C. Wohlin, M. Höst and K. Henningsson, "Empirical Research Methods in Software Engineering", In Lecture Notes in Computer Science: Empirical Methods and Studies in Software Engineering: Experiences from ESERNET, edited by A. I. Wang and R. Conradi, Springer Verlag, 2003.

In Empirical Methods and Studies in Software Engineering: Experiences from ESERNET, pp. 7-23, editors Reidar Conradi and Alf Inge Wang, Lecture Notes in Computer Science, Spinger-Verlag, Germany, LNCS 2765.

Empirical Research Methods in Software Engineering

Claes Wohlin[1], Martin Höst[2], and Kennet Henningsson[1]

[1]Dept. of Software Engineering and Computer Science Blekinge Institute of Technology
Box 520, SE-372 25 Ronneby, Sweden
Claes.Wohlin@bth.se, Kennet.Henningsson@bth.se

[2]Dept. of Communication Systems, Lund University
Box 118, SE-221 00 Lund, Sweden
Martin.Host@telecom.lth.se

Abstract. Software engineering is not only about technical solutions. It is to a large extent also concerned with organizational issues, project management and human behaviour. For a discipline like software engineering, empirical methods are crucial, since they allow for incorporating human behaviour into the research approach taken. Empirical methods are common practice in many other disciplines. This chapter provides a motivation for the use of empirical methods in software engineering research. The main motivation is that it is needed from an engineering perspective to allow for informed and well-grounded decision. The chapter continues with a brief introduction to four research methods: controlled experiments, case studies, surveys and post-mortem analyses. These methods are then put into an improvement context. The four methods are presented with the objective to introduce the reader to the methods to a level that it is possible to select the most suitable method at a specific instance. The methods have in common that they all are concerned with quantitative data. However, several of them are also suitable for qualitative data. Finally, it is concluded that the methods are not competing. On the contrary, the different research methods can preferably be used together to obtain more sources of information that hopefully lead to more informed engineering decisions in software engineering.

1 Introduction

To become a true engineering discipline software engineering has to adopt and adapt research methods from other disciplines. Engineering means, among other things, that we should be able to understand, plan, monitor, control, estimate, predict and improve the way we engineer our products. One enabler for doing this is measurement. Software measurement forms the basis, but it is not sufficient. Empirical methods such as controlled experiments, case studies, surveys and post-mortem analyses are needed to help us evaluate and validate the research results. These methods are needed so that it is possible to scientifically state whether something is better than something else. Thus, empirical methods provide one important scientific basis for software engineering. For some type of problems other methods, for example the use of mathematical models for predicting software reliability, are better suited, but in most cases the best method is applying empiricism. The main reason being that software development is human intensive, and hence it does not lend itself to analytical approaches. This means that empirical methods are essential to the researcher.

The empirical methods are however also crucial from an industrial point of view. Companies aspiring to become learning organisations have to consider the following definition of a learning organisation:

"*A learning organisation is an organisation skilled at creating, acquiring, and transferring knowledge, and at modifying its behavior to reflect new knowledge and insights*" [1]

Garvin continues with stating that learning organisations are good at five activities: systematic problem solving, experimentation, learning from past experiences, learning from others, and transferring knowledge. This includes relying on scientific methods rather than guesswork. From the perspective of this chapter, the key issue is the application of a scientific method and the use of empirical methods as a vehicle for systematic improvement when engineering software. The quote from Garvin is in-line with the concepts of the Quality Improvement Paradigm and the Experience Factory [2] that are often used in a software engineering context.

In summary, the above means that software engineering researchers and learning organisations both have a need to embrace empirical methods. The main objective of this chapter is to provide an introduction to four empirical research methods and to put them into an engineering context.

The remainder of this chapter is outlined as follows. Four empirical methods are briefly introduced in Section 2 to provide the reader with a reference framework to better understand the differences and similarities between the methods later. In Section 3, the four empirical methods are put into an improvement context before presenting the methods in some more details in Sections 4-7. The chapter is concluded with a short summary in Section 8 and references in Section 9.

2 Overview of Empirical Methods

There are two main types of research paradigms having different approaches to empirical studies. **Qualitative research** is concerned with studying objects in their natural setting. A qualitative researcher attempts to interpret a phenomenon based on explanations that people bring to them [3]. Qualitative research begins with accepting that there is a range of different ways of interpretation. It is concerned with discovering causes noticed by the subjects in the study, and understanding their view of the problem at hand. The subject is the person, which is taking part in a study in order to evaluate an object.

Quantitative research is mainly concerned with quantifying a relationship or to compare two or more groups [4]. The aim is to identify a cause-effect relationship. The quantitative research is often conducted through setting up controlled experiments or collecting data through case studies. Quantitative investigations are appropriate when testing the effect of some manipulation or activity. An advantage is that quantitative data promotes comparisons and statistical analysis. The use of quantitative research methods is dependent on the application of measurement, which is further discussed in [5].

