Towards a Decision-making Structure for Selecting a Research Design in Empirical Software Engineering

Claes Wohlin, Blekinge Institute of Technology, Sweden,
email: claes.wohlin@bth.se

Aybüke Aurum, Sydney, Australia, email: aybuke.aurum@gmail.com

Abstract

Several factors make empirical research in software engineering particularly challenging as it requires studying not only technology but its stakeholders' activities while drawing concepts and theories from social science. Researchers, in general, agree that selecting a research design in empirical software engineering research is challenging, because the implications of using individual research methods are not well recorded.

The main objective of this article is to make researchers aware and support them in their research design, by providing a foundation of knowledge about empirical software engineering research decisions, in order to ensure that researchers make well-founded and informed decisions about their research designs.

This article provides a decision-making structure containing of a number of decision points, each one of them representing a specific aspect on empirical software engineering research. The article provides an introduction to each decision point and its constituents, as well as to the relationships between the different parts in the decision-making structure. The intention is that structure should act as a starting point for the research design before going into the details of the research design chosen. The article provides an in-depth discussion of decision points in relation to the research design when conducting empirical research.

Keywords: Research Methods, Empirical Software Engineering Research, Selecting Research Method, Research Design

1. Introduction

There is a growing interest in the evolution of empirical software engineering research with respect methodological assumptions (Easterbrook et al., 2008; Sjøberg. et al., 2007; Perry et al., 2000). Considerable attention has been paid to the issues of research diversity and methodological pluralism in the last fifteen years. Specifically qualitative research methods have become more common beside quantitative research methods, and the use of a mix of methods is increasing (Dybå et al., 2011, McLeod et al., 2011; Seaman 1999; Runeson and Höst, 2009; Carver et al., 2004; Kontio et al., 2004).

Empirical research takes many forms and it may be perceived as a challenge to know which research approaches and research methods to apply in different situations. The plethora of methodologies such as case studies, action research or design science research makes it very difficult to choose an appropriate research method in a given situation. As a researcher, the different options work as a toolbox, and it is far from trivial to choose the right tool in a given research study. Despite the challenges it is crucial that researchers take well founded and informed decisions regarding which research methodologies, methods and approaches to use in a given situation. No matter what its form is, the essence of empirical research aims to acquire knowledge by empirical methods (Sjøberg et al., 2007; Perry et al., 2000). However, empirical research is much more than mere speculation or assumptions about software development activities or evaluating the use of the technology by stakeholders, e.g. individual developers, projects or software organizations. Researchers must be able to use appropriate research methods to investigate their research questions and for collecting and analysing the data. Furthermore, researchers must become more aware of the options and potential consequences of their decisions.

There are several factors that make empirical research in software engineering particularly challenging. Firstly, studying human activities in software development is always challenging as it requires studying not only the technology in use, but also social and cognitive processes that surrounds the stakeholders (Bertelsen, 1997; Shaw 2002; Easterbrook et al., 2008). Thus, the research in empirical software engineering borrows concepts and theories from social sciences research, as it involves study of human activities. Given the variety of uses of the concepts and terminology of social science research, researchers have difficulty to explaining their research design (Grix, 2002). Secondly, selecting a research method in empirical software engineering research is problematic, because the implications of using individual methods are not well recorded. Many researchers select inappropriate methods because they do not fully understand the underlying assumptions for the methods they select or they possess limited knowledge about alternatives (Sjøberg et al., 2007; Easterbrook et al., 2008). Hence, researchers are reluctant about providing an explicit picture about their research paradigm, the formulation of their research methods and the standards for judging quality of results (Shaw, 2002).

Much recent self-reflection in software engineering research has involved a discussion of empirical software engineering research and what constitutes a scientific discipline (Seaman, 1999; Sjøberg et al., 2007; Shaw, 2003; Runeson and Höst 2009; McLeod et al., 2011; Dybå et al., 2011). Researchers agree that there is a need to increase the shared understanding about conducting empirical research in software engineering. In this article we argue that it is critical to have a clear and transparent knowledge of the research design and the underlying research process to (i) understand the interrelationship of the main components of research; (ii) avoid confusion when discussing the logic behind the research design or assumptions that have been made (iii) be able to present the research results with confidence and being able to persuade the reader of its conclusions, (iv) be able to comply research standards (v) be able to understand and put other researchers' work in context.

The main objective of this article is to make researchers more aware of options in relation to the research design, and hence to support researchers in their selection of a research design. Furthermore, the objective is to highlight the implications of the research design decisions, when conducting empirical research and provide a decision-making structure for empirical software engineering research. The decision-making structure contains a number of parts, each one of them representing a specific aspect on empirical software engineering research. The article provides an introduction to each part and its constituents, as well as to the relationships and potential consequences of taking certain decisions in relation to the different parts in the decision-making structure.

The remainder of the article is outlined as follows. Related work is presented in Section 2. In Section 3, a decision-making structure for decision points in research is outlined. The structure is further elaborated in Section 4. Given the decision-making structure, the different decision options and limitations are discussed in Section 5. The decision-making structure is illustrated with three examples in Section 6. The article is concluded with a discussion in Section 7 and some conclusions in Section 8.

