Experimental Design and Analysis in Software Engineering

Part 2: How to Set Up an Experiment

Shari Lawrence Pfleeger
Centre for Software Reliability
City University
Northampton Square
London EC1V 0HB England
phone: +44-71-477-8426
fax: +44-71-477-8585
shari@csr.city.ac.uk

Formal experiments, like software development itself, require a great deal of care and planning if they are to provide meaningful, useful results. In the first tutorial, we examined why you might want to perform an experiment. Here, we discuss the planning needed to define and run a formal experiment, including consideration of several key characteristics of the experiment.

THE STEPS OF EXPERIMENTATION

There are several steps to carrying out a formal experiment:

- conception
- design
- preparation
- execution
- analysis
- dissemination and decision-making

We discuss each of these steps in turn.

Conception

The first step is to decide what you want to learn more about, and define the goals of your experiment. The conception stage includes the type of analysis described in the last tutorial, to ensure that a formal experiment is the most appropriate research technique for you to use. Next, you must state clearly and precisely the objective of your study. The objective may include showing that a particular method or tool is superior in some way to another method or tool. Alternatively, you may wish to show that, for a particular method or tool, differences in environmental conditions or quality of resources can affect the use or output of the method or tool. No matter what you choose as your objective, it must be stated so that it can be clearly evaluated at the end of the experiment. That is, it should be stated as a question you want to answer. Then, the next step is to design an experiment that will provide the answer.

Design

Once your objective is clearly stated, you must translate the objective into a formal hypothesis. Often, there are two hypotheses described: the null hypothesis and the experimental (or alternative) hypothesis. The null hypothesis is the one that assumes that there is no difference between two treatments (that is, between two methods, tools, techniques, environments, or other conditions whose effects you are measuring) with respect to the dependent variable you are measuring (such as productivity, quality or cost). The alternative hypothesis posits that there is a significant difference between the two treatments. For example, suppose you want to find out if the cleanroom technique of developing software produces code of higher quality than your current development process. Your hypotheses might be formulated as follows:

Null hypothesis: There is no difference in code quality between code produced using cleanroom and code produced using our current process.

Alternative hypothesis: The quality of code produced using cleanroom is higher than the quality of code produced using our current process.

It is easy to tell which hypothesis is to be the null hypothesis and which the alternative. The distinguishing characteristic involves statistical assumptions: the null hypothesis is assumed to be true unless the data indicates otherwise. Thus, the experiment focuses on departures from the null hypothesis, rather than on departures from the alternative hypothesis. In this sense, "testing the hypothesis" means determining whether the data is convincing enough to reject the null hypothesis and accept the alternative as true.

Hypothesis definition is followed by the generation of a formal design to test the hypothesis. The experimental design is a complete plan for applying differing experimental conditions to your experimental subjects so that you can determine how the conditions affect the behavior or result of some activity. In particular, you want to plan how the application of these conditions will help you to test your hypothesis and answer your objective question.

To see why a formal plan or design is needed, consider the following objective for an experiment:

> We want to determine the effect of using the Ada language on the quality of the resulting code.

The problem as stated is far too general to be useful. You must ask specific questions, such as

- How is quality to be measured?
- How is the use of Ada to be measured?
- What factors influence the characteristics to be analyzed? For example, will experience, tools, design techniques, or testing techniques make a difference?
- Which of these factors will be studied in the investigation?
- How many times should the experiment be performed, and under what conditions?
- In what environment will the use of Ada be investigated?
- How should the results be analyzed?
- How large a difference in quality will be considered important?

These are just a few of the questions that must be answered before the experiment can begin.

There is formal terminology for describing the components of your experiment. This terminology encourages you to consider all aspects of the experiment, so that it is completely planned to ensure useful results. The new method or tool you wish to evaluate (compared with an existing or different method or tool) is called the treatment. You want to determine if the treatment is beneficial in certain circumstances. That is, you want to determine if the treatment produces results that are in some way different. For example, you may want to find out whether a new tool increases productivity compared with your existing tool and its productivity. Or, you may want to choose between two techniques, depending on their effect on the quality of the resulting product.