It is possible for qualitative and quantitative research to investigate the same topics but each of them will address a different type of question. For example, a quantitative investigation could be launched to investigate how much a new inspection method decreases the number of faults found in test. To answer questions about the sources of variations between different inspection groups, we need a qualitative investigation.

As mentioned earlier quantitative strategies such as controlled experiments are appropriate when testing the effects of a treatment, while a qualitative study of beliefs and understandings are appropriate to find out *why* the results from a quantitative investigation are as they are. The two approaches should be regarded as complementary rather than competitive.

In general, any empirical study can be mapped to the following main research steps: Definition, Planning, Operation, Analysis & interpretation, Conclusions and Presentation & packaging. The work within the steps differs considerably depending on the type of empirical study. However, instead of trying to present four different research methods according to this general process, we have chosen to highlight the main aspects of interest for the different types of studies.

Depending on the purpose of the evaluation, whether it is techniques, methods or tools, and depending on the conditions for the empirical investigation, there are four major different types of investigations (strategies) that are addressed here:

- **Experiment.** Experiments are sometimes referred to as research-in-the-small [6], since they are concerned with a limited scope and most often are run in a laboratory setting. They are often highly controlled and hence also occasionally referred to as controlled experiment, which is used hereafter. When experimenting, subjects are assigned to different treatments at random. The objective is to manipulate one or more variables and control all other variables at fixed levels. The effect of the manipulation is measured, and based on this a statistical analysis can be performed. In some cases it may be impossible to use true experimentation, we may have to use quasi-experiments. The latter term is often used when it is impossible to perform random assignment of the subjects to the different treatments. An example of a controlled experiment in software engineering is to compare two different methods for inspections. For this type of studies, methods for statistical inference are applied with the purpose of showing with statistical significance that one method is better than the other [7, 8, 9].
- **Case study.** Case study research is sometimes referred to as research-in-the-typical [6]. It is described in this way due to that normally a case study is conducted studying a real project and hence the situation is "typical". Case studies are used for monitoring projects, activities or assignments. Data is collected for a specific purpose throughout the study. Based on the data collection, statistical analyses can be carried out. The case study is normally aimed at tracking a specific attribute or establishing relationships between different attributes. The level of control is lower in a case study than in an experiment. A case study is an observational study while the experiment is a controlled study [10]. A case study may, for example, be aimed at building a model to predict the number of faults in testing. Multivariate statistical analysis is often applied in this type of studies. The analysis methods include linear regression and principal component analysis [11]. Case study research is further discussed in [9, 12, 13,14].

The following two methods are both concerned with research-in-the-past, although they have different approaches to studying the past.

- **Survey**. The survey is by [6] referred to as research-in-the-large (and past), since it is possible to send a questionnaire to or interview a large number people covering whatever target population we have. Thus, a survey is often an investigation performed in retrospect, when e.g. a tool or technique, has been in use for a while [13]. The primary means of gathering qualitative or quantitative data are interviews or questionnaires. These are done through taking a sample that is representative from the population to be studied. The results from the survey are then analyzed to derive descriptive and explanatory conclusions. They are then generalized to the population from which the sample was taken. Surveys are discussed further in [9,15].
- **Post-mortem analysis**. This type of analysis is also conducted on the past as indicated by the name. However, it should be interpreted a little broader than as post-mortem. For example, a project does not have to be finished to launch a post-mortem analysis. It would be possible to study any part of a project retrospectively using this type of analysis. Thus, this type of analysis may, in the descriptive way used by [6], be described as being research-in-the-past-and-typical. It can hence be viewed as related to both the survey and the case study. The post-mortem may be conducted by looking at project documentation (archival analysis [9]) or by

interviewing people, individually or as a group, who have participated in the object that is being analysed in the post-mortem analysis.

An experiment is a formal, rigorous and controlled investigation. In an experiment the key factors are identified and manipulated. The separation between case studies and experiment can be represented by the notion of a state variable [13]. In an experiment, the state variable can assume different values and the objective is normally to distinguish between two situations, for example, a control situation and the situation under investigation. Examples of a state variable could be, for example, the inspection method or experience of the software developers. In a case study, the state variable only assumes one value, governed by the actual project under study.

Case study research is a technique where key factors that may have any affect on the outcome are identified and then the activity is documented [12, 14]. Case study research is an observational method, i.e. it is done by observation of an on-going project or activity.

Surveys are very common within social sciences where, for example, attitudes are polled to determine how a population will vote in the next election. A survey provides no control of the execution or the measurement, though it is possible to compare it with similar ones, but it is not possible to manipulate variables as in the other investigation methods [15].