2. Related Work

Several researchers have addressed the difficulties for selection of an appropriate research method in empirical software engineering research in the last two decades (Perry et al. 2000, Shaw 2003, Easterbrook et al., 2008). Perry et al. (2000) point out the lack of hypotheses, having no research questions and having no concrete solutions that can be related to a theory or practice as some of the problems in software engineering research. On the other hand, Easterbrook et al. (2008) discuss the main problem as the selection of the research method in empirical software engineering research because the pros and cons of research methods are not well documented and many researchers lack the knowledge about implications of the research methods that they use. Shaw (2003) emphasizes the lack of guidance for software engineering research. Some attempts have been made to both understand to what extent empirical methods are used in software engineering and to classify the approaches use, see for example Tichy et al. (1995) and Zelkowitz and Wallace (1997). These two studies took samples from literature. To conduct a full systematic literature review with respect to empirical methods used in software would be a major and challenging undertaking. Smite et al. (2010) conducted a systematic literature review in the area of global software engineering, and the findings were quite discouraging from an empirical point of view. The authors point out that it was not easy to deduce the methodological part, for example, both the empirical background and the methods of investigation were in many cases unclear. Thus, guidelines are indeed needed.

Over the last ten years, several researchers published guidelines for software engineering researchers.

Seaman (1999) encourages using a qualitative approach in research and provides guidelines of how qualitative data from interviews and participant observation can be used in empirical software engineering research. Kontio et al. (2004) present three studies that used focus group interviews and provide some guidelines of how to use focus groups. The authors argue that focus groups can be cost efficient as several participants are interviewed at the same time, and allow in-depth discussions in the meetings. This is a useful data collection approach as others can confirm the ideas of one participant, and participants can benefit from this interaction because it provides an environment where they can discuss their business perspectives. On the other hand common problems in group discussions such as group dynamics can affect the outcome of the interviews. Furthermore, some participants in the group may not be able to comprehend complex issues due to their level of knowledge.

Sjøberg et al. (2007) emphasise that empirical software engineering research can include both qualitative and quantitative methods for collecting and analysing data. According to the authors experimentation, surveys, case studies, action research and the secondary research in software engineering (e.g. systematic literature review and meta-analysis) are the primary approaches in software engineering research. Easterbrook et al. (2008) identify five research methods that are relevant to software engineering research, namely controlled experiment, case studies, survey research, ethnographies and action research. The authors emphasised the philosophical stance behind each method, based on the argument that this will help researcher selecting an appropriate research method. The authors point out the limitation of each research method and suggest that mixing methods can resolve this issue. Runeson and Höst (2009) present guidelines for conducting case studies in empirical software engineering research and point out that surveys, experiments and action research are the three major research methodologies that are related to case study which may contain surveys, literature review, archival analysis, interviews and observations as data collection approaches.

Perry et al. (2000) point out that no research is perfect and research design requires trade-offs among accuracy of interpretation, relevance, impact, resource constrains and risk.

3. Decision-making Structure

Here we take the view that empirical research follows a pattern of identification of problem/research question, study design involving several decision points and interpreting data and drawing conclusions as part of the findings of the research. Thus, empirical software engineering research is not only about conducting a specific empirical study. It has broader implications, including whether the objective is to conduct basic or applied research, the intention is to explain, describe, explore or evaluate the research issue studied, and the type of data to be collected to name a few. These examples illustrate that it is important to address a number of decisions points when conducting research in general and empirical software engineering research in particular. Based on this, we would like to introduce a decision-making structure to make researchers aware of different options and to support researchers in making informed decisions in relation to their design of research studies to answer their specific research questions.

In Figure 1, adopting the view from Smite et al. (2013), a researcher operates in a decision space where the researcher faces several decision points during the study design. The terminology used for decision points is based on Collis and Hussay (2009) classification as well as long discussion sessions between the authors of this article. However, it should be noted that there are a lot of different opinions and no consensus about terminology in research related activities. Mkansi and Acheampong (2012) provide a detailed discussion on inconsistent usage of terminology in literature. In software engineering, it could be exemplified with the misuse of the term "case study", which all too often is used to describe any example or illustration instead of solely being used according to the well accepted definitions as provided by for example Yin (2002), Bensabat et al. (1987) and Runeson et al. (2009). The choices made in this article can always be challenged, but the main objective is not to try to "standardize" the terminology; the main objective is to present the decision points and how the decision points may be put together into a decision-making structure as illustrated in Figures 1 and 2.

The bull's eye to the left in Figure 1 shows the starting point, the identification of the research question. Ideally once the research problem is identified, one or more research questions is defined. After this stage, the decision points involve selection of (i) research outcome[1], (ii) research logic, (iii) research purpose, (iv) research approach, (v) research process, (vi) research methodology, (vii) data collection methods and (viii) data analysis methods. These eight decision points are further elaborated in Section 4 where options in relation to each decision point is discussed in more detail. The decision points outline the structure of the decision process during study design. The decision points are logically ordered from left to right. However, it is common that decisions about the research design are not taken from left to right as is illustrated in Section 6. Our motivation in the

[1] This is an example of the different opinions related to terminology. Others may refer to this as "research type".

selection of decision points and the order of the decision points are largely determined by the nature of research design which requires a structured and efficient means to deal with the research activities (Collis and Hussay, 2009; Creswell, 2013) as well as the authors' experience which are supported by several examples in the paper.

