.

Description of the Lethbridge survey

In 1998, Lethbridge (1998 and 2000) conducted surveys to help him understand those areas where practitioners feel they need more or better education. The goal of the surveys was to provide information to educational institutions and companies as they

plan curricula and training programs. A secondary goal involved providing data that will assist educators and practitioners in evaluating existing and proposed curricula.

Lethbridge and his team recruited participants for the surveys in two ways: by approaching companies directly and asking them to participate, and by advertising for participants on the Web. To determine the effects of formal education, Lethbridge presented the respondents with a list of topics related to computer science, mathematics and business. For each topic, the respondent was asked "How much did you learn about this in your formal education?" The choices for answers ranged on a six-point Likert scale from "learned nothing" to "learned in depth." Other questions included

- What is your current knowledge about this considering what you have learned on the job as well as forgotten?
- How useful has this specific material been to you in your career?
- How useful would it be (or have been) to learn more about this (e.g. additional courses)?
- How much influence has learning the material had on your thinking (i.e. your approach to problems and your general maturity), whether or not you have directly used the details of the material? Please consider influence on both your career and other aspects of you life.

Description of the Finnish study

Recently, Ropponen and Lyytinen (2000) described their examination of risk management practices. They administered a survey addressing two overall questions:

- What are the components of software development risk?
- What risk management practices and environmental contingencies help to address these components?

To find out the answers, the researchers mailed a questionnaire to each of a pre-selected sample of members of the Finnish Information Processing Association whose job title was "manager" or equivalent. However, they sent the questionnaire to at most two managers in the same company.

Ropponen and Lyytinen asked twenty questions about risk by presenting scenarios and asking the respondents to rate their occurrence with a five-point Likert scale, ranging from "hardly ever" to "almost always." For example, the scenarios included

"Your project is canceled before completing it."

and

"Subcontracted tasks in the project are performed as expected."

The researchers posed additional questions relating to organizational characteristics, such as the organization's size, industry, type of systems developed, and contractual arrangement. They also sought technology characteristics, such as the newness of the technology, the complexity and novelty of technological solutions, and the process technologies used. Finally, they asked questions about the respondents themselves: their experience with different sizes of projects, their education, the software used, and experience with project management.

References for survey techniques

Subsequent installments of this series will focus on survey research in software engineering. To find more general information about surveys, we recommend the following publications.

Arlene Fink, *The Survey Handbook*, Sage Publications, 1995.

> This comprehensive book explains how to ask survey questions, how to design surveys, how to sample in surveys, how to analyze survey data, and how to report on surveys.

Linda Bourque and Eve Fielder, *How to Conduct Self-administered and Mail Surveys*, Sage Publications, 1995.

> This book looks in detail at self-administered and mail surveys.

Mark Litwin, *How to Measure Survey Reliability and Validity*, Sage Publications, 1995.

> This book defines survey reliability and validity and explains how to measure them.

Bibliography

Lethbridge, Timothy, "A survey of the relevance of computer science and software engineering education," *Proceedings of the 11th International Conference on Software Engineering*, IEEE Computer Society Press, 1998.

Lethbridge, Timothy, "What knowledge is important to a software professional," *IEEE Computer*, May 2000.

Ropponen, J. and K. Lyytinen, "Components of software development risk: How to address them. A project manager survey," *IEEE Transactions on Software Engineering* 26(2), February 2000.

Zelkowitz, Marvin V., Dolores R. Wallace and David Binkley, "Understanding the culture clash in software engineering technology transfer," University of Maryland technical report, 2 June 1998.

Principle 22. â€œBlessing in disguiseâ€ or â€œTurn Lemons into Lemonadeâ€. Use harmful factors (particularly, harmful effects of the environment or surroundings) to achieve a positive effect. Use waste heat to generate electric power.Â Same for airplane parts. Fiberglass surfboards are lighter and more controllable and easier to form into a variety of shapes than wooden ones. Name. Email. Turning lemons into lemonade. Blog Post by Brad W. Setser. August 30, 2007.Â I suspect â€“ and I am certainly not an expert on this, only an interested observer â€“ that a lot of this complexity is central to a certain part of the securitization process. After all, the idea behind a CDO is that by combining a whole bunch of different â€“ and hopefully uncorrelated â€“ payment streams, you can create a new security that is less risky than the original securities. All the payments go into a common pot, and someone â€“ the holder of the equity tranche â€“ agrees to take all of the losses. The holder of the equity tranche is betting that nothing much will default. If all goes well, th