Finally, a post-mortem analysis may be viewed as inheriting properties from both surveys and case studies. A post-mortem may contain survey elements, but it is normally concerned with a case. The latter could either be a full software project or a specific targeted activity.

For all four methods, it is important to consider the population of interest. It is from the population that a sample should be found. The sample should preferably be chosen randomly from the population. The sample consists of a number of subjects, for example in many cases individuals participating in a study. The actual population may vary from an ambition to have a general population, as is normally the objective in experiments where we would like to generalize the results, to a more narrow view, which may be the case in post-mortem analyses and case studies.

Some of the research strategies could be classified both as qualitative and quantitative, depending on the design of the investigation, as shown in Table 1. The classification of a survey depends on the design of the questionnaires, i.e. which data is collected and if it is possible to apply any statistical methods. Also, this is true for case studies but the difference is that a survey is done in retrospect while a case study is done while a project is executed. A survey could also be launched before the execution of a project. In the latter case, the survey is based on previous experiences and hence conducted in retrospect to these experiences although the objective is to get some ideas of the outcome of the forthcoming project. A post-mortem is normally conducted close to finish an activity or project. It is important to conduct it close in time to the actual finish so that people are still available and the experiences fresh.

Experiments are purely quantitative since they have a focus on measuring different variables, change them and measure them again. During these investigations quantitative data is collected and then statistical methods are applied. Sections 4-7 give introductions to each empirical strategy, but before this the empirical methods are put into an improvement context in the following section. The introduction to controlled experiments is longer than for the other empirical methods. The main reason being that the procedure for running controlled experiments is more formal, i.e. it is sometimes referred to as a fixed design [9]. The other methods are more flexible and it is hence not possible to describe the actual research process in the same depth. Table 1 indicates this, where the qualitative and quantitative nature of the methods are indicated. Methods with a less fixed design are sometimes referred to as flexible design [9], which also indicates that the design may change during the execution of the study due to events happening during the study.

Table 1. Qualitative vs. quantitative in empirical strategies.

Strategy	**Qualitative / Quantitative**
Experiment	Quantitative
Case study	Both
Survey	Both
Post-mortem	Both

3 Empirical Methods in an Improvement Context

Systematic improvement includes using a generic improvement cycle such as the Quality Improvement Paradigm (QIP) [2]. This improvement cycle is generic in the sense that it can both be viewed as a recommended way to work with improvement of software development, but it may also be used as a framework for conducting empirical studies. For simplicity, it is here primarily viewed as a way of improving software development, and complemented with a simple three steps approach on how the empirical methods can be used as a vehicle for systematic engineering-based improvement.

The QIP consists of six steps that are repeated iteratively:

1. **Characterize**. The objective is to understand the current situation and establish a baseline.
2. **Set goals**. Quantifiable goals are set and given in terms of improvement.
3. **Choose process/method/technique**. Based on the characterization and the goals, the part to improve is identified and a suitable improvement candidate is identified.
4. **Execute**. The study or project is performed and the results are collected for evaluation purposes.
5. **Analyze**. The outcome is studied and future possible improvements are identified.
6. **Package**. The experiences are packaged so that it can form the basis for further improvements.

It is in most cases impossible to start improving directly. The first step is normally to understand the current situation and then improvement opportunities are identified and they need to be evaluated before being introduced into an industrial process as an improvement. Thus, systematic improvement is based on the following steps:

- Understand,
- Evaluate, and
- Improve.

As a scenario, it is possible to imagine that one or both of the two methods looking at the past are used for understanding and baselining, i.e. a survey or a post-mortem analysis may be conducted to get a picture of the current situation. The objectives of a survey and a post-mortem analysis are slightly different as discussed in Section 2. The evaluation step may either be executed using a controlled experiment or a case study. It is most likely a controlled experiment if the identified improvement candidate is evaluated in a laboratory setting and compared with another method, preferably the existing method or a method that may be used for benchmarking. It may be a case study if it is judged that the improvement candidate can be introduced in a pilot project directly. This pilot project ought to be studied and a suitable method is to use a case study. In the actual improvement in an industrial setting (normally initially in one project), it is probably best suited to use a case study approach, which then may be compared with the situation found when creating the understanding. Finally, if the evaluation comes out positive, the improvement is incorporated in the standard software development process.

The above means that the four methods presented here should be viewed as complementary and not competing. They have all their benefits and drawbacks. The scenario above should be viewed as one possible way of using the methods as complementary in improving the way software is engineered.

Next, the four methods are presented in more detail to provide an introduction and understanding of them. The objective is to provide sufficient information so that a researcher intending to conduct an empirical study in software engineering can select an appropriate method given the situation at hand.