The circles in Figure 1 show the options that are available at each decision point. These options offer alternative ways of designing research. Thus, the figure illustrates how different decisions generate a path through the decision space, and finally reaches the research findings, i.e. the bull's eye to the right in Figure 1. This is the ideal situation. In many cases, a researcher may decide to start with a decision on any of the decision points, which then may reflect on the other decision points, since there is a dependency between several of the decision points. We will refer to a decision taken somewhere in the path without the previous decisions being taken as a locking decision, since it locks the path to go through a specific point.

Note that in Figure 1, the decision space is fictitious and the number of options as well as the outlined strategy is given as an example of a study design in this decision space. The decision space and the actual options are further elaborated in Section 4.

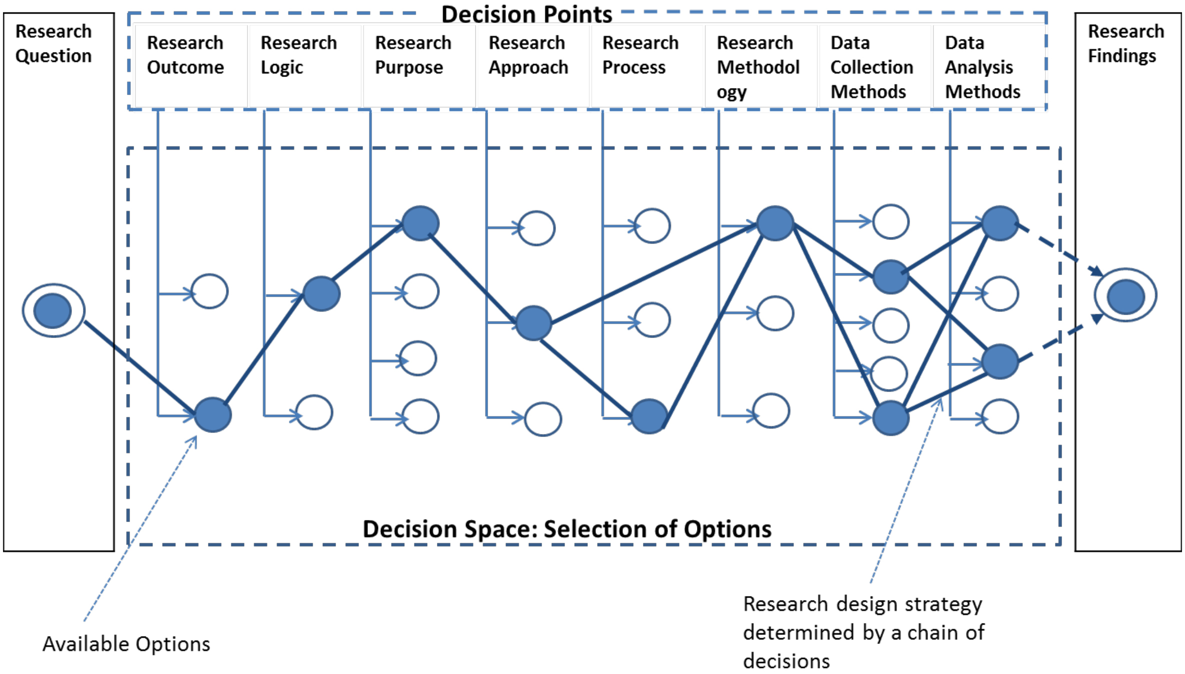

Figure 1: Decision points in research design

4. Research Design

A conceptual research decision-making structure is developed to guide our discussion for creating research designs and selecting appropriate research methods. Figure 2 illustrates the research decision-making structure. The decision points given in Figure 1 are aligned with Figure 2 i.e. this structure elaborates the decision points that were illustrated in Figure 1. The different phases, decision points and their options are described and discussed in the subsections below.

Research problems exist in the research domain itself or in practice and deal with behaviour or phenomena of interest. Once the researcher reviews the literature to understand the current state of knowledge in the study area and identifies a research problem, he/she then identifies the research question(s) that justifies the objective of the study. The researcher may need to go back and forward several times between the research question(s) and the gap identified in literature, until the gap is clearly defined and theories that may help the investigation of the research question(s) are determined.

Research Question(s)

Formulation of the research question(s) is critical. Firstly, it needs to address a significant and useful problem, which implies that the results are publishable (Chen and Hirschheim, 2004). Secondly, a research question determines, or strongly influences, the rest of the process in the research, including research methodology, data collection methods and data analysis methods. However, in practice, the research question(s) may evolve during the research and the researcher may need to adjust the research question(s) several times to fit with the results of their findings (Easterbrook et al., 2008; Runeson and Höst, 2009).