Your experiment will consist of a series of tests of your methods or tools, and the experimental design describes how these tests will be organized and run. In any individual test run, only one treatment is used. An individual test of this sort is sometimes called a trial, and the experiment is formally defined as the set of trials. Sometimes, your experiment can involve more than one treatment, and you want to compare and contrast the differing results from the different treatments. The experimental objects or experimental units are the objects to which the treatment is being applied. Thus, a development or maintenance project can be your experimental object, and aspects of the project's process or organization can be changed to affect the outcome. Or, the experimental objects can be programs or modules, and different methods or tools can be used on those objects. For example, if you are investigating the degree to which a design-related treatment results in reusable code components, you may consider design components as the experimental objects.

At the same time, you must identify who is applying the treatment; these people are called the experimental subjects. The characteristics of the experimental subjects must be clearly defined, so that the effects of differences among subjects can be evaluated in terms of the observed results.

When you are comparing using the treatment to not using it, you must establish a control object, which is an object not using the treatment. The control provides a baseline of information that enables you to make comparisons. In a case study, the control is the environment in which the study is being run, and it already exists at the time the study begins. In a formal experiment, the control situation must be defined explicitly and carefully, so that all the differences between the control object and the experimental object are understood.

The response variables or dependent variables are those factors that are expected to change or differ as a result of applying the treatment. In the Ada example above, quality may be considered as several attributes: the number of defects per thousand lines of code, the number of failures per thousand hours of execution time, and the number of hours of staff time required to maintain the code after deployment, for example. Each of these is considered a response variable. In contrast, state variables or independent variables are those variables that may influence the application of a treatment and thus indirectly the result of the experiment. State variables usually describe characteristics of the developers, the products, or the processes used to produce or maintain the code. It is important to define and characterize state variables so that their impact on the response variables can be investigated. Moreover, state variables can be useful in defining the scope of the experiment and in choosing the projects that will participate. For example, use of a particular design technique may be a state variable. You may decide to limit the scope of the experiment only to those projects that use Ada for a program design language, rather than investigating Ada on projects using any design technique. Or you may choose to limit the experiment to projects in a particular application domain, rather than considering all possible projects. Finally, as we saw in the first tutorial, state variables (and the control you have over them) help to distinguish case studies from formal experiments.

The number of and relationships among subjects, objects and variables must be carefully described in the experimental plan. The more subjects, objects and variables, the more complex the experimental design becomes and often the more difficult the analysis. Thus, it is very important to invest a great deal of time and care in designing your experiment, rather than rush to administer trials and collect data. This and subsequent tutorials address in more detail the types of issues that must be identified and planned in a formal experimental design. In many cases, the advice offered here should be supplemented with the advice of a statistician, especially when many subjects and objects are involved. Once the design is complete, you will know what experimental factors (that is, response and state variables) are involved, how

many subjects will be needed, from what population they will be drawn, and to what conditions or treatments each subject will be exposed. In addition, if more than one treatment is involved, the order of presentation or exposure will be laid out. Criteria for measuring and judging effects will be defined, as well as the methods for obtaining the measures.

Preparation

Preparation involves readying the subjects for application of the treatment. For example, tools may be purchased, staff may be trained, or hardware may be configured in a certain way. Instructions must be written out or recorded properly. If possible, a dry run of the experiment on a small set of people may be useful, so that you are sure that the plan is complete and the instructions understandable.

Execution

Finally, the experiment can be executed. Following the steps laid out in the plan, and measuring attributes as prescribed by the plan, you apply the treatment to the experimental subjects. You must be careful that items are measured and treatments are applied consistently, so that comparison of results is sensible.

Analysis

The analysis phase has two parts. First, you must review all the measurements taken to make sure that they are valid and useful. You organize the measurements into sets of data that will be examined as part of the hypothesis-testing process. Second, you analyze the sets of data according to the statistical principles described in later tutorials. These statistical tests, when properly administered, tell you if the null hypothesis is supported or refuted by the results of the experiment. That is, the statistical analysis gives you an answer to the original question addressed by the research.

Dissemination and Decision-Making

At the end of the analysis phase, you will have reached a conclusion about how the different characteristics you examined affected the outcome. It is important to document your conclusions in a way that will allow your colleagues to duplicate your experiment and confirm your conclusions in a similar setting. To that end, you must document all of the key aspects of the research: the objectives, the hypothesis, the experimental subjects and objects, the response and state variables, the treatments, and the resulting data. Any other relevant documentation should be included: instructions, tool or method characteristics (e.g. version, platform, vendor), training manuals and more. You must state your conclusions clearly, making sure to address any problems experienced during the running of the experiment. For example, if staff changed during project development, or if the tool was upgraded from one version to another, you must make note of that.