4 Controlled experiments

4.1 Introduction

In an experiment the researcher has control over the study and how the participants carry out the tasks that they are assigned to. This can be compared to a typical case study, see below, where the researcher is more of an observer. The advantage of the experiment is, of course, that the study can be planned and designed to ensure high validity, although the drawback is that the scope of the study often gets smaller. For example, it would be possible to view a complete software development project as a case study, but a typical experiment does not include all activities of such a project.

Experiments are often conducted to compare a number of different techniques, methods, working procedures, etc. For example, an experiment could be carried out with the objective of comparing two different reading techniques for inspections. In this example two groups of people could independently perform a task with one reading technique each. That is, if there are two reading techniques, R1 and R2, and two groups, G1 and G2, then people in group G1 could use technique R1 and people in group G2 could use technique R2. This small example is used in the following subsections to illustrate some of the concepts for controlled experiments.

4.2 Design

Before the experiment can be carried out it must be planned in detail. This plan is often referred to as the design of the experiment.

In an experiment we wish to draw conclusions that are valid for a large population. For example, we wish to investigate whether reading technique R1 is more effective than reading technique R2 in general for any developer, project, organisation, etc. However, it is, of course, impossible to involve every developer in the study. Therefore, a sample of the entire population is used in the experiment. Ideally, it would be possible to randomly choose a sample from the population to include in the study, but this is for obvious reasons mostly impossible. Often, we end up trying to determine to which population we can generalise the results from a certain set of participants.

The main reason for the above is that relation between sample and population is intricate and difficult to handle. In the software engineering domain, it is mostly desirable to sample from all software developers, or a subset of them, for example all software designers using a specific programming language. This is for practical reasons impossible. Thus, in the best case it is possible to choose from software developers in the vicinity of the researcher. This means that it is not a true sample from the population, although it may be fairly good. In many cases, it is impossible to have professional developers and students are used, and in particular we have to settle for students in a specific course. The latter is referred to as convenience sampling [9]. This situation leads to that we in most cases must go from subjects to population when the preferred situation is to go from population to subjects through random sampling. This should not necessarily be seen as a failure. It may be a complementary approach. However, it is important to be aware of the difference and also to consider how this affects the statistical analysis, since most statistical methods have developed based on the assumption of a random sample from the population of interest. The challenge of representative samples is also discussed in Chapter 3.

Another important principle of experiments is randomisation. With this we mean that when it is decided which treatment every participant should be subject to, this is done by random. For example, if 20 persons participate in the study where the two reading techniques R1 and R2 are compared, it is decided by random which 10 persons that should use R1 and which 10 persons that should use R2.

In experiments a number of variables are often defined. Two important types of variables are:

- Independent variables: These variables describe the treatments in the experiment. In the above example, the choice of reading technique is an independent variable that can take one of the two values R1 or R2.
- Dependent variables: These variables are studied to investigate whether they are influenced by the independent variables. For example, the number of defects can be a dependent variable that we believe is dependent on whether R1 or R2 is used. The objective of the experiment is to determine if and how much the dependent variables are affected by the independent variables.

The independent and dependent variables are formulated to cover one or several hypotheses that we have with respect to the experiment. For example, we may hypothesize that the number of defects is dependent on the two reading techniques in the example. Hypothesis testing is discussed further in relation to the analysis.

The independent and dependent variables are illustrated in Figure 1 together with the confounding factors. Confounding factors are variables that may affect the dependent variables without the knowledge of the researcher. It is hence crucial to try to identify the factors that otherwise may affect the outcome in an undesirable way. These factors are closely related to the threats against the validity of the empirical study. Thus, it is important to consider confounding factors and the threats to the study throughout the performance of any empirical study. The threats to empirical studies are discussed in Section 4.4. One objective of the design is to minimise the effect of these factors.

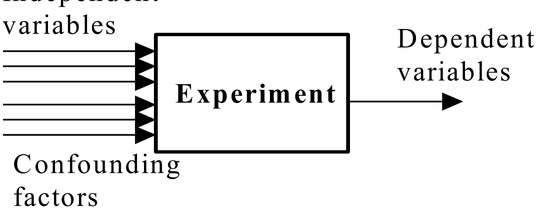

Figure 1. Variables in an experiment.

Often one of several available standard designs is used. Some examples of standard designs are:

- *Standard design 1:* One independent variable with two values: For example, two techniques should be compared and each participant uses one of the techniques.

- *Standard design 2:* One independent variable with two values, paired design: The difference between this design and standard design 1 is that each person in this design is subject to both treatments. The order in which each participant should apply the treatments is decided by random. For example, if the two reading techniques should be evaluated, half of the participants first uses R1 and then R2, and the other half first uses R2 and then R1. The reason for using the treatments in different order is that effects of the order should be ruled out.
- *Standard design 3:* One independent variable with more than two values: The difference between this design and standard design 1 is that more than two treatments are compared. For example, three reading techniques may be compared.
- *Standard design 4:* More than one independent variable: With this design more than one aspect can be evaluated in an experiment. For example, both the choice of reading technique and requirements notation may be compared in one experiment.