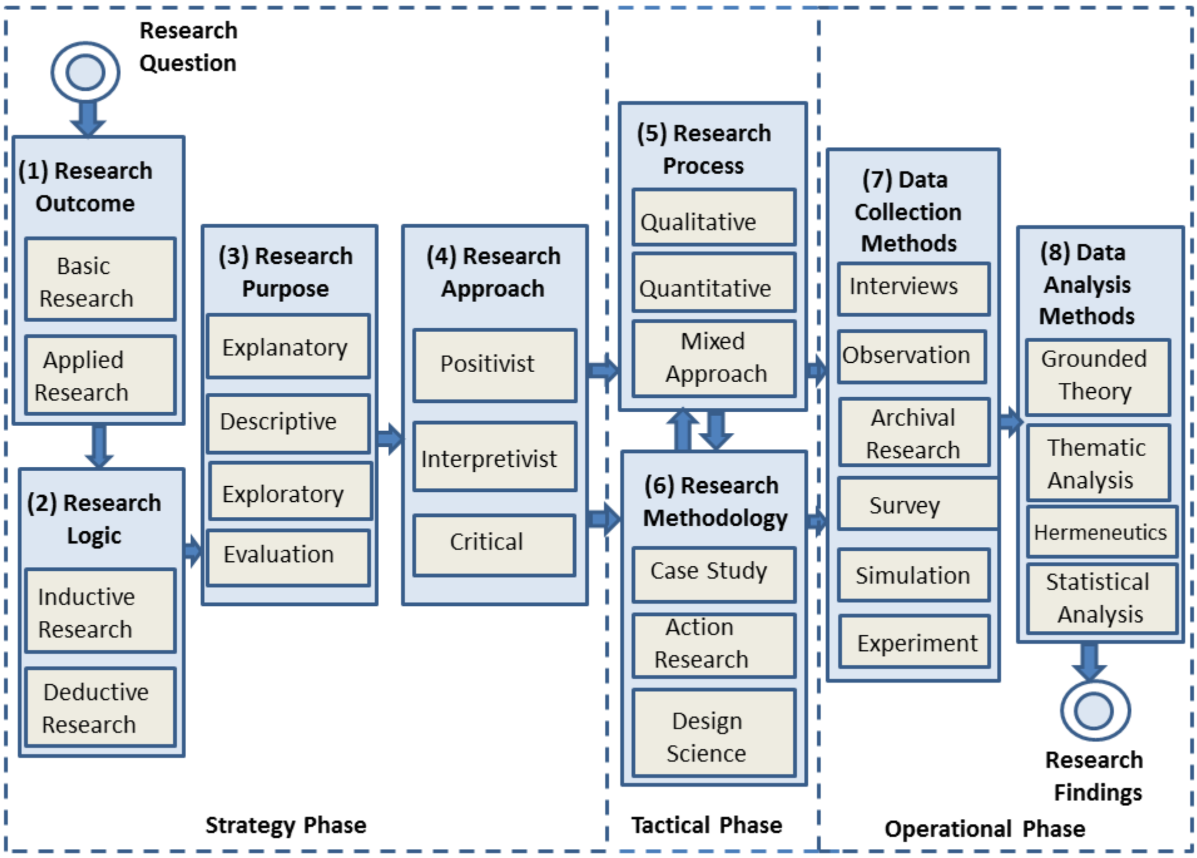

Figure 2. Research decision-making structure

A research question may be related to a set of hypothesis, concepts, or relationships between concepts or two phenomena that require clarification. Easterbrook et al. (2008) argue that in an early stage of the research, the research question(s) tends to be explorative, but once the researcher has a clear idea about the problem, the research question(s) tends to search for the patterns, the relationships between the two phenomena or search for a causal effect between the two phenomena. Shaw (2003), who lists the type of research questions that were commonly investigated in SE research, points out that the type of research question change overtime in empirical software engineering research.

Three Phases of Research

In Figure 2, decision points are grouped into three phases: strategic, tactical and operational.

(i) The strategy phase involves a plan that gives direction to the researcher for the tactical and operational phase of the research. This phase enables the researcher to conduct his/her research systematically, and to position the research in relation to different general approaches to research. An effective research strategy requires understanding the research topic as well as having knowledge about each decision point presented in Figure 2. The research strategy involves decisions on research outcome, research logic, research purpose, and research approach. The strategy phase sets the stage for the research.

(ii) The tactical phase involves decisions on how to operationalize the research activities in terms of how to approach the research question more specifically. The decision points are research process and research methodology. The tactical decisions enable the research to achieve the research goal. The tactical phase focuses on selecting the actual process and methodology to use to achieve the research goal.

(iii) The operational phase involves decisions on actions that will be taken when implementing the research, including data collection methods and data analysis techniques. Thus, actually planning the details and to collect the data to be able to respond to the stated research question(s). The operational phase is focused on actually carrying out the empirical research by collecting and analysing the data.

The research decision-making structure, in Figure 2, shows research from both a software engineering and an information systems perspective and its components are built from literature. Figure 2 show three phases of research with eight decision points. Each decision point offers several options. The decision options given in Figure 2 do not demonstrate the full coverage, although the objective here has been to cover the most commonly used approaches in software engineering. However, it is impossible to claim full coverage given the number of alternative ways research may be conducted. In other words at each decision point, there might be more decision options available to the researchers. Figure 2 displays only the options that are covered in this article. For example, at decision point 8, there is only four data analysis methods displayed. In reality there are more than four data analysis methods available to researchers, e.g. pattern recognition and protocol analysis. The decision points illustrated in Figure 2 are useful guidelines for researchers as it provides a common ground to them when they present their work. The following explains the eight decision points illustrated in Figures 1 and 2.

4.1 Research Strategy Phase

There are four decision points in the strategy phase, offering several options as outlined below.

4.1.1 Research Outcome (Decision Point 1)

The outcome of research can be classified as basic or applied research (Collis and Hussey, 2009). Basic research (a.k.a. pure research) is applied to a problem where the emphasis is the understanding of the problem rather than providing a solution to a problem, hence the main contribution is the knowledge generated from the research. For example, the researcher may want to investigate how risk management is handled in agile projects in general. This type of research tends to be less specific and the outcome of this type of research is knowledge (Collis and Hussey, 2009).