The experimental results may be used in three ways. First, you may use them to support decisions about how you will develop or maintain software in the future: what tools or methods you will use, and in what situations. Second, others may use your results to suggest changes to their development environment. The others are likely to replicate your experiment to confirm the results on their similar projects. Third, you and others may perform similar experiments with variations in experimental subjects or state variables. These new experiments will help you and others to understand how the results are affected by carefully controlled changes. For example, if your experiment demonstrates a positive change in quality by using Ada, others may test to see if the quality can be improved still further by using Ada in concert with a particular Ada-related tool or in a particular application domain.

PRINCIPLES OF EXPERIMENTAL DESIGN

Useful results depend on careful, rigorous and complete experimental design. Next, we examine the principles that you must consider in designing your experiment. Each principle addresses the need for simplicity and for maximizing information. Simple designs help to make the experiment practical, minimizing the use of time, money, personnel and experimental resources. An added benefit is that simple designs are easier to analyze (and thus are more economical) than complex designs. Maximizing information gives you as complete an understanding of your experimental conditions and results as possible, enabling you to generalize your results to the widest possible situations.

Involved in the experimental design are two important concepts: experimental units and experimental error. As noted above, an experimental unit is the experimental object to which a single treatment is applied. Usually, you apply the treatment more than once. At the very least, you apply it to the control group as well as at least one other group that differs from the control by a state variable. In many cases, you apply the treatment many times to many groups. In each case, you examine the results to see what the differences are in applying the treatments. However, even when you keep the conditions the same from one trial to another, the results can turn out to be slightly different. For example, you may be investigating

the time it takes for a programmer to recognize faults in a program. You have seeded a collection of similar programs with a set of known defects representing certain fault types, and you ask a programmer to find as many defects as possible. But the programmer may take more time today to find the same set of faults as he or she took yesterday. To what is this variation attributable? Experimental error describes the failure of two identically treated experimental units to yield identical results. The error can reflect a host of problems:

- errors of experimentation
- errors of observation
- errors of measurement
- the variation in experimental resources
- the combined effects of all extraneous factors that can influence the characteristics under study but which have not been singled out for attention in the investigation

Thus, in our example of timing the programmer while he or she finds faults, the differences may be due to such things as:

- the programmer's mind wandered during the experiment
- the timer measured elapsed time inexactly
- the programmer was distracted by loud noises from another room
- the programmer found the faults in a different sequence today than yesterday

The aim of a good experimental design is to control for as many variables as possible, both to minimize variability among participants and to minimize the effects of irrelevant variables (such as noise in the next room or the order of presentation of the experiment). Ideally, we would like to eliminate the effects of other variables so that only the effects of the independent variables are reflected in the values of the dependent variable. That is, we would like to eliminate experimental error. Realistically, this complete elimination is not always possible. Instead, we try to design the experiment so that the effects of irrelevant variables are distributed equally across all the experimental conditions, rather than allowing them to inflate artificially (or bias) the results of a particular condition. In fact, statisticians like whenever possible to measure the extent of the variability under "normal circumstances."

The three major principles described below, replication, randomization and local control, address this problem of variability by giving us guidance on forming experimental units so as to minimize experimental error. We consider each of these in turn.

Replication

Replication is the repetition of the basic experiment. That is, replication involves repeating an experiment under identical conditions, rather than repeating measurements on the same experimental unit. This repetition is desirable for several reasons. First, replication (with associated statistical techniques) provides an estimate of experimental error that acts as a basis for assessing the importance of observed differences in an independent variable. That is, replication can help us to know how much confidence we can place in the results of the experiment. Second, replication enables us to estimate the mean effect of any experimental factor.

It is important to ensure that replication does not introduce the confounding of effects. Two or more variables are said to be confounded if it is impossible to separate their effects when the subsequent analysis is performed. For example, suppose you want to compare the use of a new tool with your existing tool. You set up an experiment where programmer A uses the new tool in your development environment, while programmer B uses the existing tool. When you compare measures of quality in the resulting code, you cannot say how much of the difference is due to the tools because you have not accounted for the difference in the skills of the programmers. That is, the effects of the tools (one variable) and the programmers' skills (another variable) are confounded. This confounding is introduced with the replication when the repetition of the experiment does not control for other variables (like programmer skills).

Similarly, consider the comparison of two testing techniques. A test team is trained in test technique X and asked to test a set of modules. The number of defects discovered is the chosen measure of the technique's effectiveness. Then, the test team is trained in test technique Y, after which they test the same modules. A comparison of the number of defects found with X and with Y may be confounded with the similarities between techniques or a learning curve in going from X to Y. Here, the sequence of the repetition is the source of the confounding.