The designs that are presented here are a summary of some of the most commonly used designs. There are alternatives and more complicated designs. For example, sometimes experiments are carried out as a combination of a pre-study and a main experiment.

4.3 Operation

In the operation of an experiment a number of parts can be included. These include both parts that have to be done when starting the experiment and when actually running the experiment. Three key parts are:

- Commit participants: It is important that every participant is committed to the tasks. There are a number of factors to consider, for example, if the experiment concerns sensitive material, it will be difficult to get committed people.
- Prepare instrumentation: All the material that should be used during the experiment must be prepared. This may include written instructions to the participants, forms that should be used by the participants during the tests, etc. The instrumentation should be developed according to the design of the experiment. In most cases different participants should be given different sets of instructions and forms. In many cases paper-based forms are used during an experiment. It is, however, possible to collect data in a number of other ways, e.g., web-based forms, interviews, etc.
- Execution: The actual execution denotes the part of the experiment where the participants, subject to their treatment, carry out the task that they are assigned to. For example, it may mean that some participants solve a development assignment with one development tool and the other participants solve the same assignment with another tool. During this task the participants use the prepared instrumentation to receive instructions and to record data that can be used later in analysis.

4.4 Analysis and interpretation

Before actually doing any analysis, it is important to validate that the data is correct, and that the forms etc. have been filled out correctly. This activity may also be sorted under execution of the experiment, and hence be carried out before the actual analysis.

The first part in the actual analysis is normally to apply descriptive statistics. This includes plotting the data in some way to obtain an overview of the data. Part of this analysis is done to identify and handle outliers. An outlier denotes a value that is atypical and unexpected in the data set. They may, for example, be identified through box-plots [16] or scatter-plots. Every outlier must be investigated and handled separately. It may be that the value simply is wrong. Then it may be corrected or discarded. It may also, of course, be the case that the value is correct. In that case it can be included in the analysis or, if the reason for the atypical value can be identified, it may be handled separately.

When we have made sure that the data is correct and received a good understanding of the data from the descriptive statistics then the analysis related to testing one or several hypotheses can start. In most cases the objective here is to decide whether there is an effect of the value of the independent variable(s) on the value of the dependent variable(s). This is in most cases analysed through hypothesis testing. To understand hypothesis testing some important definitions must be understood:

- The null hypothesis H_0 denotes that there is no effect of the independent variable on the dependent variable. The objective of the hypothesis test is to reject this hypothesis with a known significance.
- P(type-I error) = P(reject H_0 | H_0 is true). This probability may also be called the significance of a hypothesis test.
- P(type-II error) = P(not reject H_0 | H_0 is false)
- Power = 1 - P(type-II error) = P(reject H_0 | H_0 is false)

When the test is carried out, a maximum P(type-I error) is first decided. Then a test is used in order to decide whether it is possible to reject the null hypothesis or not. When choosing a test, it must be decided whether to use parametric or non-parametric tests. Generally, there are harder requirements on the data for parametric test. They are, for example, based on that the data is normally distributed. However, parametric tests generally have higher power than non-parametric tests, i.e. less data is needed to obtain significant results when using parametric tests. The difference is not large. It is, of course, impossible to provide any exact figure, but it is in most cases in the order of 10%. For every design there is a number of tests that may be used. Some examples of tests are given in Table 2.

Table 2. Examples of tests.

Standard design (see above)	Parametric tests	Non-parametric tests
Standard design 1	t-test	Mann-Whitney
Standard design 2	Paired t-test	Wilcoxon, Sign-test
Standard design 3	ANOVA	Kruskal-Wallis
Standard design 4	ANOVA	

The tests in Table 2 are all described in a number of basic statistical references. More information on parametric tests can be found in [7], and information on the non-parametric tests can be found in, for example, [8] and [17].

Before the results are presented it is important to assess how valid the results are. Basically there are four categories of validity concerns, which are discussed in a software engineering context in [18]:

- Internal: The internal validity is concerned with factors that may affect the dependent variables without the researcher's knowledge. An example of an issue is whether the history of the participants affects the result of an experiment. For example, the result may not be the same if the experiment is carried out directly after a complicated fault in the code has caused the participant a lot of problem compared to a more normal situation. A good example of how confounding factors may threaten the internal validity in a study is presented in [19].
- External: The external validity is related to the ability to generalise the results of the experiments. Examples of issues are whether the problem that the participants

have been working with is representative and whether the participants are representative of the target population.
- Conclusion: The conclusion validity is concerned with the possibility to draw correct conclusions regarding the relationship between treatments and the outcome of an experiment. Examples of issues to consider are whether the statistical power of the tests is too low, or if the reliability of the measurements is high enough.
- Construct: The construct validity is related to the relationship between the concepts and theories behind the experiment and what is measured and affected. Examples of issues are whether the concepts are defined clearly enough before measurements are defined, and interaction of different treatments when persons are involved in more than one study.