On the other hand, applied research is the type of research where the researcher provides a solution to a specific problem by applying knowledge with an aim of improving existing practice or application (Nunamaker et al., 1991; Collis and Hussey, 2009). Applied research cannot stand alone; it relies on basic research because it applies the scientific knowledge from basic knowledge in an existing practice (Bhattacherjee, 2012) For example, the researcher may want to investigate how a specific risk management approach in agile project can be customized to a specific company. The output from this research is likely to be a company specific solution, potentially with some experiences and conclusions of how it can be applied at other companies too.

4.1.2 Research Logic (Decision Point 2)

Research logic refers to in which direction the research proceeds in terms of whether it moves from general to specific or vice versa (Collis and Hussay, 2009). There are two common ways of reasoning in empirical software engineering research: deductive versus inductive research (Johnson, 1996; Shye, 1988; Creswell, 2013).

Deductive research works from the more general to the more specific. It allows researchers to establish hypotheses by using theory. The researcher collects data to confirm or reject the hypothesis. Deductive research is a top-down approach and often called theory-testing research. Deductive reasoning works from more general conclusion to more specific observation. Deductive research tends to lend itself to quantitative research as it aims testing a theory.

Inductive research is based on inductive arguments, it moves from the specific to the general. The researcher infers theoretical concepts and patterns from observed data. The researcher begins with specific observations, detects theoretical patterns, and develops some general conclusions or theories. It can be used, for example, when a researcher is trying to understand software processes, product, people and environment (Basili, 1993). This approach is also called theory-building research or a bottom-up approach (Bhattacherejee, 2012). Inductive reasoning works from specific observation to more general conclusion, which may lend itself to both qualitative and quantitative research

4.1.3 Research Purpose (Decision Point 3)

The purpose of research can be classified as exploratory, descriptive, explanatory and evaluation research (Collis and Hussey, 2009).

Exploratory research is applied when there is not much information available in the topic area and the researcher aims to gather some insights about the problem. The aim is to explore the problem area and provide background information that can be used for the descriptive or explanatory research. Explorative research can be both qualitative and quantitative research. Typical data collection methods are observation, interviews, and focus group interviews. An example of this type of research may be an investigation of technical debt in agile projects with the aim of managing project risk.

Descriptive research is, as its name suggests, applied to describe a phenomenon or characteristics of a problem. It is more focussed than exploratory research, and goes further than exploratory research. The research questions tend to start with "*what*" or "*how*" as it aims to describe the phenomena (Collis and Hussey, 2009). An example of a research question might be "How is technical debt managed in agile projects to mitigate risk?" The research may need to narrow down the research question later on e.g. only focusing on agile projects in a certain industrial domain.

Explanatory research is applied when examining the nature of certain relationships between the elements of a problem. The research questions tend to start with "*why*" as it aims to explain the phenomena. Answering "why" question may involve qualitative as well as quantitative research purpose such as using interviews, surveys, and document analysis. For example, the researcher may collect data on the size of the projects and the number of defects fixed in successful/failed agile projects. A statistical analysis of the data may display that the larger the project the more likely a project is to fail. The critical issue in explanatory research is to identify and control the variables in a way that causal relationships can be explained better (Collis and Hussey, 2009).

Evaluation research aims to determine the impact of methods, tools, or frameworks that may encompass the other three research purposes: exploratory, descriptive and explanatory research (Engel and Schutt, 2010). Evaluation research is a well-known approach in social science as it involves research methodologies to judge and improve the planning, monitoring, effectiveness and efficiency of human services or products they use (Rossi and Freeman, 1993). The research may use tools of research to describe, explore and assess the needs of different population groups (e.g. software teams, or employees who use a particular software), evaluating the effectiveness of a particular method, framework, or a software technology. Runeson and Höst (2009) denote this type of research as "improving" in the engineering context. In this article we call this type of research "evaluation research" as empirical software engineering research also aims to evaluate software technologies, frameworks and methods and gather information on the basis of systematic observation and experiments.

4.1.4 Research Approach (Decision Point 4)

Research paradigms are based on ontological (what we think can be researched), epistemological (how we know about it) and methodological (how we will acquire it) assumptions and determine the research inquiry or research approach, what it is and what fall within or outside the limits of the research inquiry (Grix, 2002; Myers 1997; Klein and Myers, 1999; Orlikowski and Baroudi, 1991).

Gregg et al. (2001) argue that different research paradigms do not address the requirements of software engineering research as they do in information systems research. On the other hand, Easterbrook et al. (2008) point out that even though many researchers in software engineering avoid addressing underlying philosophies in their approach; they tend to use one of the following four research approaches: positivist, constructivism (or interpretivist), critical research and pragmatism. Coleman and O'Connor (2007b) point out that positivist and interpretivist approaches are the most commonly used approaches in literature. In this article we address three approaches: positivist, interpretivist and critical research approaches, as we have seen several examples of these approaches in empirical software engineering and information systems research. These three approaches may in software engineering be exemplified with the following: Staron et al. (2006) use a positivist approach when conducting experiments on UML models, Moe et al. (2012) use an interpretivist approach when performing a case study in industry and Cecez-Kecmanovic (2011) argues for a critical approach and provides guidance for conducting critical research.