For this reason, the experimental design must describe in detail the number and kinds of replications of the experiments. It must identify the conditions under which each experiment is run (including the order of experimentation), and the measures to be made for each replicate.

Randomization

Replication makes possible a statistical test of the significance of the results. But it does not ensure the validity of the results. That is, we want to be sure that the experimental results clearly follow from the treatments that were applied, rather than from other variables. Some aspect of the experimental design must organize the exper-

imental trials in a way that distributes the observations independently, so that the results of the experiment are valid. Randomization is the random assignment of subjects to groups or of treatments to experimental units, so that we can assume independence (and thus validity) of results. Randomization does not guarantee independence, but it allows us to assume that the correlation on any comparison of treatments is as small as possible. In other words, by randomly assigning treatments to experimental units, you can try to keep some treatment results from being biased by sources of variation over which you have no control.

For example, sometimes the results of an experimental trial can be affected by the time, the place or unknown characteristics of the participants. These uncontrollable factors can have effects that hide or skew the results of the controllable variables. To spread and diffuse the effects of these uncontrollable or unknown factors, you can assign the order of trials randomly, assign the participants to each trial randomly, or assign the location of each trial randomly, whenever possible.

A key aspect of randomization involves the assignment of subjects to groups and treatments. If we use the same subjects in all experimental conditions, we say that we have a related within-subjects design. However, if we use different subjects in different experimental conditions, we have an unrelated between-subjects design. If there is more than one independent variable in the experiment, we can consider the use of same or different subjects separately for each of the variables. (We will describe this issue in more detail later on.)

Thus, your experimental design should include details about how you plan to randomize assignment of subjects to groups or of treatments to experimental units. In cases where complete randomization is not possible, you should document the areas where lack of randomization may affect the validity of the results. In later tutorials, we shall see examples of different designs and how they involve randomization.

Local Control

One of the key factors that distinguishes a formal experiment from a case study is the degree of control. Local control is the aspect of the experimental design that reflects how much control you have over the placement of subjects in experimental units and the organization of those units. Whereas replication and randomization ensure a valid test of significance, local control makes the design more efficient by reducing the magnitude of the experimental error. Local control is usually discussed in terms of two characteristics of the design: blocking and balancing the units.

Blocking means allocating experimental units to blocks or groups so the units within a block are relatively homogeneous. The blocks are designed so that the predictable variation among units has been confounded with the effects of the blocks. That is, the experimental design captures the anticipated variation in the blocks by grouping like varieties, so that the variation does not contribute to the experimental error. For example, suppose you are investigating the comparative effects of three design techniques on the quality of the resulting code. The experiment involves teaching the techniques to twelve developers and measuring the number of defects found per thousand lines of code to assess the code quality. It may be the case that the twelve developers graduated from three universities. It is possible that the universities trained the developers in very different ways, so that the effect of being from a particular university can affect the way in which the design technique is understood or used. To eliminate this possibility, three blocks can be defined so that the first block contains all developers from university X, the second block from university Y, and the third block from university Z. Then, the treatments are assigned at random to the developers from each block. If the first block has six developers, you would expect two to be assigned to design method A, two to B, and two to C, for instance.

Balancing is the blocking and assigning of treatments so that an equal number of subjects is assigned to each treatment, wherever possible. Balancing is desirable because it simplifies the statistical analysis, but it is not necessary. Designs can range from being completely balanced to having little or no balance.

In experiments investigating only one factor, blocking and balancing play important roles. If the design includes no blocks, then it must be completely randomized. That is, subjects must be assigned at random to each treatment. A balanced design, with equal numbers of subjects per treatment, is preferable but not necessary. If one blocking factor is used, subjects are divided into blocks and then randomly assigned to each treatment. In such a design, called a randomized block design, balancing is essential for analysis. Thus, this type of design is sometimes called a complete balanced block design. Sometimes, units are blocked with respect to two different variables (e.g., staff experience and program type) and then assigned at random to treatments so that each blocking variable combination is assigned to each treatment an equal number of times. In this case, called a Latin Square, balancing is mandatory for correct analysis.

Your experimental design should include a description of the blocks defined and the allocation of treatments to each. This part of the design will assists the analysts in understanding what statistical techniques apply to the data that results from the experiments. We will discuss the analysis in more depth in future tutorials.