Obviously, it is important to have these validity concerns in mind already when the designing the experiment and in particular when using a specific design type. In the analysis phase it is too late to change the experiment in order to obtain better validity. The different validity threats should also be considered for the other type of empirical studies discussed in the following sections.

When the analysis is completed the next step is to draw conclusions and take actions based on the conclusions.

More in-depth descriptions of controlled experiments in a software engineering context can be found in [18] and [20].

5 Case Study

5.1 Introduction

A case study is conducted to investigate a single entity or phenomenon within a specific time space. The researcher collects detailed information on, for example, one single project during a sustained period of time. During the performance of a case study, a variety of different data collection procedures may be applied [4].

If we would like to compare two methods, it may be necessary to organize the study as a case study or an experiment. The choice depends on the scale of the evaluation. An example can be to use a pilot project to evaluate the effects of a change compared to some baseline [6].

Case studies are very suitable for industrial evaluation of software engineering methods and tools because they can avoid scale-up problems. The difference between case studies and experiments is that experiments sample over the variables that are being manipulated, while case studies sample from the variables representing the typical situation. An advantage of case studies is that they are easier to plan but the disadvantages are that the results are difficult to generalize and harder to interpret, i.e. it is possible to show the effects in a typical situation, but it cannot be generalized to every situation [14].

If the effect of a process change is very widespread, a case study is more suitable. The effect of the change can only be assessed at a high level of abstraction because the process change includes smaller and more detailed changes throughout the development process [6]. Also, the effects of the change cannot be identified immediately. For example, if we would like to know if a new design tool increases the reliability, it may be necessary to wait until after delivery of the developed product to assess the effects on operational failures.

Case study research is a standard method used for empirical studies in various sciences such as sociology, medicine and psychology. Within software engineering, case studies should not only be used to evaluate how or why certain phenomena occur, but also to evaluate the differences between, for example, two design methods. This means in other words, to determine "which is best" of the two methods [14]. An example of a case study in software engineering is an investigation if the use of perspective-based reading increases the quality of requirements specifications. A study like this cannot verify that perspective-based reading reduces the number of faults that reaches test, since this requires a reference group that does not use perspective-based techniques.

5.2 Case study arrangements

A case study can be applied as a comparative research strategy, comparing the results of using one method or some form of manipulation, to the results of using another approach. To avoid bias and to ensure internal validity, it is necessary to create a solid base for assessing the results of the case study. There are three ways to arrange the study to facilitate this [6].

A comparison of the results of using the new method against a company baseline is one solution. The company should gather data from standard projects and calculate characteristics like average productivity and defect rate. Then it is possible to compare the results from the case study with the figures from the baseline.

A sister project can be chosen as a baseline. The project under study uses the new method and the sister-project the current one. Both projects should have the same characteristics, i.e. the projects must be comparable.

If the method applies to individual product components, it could be applied at random to some components and not to others. This is very similar to an experiment, but since the projects are not drawn at random from the population of all projects, it is not an experiment.

5.3 Confounding factors and other aspects

When performing case studies it is necessary to minimize the effects of confounding factors. A confounding factor is, as it is described in Section 4, a factor that makes it impossible to distinguish the effects from two factors from each other. This is important since we do not have the same control over a case study as in an experiment. For example, it may be difficult to tell if a better result depends on the tool or the experience of the user of the tool. Confounding effects could involve problems with learning how to use a tool or method when trying to assess its benefits, or using very enthusiastic or sceptical staff.

There are both pros and cons with case studies. Case studies are valuable because they incorporate qualities that an experiment cannot visualize, for example, scale, complexity, unpredictability, and dynamism. Some potential problems with case studies are as follow.

A small or simplified case study is seldom a good instrument for discovering software engineering principles and techniques. Increases in scale lead to changes in the type of problems that become most indicative. In other words, the problem may be different in a small case study and in a large case study, although the objective is to study the same issues. For example, in a small case study the main problem may be the actual technique being studied, and in a large case study the major problem may be the amount of people involved and hence also the communication between people.

Researchers are not completely in control of a case study situation. This is good, from one perspective, because unpredictable changes frequently tell them much about the problems being studied. The problem is that we cannot be sure about the effects due to confounding factors.

More information on case study research can be found in, for example, [12] and [14].

6 Survey

Surveys are conducted when the use of a technique or tool already has taken place [13] or before it is introduced. It could be seen as a snapshot of the situation to capture the current status. Surveys could, for example, be used for opinion polls and market research.