Positivist research works based on an assumption that the researcher and the reality are separate. This approach advocates an objective approach and believes that research is reliable if it can be repeated and another researcher would reach a similar conclusion (Klein and Myers, 1999). It assumes that the social world is made up of facts that can be studied like the natural world using a scientific approach. It aims to discover the truth and general laws of cause and effect in social behaviour (Klein and Myers, 1999). It tends to use quantitative methods. It tries to measure the world through the empirical data, formal propositions, and quantifiable measures of variables, hypotheses testing and the drawing of inferences about a phenomenon from a sample population (Orlikowski and Baroudi, 1991). Studies in this category can be based on theory or descriptive with no theoretical grounding or interpretation of phenomenon using case studies with some simple descriptive statistics (Orlikowski and Baroudi, 1991). Positivist research tends to fall in the explanatory category as a research purpose (Klein and Myers, 1999). Common methods are controlled experiments, surveys and archival data analysis. Case studies can also be conducted in the positivist approach; however as Easterbrook et al. (2008) pointed out the researcher needs to be able to show that the research is conducted in isolation from its context.

Interpretivist research aims to understand the human activities in a specific situation from the participants' perspective, hence it emphasises the context (Klein and Myers, 1999; Myers 1997). It rejects the possibility of "objective" research and believes that research can be subjective. It assumes behaviour is influenced by the *meanings* people attach to events (Orlikowski and Baroudi, 1991). It aims to understand the deeper structure of a phenomenon within cultural and contextual situations where the phenomenon is studied in its natural setting and from the participant's perspective without including the researcher's prior understanding of the situation. It assumes that validity of research can be gained by gathering qualitative data that is rich and in-depth (Orlikowski and Baroudi, 1991). It tends to use qualitative methods e.g. interviews or ethnographies (Myers, 1997). It is sometimes referred as constructivism. An interpretive case study or a survey may also fall in the exploratory and descriptive categories as a research purpose (Easterbrook et al., 2008).

Critical research aims to critically evaluate existing system, based on the assumptions that social and cultural variables impact the existing system and that the interconnections cannot be ignored. In critical research, knowledge is considered subjective, depending on whose perspective the researcher takes and whose eyes view the problem (Brooke, 2002; Myers and Klein, 2011). Critical research aims to reveal contradictions and conflicts within the existing system while positivist and interpretivist research aim to predict or explain the current situation (Orlikowski and Baroudi, 1991). The critical research often involves long-term historical studies of organization processes and structure. It tends to use qualitative methods and is likely to be a longitudinal study (Orlikowski and Baroudi, 1991). Easterbrook et al. (2008) discuss how well case studies and in particular action research reflect the philosophy of critical research.

Klein and Myers (1999) point out that research can be classified as positivist "if there is evidence of formal propositions, hypothesis testing, and the drawing of inferences about a phenomenon from a representative sample to a stated population"; it can be classified as interpretivist "if it is assumed that our knowledge of reality is gained only through social constructions such a language, shared meanings, documents, and other artefacts"; it can be classified as critical "if the main task is seen as being one of social critique whereby the restrictive and alienating conditions of the status quo are brought to light".

4.2 Research Tactical Phase

There are two decision points in the tactical phase: research process and research methodology. The researcher can move from the tactical phase to the operational phase by following one of three paths: (i) the researcher selects one of the research processes (decision point 5) and then one of the research methodologies (decision point 6) and moves to data collection methods (decision point 7), (ii) the researcher selects one of the research processes (decision point 5) and moves to research methods (decision point 7); or (iii) the researcher selects one of the research methodologies (decision point 6) and moves to data collection methods (decision point 7).

4.2.1 Research Process (Decision Point 5)

In general, there are two widely recognized research processes called quantitative research and qualitative research. An alternative option is the combination of both qualitative and quantitative research, denoted as mixed research (Creswell, 2013). The distinction between qualitative and quantitative research is ambiguous but the use of the distinction provides a helpful umbrella for a range of issues concerned with the social aspects of empirical software engineering research. The distinction between qualitative and quantitative research comes not only from the type of data collected but also the research approach, its objectives, types of research questions posed, evaluation of research and the degree of flexibility built into the research design as well (Mack et al., 2005; Creswell, 2013).

Qualitative research is a matter of inquiry that aims to study the social and cultural phenomena. It is conducted when **a** researcher aims to understand the perspectives of their research subjects. The main idea is that by gaining access to the perspectives of insiders, researchers can also gain access to new ways of seeing the world (Hannah and Lautsch, 2011). Qualitative data refers to verbal descriptions by reflecting the world as seen by participants. Qualitative research involves the use of qualitative data collection such as interviews, written documents and participant observation to understand and explain social phenomena. Qualitative methods are well suited for building theory, writing rich descriptions, explaining relationships, and describing groups of norms e.g. standards, models and frameworks. Klein and Myers (1999) argue that the interpretative approach is not a synonym for qualitative methods. The authors discuss that qualitative research can be done with a positivist or interpretive approach. The same way a case study or action research can be positivist or interpretive.

Quantitative research involves studies that refer to collecting quantitative data directly or cases where qualitative data is quantified to allow, for example, for statistical analysis (Pinsonneault and Kraemer, 1993; Klein and Myers, 1999). The quantification of qualitative data is one form of a mixed research process.

The objective is to describe the characteristics of the population and in many cases predict causal relationships. Quantitative research emphasizes using metrics, measuring with numbers, and analysing data by using statistical techniques. Examples of quantitative research methods include questionnaires, experiments, and simulation.

Researchers argue that in today's complex environment, existing research paradigms cannot adequately cover the development and implementation of innovative software in an organizational or individual context (Wynekoop and Russo, 1997; Mingers, 2001; Johnson and Onwuegbuzie, 2004). It has been acknowledge that the use of multiple research approaches is necessary to adequately investigate the software development process. A mixed research approach involves studies collecting both qualitative and quantitative data. Mixing may involve not only the type of research process, research methods, and data analysis methods (Creswell, 2013). .

4.2.2 Research Methodology (Decision Point 6)

An essential part of research is the decision on research methodologies, which encompasses the combination of research methods, processes and frameworks. Research methodologies receive varied attention from researchers, which lead to some confusion and inconsistencies in the views of researchers about research methodologies and research methods. For example, Runeson and Höst (2009) indicate survey, case study, experiment, and action research as research *methodologies*. On the other hand Sjøberg et al. (2007) introduce experiments, surveys, case studies and action research as common research *methods* in software engineering research. In a similar way Easterbrook et al. (2008) introduce controlled experiments, case studies, surveys, ethnography, action research, mixed methods as common research *methods* in software engineering research.

In this article we address three research methodologies: case study research, action research and design science research which all may consist of a combination of research methods, processes and frameworks.

Case Study Research

Case study research (a.k.a. case study) is a research inquiry that employs multiple methods of data collection to gather information from a variety sources with an aim of investigating a phenomenon (contemporary or historical) in its natural settings (Benbasat et al., 1987). A case study involves several steps including designing, conducting case study and analyzing data & developing conclusion (Yin, 2002). Different researchers perceive the definition of case study differently. For example, Collis and Hussay (2009) discuss case studies as explorative, descriptive, experimental and explanatory.

Shanks and Parr (2003) discuss positivist case studies that involve experiments. However, Benbasat et al. (1987) argue that by the nature of case studies, experiments are less likely to be part of case studies, i.e. it is unlikely that the researcher will have priori knowledge of what the variables of interest will be and how they will be measured in case studies.

In this article, action research and design science research are not included as part of case study research but rather as separate research methodologies. In a similar way, Sjøberg et al. (2007) also consider action research as a separate research methodology. However, Runeson and Höst (2009) discuss action research as part of case study research in software engineering.

Case study research is one of the most commonly used methodologies in both software engineering and information systems research (Runeson and Höst, 2009; Orlikowski and Baroudi, 1991; Kitchenham et al., 1995).

Since research in empirical software engineering focuses on software development activities or evaluating the use of technology, methods and so forth by stakeholders, the research questions related to these activities are suitable for case study research (Sjøberg et al., 2007; Runeson and Höst 2009). Case study research provides richer and more contextualized interpretation of the phenomenon than for example an experiment. Case study research is appropriate if the phenomenon can be studied from the perspectives of multiple participants and using multiple levels of analysis (Eisenhardt, 1989). Case studies can differ based on its research approach, time period that it is investigated (contemporary vs. historical), data sources, unit of analysis, length (short period vs. longitudinal study) and research process (Runeson and Höst 2009; Yin, 2002; Eisenhardt 1989). Hence, it is essential that the reasons for selecting the case study methodology be explained, as there are various ways of handling case study research.

For example, case study research can be positivist for the purpose of theory testing or non-positivists, e.g. interpretivist for theory building (Shanks and Parr 2003; Kyburz-Graber, 2007). Positivist case studies tend to be explanatory, and non-positivist studies tend to be descriptive or exploratory (Perry et al., 2000; Lee, 1989; Klein and Myers, 1999; Lyons 2009; Perry 1998).

Case study research can be conducted as a single case or multiple case studies. Yin (2002) suggests that the study of a single case is appropriate if it is a revelatory case, a critical case or a unique case and multiple case studies are appropriate if a replication is possible in multiple sites. Eisenhardt (1989) suggests several strategies for cross-case analysis by using data in many different ways, e.g. examining similarities or differences or variations between the data points from various cases.

Case study research can use quantitative research, qualitative research or a combination of both, i.e. a mixed research approach. Case study research tends to address 'How' and 'Why' research questions (Yin, 2002). Sjøberg et al. (2007) point out that case study research can also be useful for answering "which is better" type of research questions.

Eisenhardt (1989) discusses the role of theory in case study research. According to Eisenhardt, the theory can be used: (i) as an initial guide to design and data collection, (ii) as part of an iterative process of data collection and analysis, or (iii) a final product of research. The selection of cases is important when building theories from cases (Eisenhardt, 1989). The advantage of building theory from case study research is that theory testing is likely to be possible and empirically valid but it may have lack of simplicity and its generalizability may be problematic (Eisenhardt, 1989).

In case study research, choosing an appropriate unit of analysis, which might be a project, software team, an individual stakeholder, a specific type of work or the organization itself, is critical to ensure that the research question(s) is adequately addressed (Easterbrook et al., 2008).

Action Research

Action research involves "solving organizational problems through interventions while at the same time contributing knowledge" (Davison et al., 2004). In action research, a research problem is generally introduced by the organization. The researcher systematically studies the problem, people and the organization, and the research problem exists in the real-world context (Davison et al., 2004).