When performing survey research the interest may be, for example, in studying how a new development process has improved the developer's attitudes towards quality assurance. Then a sample of developers is selected from all the developers at the company. A questionnaire is constructed to obtain information needed for the research. The questionnaires are answered by the sample of developers. The information collected is then arranged into a form that can be handled in a quantitative or qualitative manner.

6.1 Survey characteristics

Sample surveys are almost never conducted to create an understanding of the particular sample. Instead, the purpose is to understand the population, from which the sample was drawn [15]. For example, by interviewing 25 developers on what they think about a new process, the opinion of the larger population of 100 developers in the company can be predicted. Surveys aim at the development of generalized suggestions.

Surveys have the ability to provide a large number of variables to evaluate, but it is necessary to aim at obtaining the largest amount of understanding from the fewest number of variables since this reduction also eases the analysis work.

It is not necessary to guess which the most relevant variables in the initial design of the study are. The survey format allows the collection of many variables, which in many cases may be quantified and processed by computers. This makes it is possible to construct a variety of explanatory models and then select the one that best fits the purposes of the investigation.

6.2 Survey purposes

The general objective for conducting a survey is either of the following [15]:

- Descriptive.
- Explanatory.
- Explorative.

Descriptive surveys can be conducted to enable assertions about some population. This could be determining the distribution of certain characteristics or attributes. The concern is not about why the observed distribution exists, rather what it is.

Explanatory surveys aim at making explanatory claims about the population. For example, when studying how developers use a certain inspection technique, we might want to explain why some developers prefer one technique while others prefer another. By examining the relationships between different candidate techniques and several explanatory variables, we may try to explain why developers choose one of the techniques.

Finally, explorative surveys are used as a pre-study to a more thorough investigation to assure that important issues are not foreseen. Creating a loosely structured questionnaire and letting a sample from the population answer it could do this. The information is gathered and analyzed, and the results are used to improve the full investigation. In other words, the explorative survey does not answer the basic research question, but it may provide new possibilities that could be analyzed and should therefore be followed up in the more focused or thorough survey.

6.3 Data collection

The two most common means for data collection are questionnaires and interviews [15]. Questionnaires could both be provided in paper form or in some electronic form, for example, e-mail or WWW pages. The basic method for data collection through questionnaires is to send out the questionnaire together with instructions on how to fill it in. The responding person answers the questionnaire and then returns it to the researcher.

Letting interviewers handle the questionnaires (by telephone or face-to-face) instead of the respondents themselves, offers a number of advantages:

- Interview surveys typically achieve higher response rates than, for example, mail surveys.
- An interviewer generally decreases the number of "do not know" and "no answer", because he/she can answer questions about the questionnaire.
- It is possible for the interviewer to observe and ask questions.

The disadvantage is the cost and time, which depend on the size of the sample, and they are also related to the intentions of the investigation.

7 Post-mortem analysis

Post-mortem analysis is a research method studying the past, but also focusing on the typical situation that has occurred. Thus, a post-mortem analysis is similar to the case study in terms of scope and to the survey in that it looks at the past. The basic idea behind post-mortem analysis is to capture the knowledge and experience from a specific case or activity after it has been finished. [21] identifies two types of post-mortem analysis: a general post-mortem analysis capturing all available information from an activity or a focused post-mortem analysis for a specific activity, for example, cost estimation.

According to [21], post-mortem analysis has mainly been targeted at large software projects to learn form their success or recovery from a failure. An example of such a process is proposed by [22]. The steps are:

1. Project survey.
   The objective is to use a survey to collect information about the project from the participants. The use of a survey ensures that confidentiality can be guaranteed.
2. Collect objective information.
   In the second step, objective information that reveals the health of the project is collected. This includes defect data, person-hours spent and so forth.
3. Debriefing meeting.
   A meeting is held to capture issues that where not covered by the survey. In addition, it provides the project participants with an opportunity to express their view.
4. Project history day.
   The history day is conducted with a selected subset of the people involved to review project events and project data.
5. Publish the results.
   Finally, a report is published. The report is focused on the lessons-learned and to use that to guide organisational improvement.

To support small- and medium-sized companies, [21] discusses a lightweight approach to post-mortem analysis, which focuses on a few vital activities and highlights that:

- Post-mortem analyses should be open for participation for all team members and other stakeholders,
- Goals may be used to focus the discussions, but it is not necessary,
- The post-mortem process consists of three main phases: preparation, data collection and analysis. These phases are further discussed in [21].

Post-mortem analyses are a flexible type of analysis method. The actual object to be studied (a whole project or specific activity) and the type of questions posed are very much dependent on the actual situation and the objectives of the analysis.

The referenced articles or the book by Whitten [23] provide more information on post-mortem analysis/review.

Finally, it should be noted that empirical methods also provide positive side effects such knowledge sharing, which is an added-value from conducting an empirical study. This is true for all types of empirical studies. In an experiment, the subjects learn from comparing competing methods or techniques. This is particular true if the subjects are debriefed afterwards in terms of obtaining information about the objective and the outcome of the experiment. In case studies and post-mortem analyses the persons participating obtain a new perspective of their work and they often reflect on their way of working through the participation in the empirical study. Finally, in the survey the learning comes from comparing answers given with the general outcome of the survey. This allows individuals to put their own answers in a more general context.

8 Summary

This chapter has provided a brief overview of four empirical research methods with a primary focus on methods that contain some quantitative part. The four methods are: controlled experiments, case studies, surveys and post-mortem analyses. The main objective has been to introduce them so that people intending to conduct empirical studies can make an appropriate selection of an empirical research method in a software engineering context.

Moreover, the presented methods must be seen as complementary in that they can be applied at different stages in the research process. This means that they can, together in a suitable combination, support each other and hence provide a good basis for sustainable improvement in software development.

9 References

[1] D. A. Garvin, “Building a Learning Organization”, in *Harward Business Review on Knowledge Management*, pp. 47-80, Harward Business School Press, Boston, USA, 1998.

[2] V. R. Basili, G. Caldiera, and H. D. Rombach, ”Experience Factory” in *Encyclopaedia of Software Engineering*, editor John J. Marciniak, John Wiley & Sons, Inc., Hoboken, N.J., USA, 2002.

[4] J. W. Creswell, *Research Design, Qualitative and Quantitative Approaches*, Sage Publications, 1994.

[3] N. K. Denzin and Y. S. Lincoln, *Handbook of Qualitative Research*, Sage Publications, London, UK, 1994.

[5] N. Fenton, and S. L. Pfleeger, *Software Metrics: A Rigorous & Practical Approach*, 2nd edition, International Thomson Computer Press, 1996.

[6] B. Kitchenham, L. Pickard and S. L. Pfleeger, “Case Studies for Method and Tool Evaluation”, *IEEE Software*, pp. 52-62, July 1995.

[7] D. C. Montgomery, *Design and Analysis of Experiments*, 4th edition, John Wiley & Sons, New York, USA, 1997.

[8] S. Siegel, J. Castellan, *Nonparametric Statistics for the Behavioral Sciences*, 2nd edition, McGraw-Hill International, New York, USA, 1988.

[9] C. Robson, *Real World Research*, 2nd edition, Blackwell, 2002.

[10] M. V. Zelkowitz and D. R. Wallace, "Experimental Models for Validating Technology", *IEEE Computer*, 31(5), pp. 23-31, 1998.

[11] B. F. J. Manly, *Multivariate Statistical Methods - A Primer,* Second Edition, Chapman & Hall, London, 1994.

[12] R. E. Stake, *The Art of Case Study Research*, SAGE Publications, 1995.

[13] S. Pfleeger, "Experimental Design and Analysis in Software Engineering Part 1-5", *ACM Sigsoft, Software Engineering Notes*, Vol. 19, No. 4, pp. 16-20; Vol. 20, No. 1, pp. 22-26; Vol. 20, No. 2, pp. 14-16; Vol. 20, No. 3, pp. 13-15; and Vol. 20, No. 4, pp. 14-17, 1994-1995.

[14] R. K. Yin, *Case Study Research Design and Methods*, Sage Publications, Beverly Hills, California, 1994.

[15] E. Babbie, *Survey Research Methods*, Wadsworth, ISBN 0-524-12672-3, 1990.

[16] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.

[17] C. Robson, *Design and Statistics in Psychology*, 3rd edition, Penguin Books, London, England, 1994.

[18] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, *Experimentation in Software Engineering – An Introduction*, Kluwer Academic Publishers, Boston, MA, USA, 1999.

[19] C. M. Judd, E. R. Smith and L .H. Kidder, *Research Methods in Social Relations*, Harcourt Brace Jovanovich College Publishers, Forth Worth, Texas, USA6th Edition, 1991

[20] N. Juristo and A. Moreno, *Basics of Software Engineering Experimentation*, Kluwer Academic Publishers, Boston, Massachusetts, USA, 2001.

[21] A. Birk, T. Dingsøyr and T. Stålhane, "Postmortem: Never Leave a Project without It", *IEEE Software*, pp. 43-45, May/June 2002.

[22] B. Collier, T. DeMarco and P. Fearey, "A Defined Process for Project Postmortem Review", *IEEE Software*, pp. 65-72, July 1996.

[23] N. Whitten, *Managing Software Development Projects - Formula for Success*, John Wiley and Sons, Inc., New York, USA, 1995.