Susman and Evered (1978) provide a structure to action research, named as canonical action research, by adding a cyclic process in five stages: 1) diagnosing, 2) action planning, 3) action taking, 4) evaluating, and 5) specifying learning. In order to be able to evaluate the research quality, Davison et al. (2004) introduce five principles: 1) researcher–client agreement, 2) cyclical process model, 3) theory, 4) change through action, and 5) learning through reflection. In other words, in action research, the researcher initiates an action in response to an organizational problem, and investigates the problem using a scientific approach, and examines how his/her actions/solutions influence the phenomenon while also learning and generating insights about the relationship between the action and the phenomenon.

Action research requires that the researcher is part of the organization during the investigation, solution development and application of the solution. Examining a problem in real-world context is the strength of this methodology; however, the potential lack of objectivity on part of the researcher forms a weakness for this methodology (Sjøberg et al., 2007). Since action research is always conducted in a real-world context, where the problem is encountered, it is usually a single case study (McKernan, 1996).

Design Science Research

Design science research (DSR) addresses through building and evaluating artefacts designed to meet the requirements of a problem. Hevner et al. (2004) introduce design science research, as a problem solving process, which requires a creation of an artefact for a specific problem in which the artefact needs to be innovative, be effective and needs to be evaluated by applying rigorous approaches. The output of DSR is an artefact, in the form of a construct, model, method or an instantiation. Ostrowski and Helfert (2011) found in their research that 78% of researchers constructed artefacts in their DSR from literature review and working with practitioners and 22% of researchers constructed artefacts from the literature review only.

DSR is not action research; however, there is an overlap on some stages of the research approach. Action research focuses on problem solving through social and organization changes which requires identification of a problem in a real-world context, identification of requirements from stakeholders and providing a solution that brings changes to the organizations. While DSR also focuses on identification of a problem, it aims to bring a solution as an IT artefact (e.g., a model or a framework) that becomes part of the knowledge base of researchers (Baskerville, 2008).

While Baskerville (2008) argues that DSR is not a methodology, Ostrowski and Helfert (2011) discuss the various DSR methodologies. In this article we take the same approach as Ostrowski and Helfert (2011) and present DSR as a methodology. The research on the study of software development lies at the heart of design science. DSR provides a means to explain the study of software products in empirical software engineering research and offers a guideline to create, improve, and evaluate IT artefacts (Hevner, et al. 2004; March and Smith 1995). The research purpose of DSR tends to be explorative as the creation of the artefact is essential before it can be evaluated. DSR can use both qualitative and quantitative research approach.

4.3 Research Operational Phase

The operational phase research refers to the process to use data collection and data analysis when investigating a research question. There are two decision points in the operational phase, data collection methods and data analysis methods, offering several options as outlined below. The researcher may choose one of more options from each decision point.

4.3.1 Data Collection Methods (Decision Point 7)

The data collection method depends on the research question (Benbasat et al., 1987). Data collection methods may involve qualitative or quantitative data. Some commonly used qualitative data collection methods in empirical software engineering research are interviews and participant observation (Seaman 1999). Some commonly used quantitative data collection methods are archival research, surveys, experiments, and simulation (Wohlin et al., 2012). There are also other data collection methods, e.g. think aloud protocol, work diaries, organizational repositories, brainstorming, and the Delphi technique are also some examples of available research methods. Further details of data collection methods in empirical software engineering can be obtained from Shull et al. (2008), Runeson et al. (2012), Wohlin et al. (2012) and Lethbridge et al. (2005). In this article we provide a brief summary of some of the data collection methods that we believe most relevant to empirical software engineering research.

Interviews

Interview is a data collection method of eliciting a vivid picture of a participant's perspective on the research topic and it involves a meeting in which the researcher asks a participant a series of questions. This method is useful as the researcher can ask in-depth questions about the topic and also follow-up on the topic with participants (Seaman, 1999).

Besides face-to-face interviews, phone interviews and interviews through Internet, e.g. Skype or Google Talk, are common ways of interviewing participants. Interviews can be with individual participants or with focus groups. Interviews can be structured, semi-structured or unstructured. Structured interviews tend to have close-ended questions. Unstructured interviews have open-ended question, typically beginning with "what" and "how" type of questions and are mostly used in in-depth interviews to capture the participant's experience. In any type of interview, the researcher must be well prepared before conducting the interviews and make plans on several issues, e.g. interview questions, length of the interviews, recording interviews, and selection of participants (Runeson and Höst, 2009). Interviews can be recorded manually through handwritten notes or using tape-recording with permission of the participants. Then the interview data is transcribed and analysed using one of the qualitative data analysis techniques.

Interviews are a commonly used research method in both empirical software engineering and information systems research, as part of case studies, action research, design science research or as a qualitative data gathering research method in its own right.

Participant Observation

Participant observation is another qualitative data collection method for eliciting non-verbal data, e.g. participants' feelings, their daily activities, their interaction with other people in the work environment, including how they communicate with each other and how long it takes to complete a task. This research method also allows researchers to observe events that the participant may be unable to articulate, or may be unwilling to articulate due to shyness or other personal reasons.

It is important that the researcher (i.e. observer) retain the level of objectivity when gathering data, analysing the data and interpreting the data (Seaman, 1999). The researcher can carry out observation in a covert (i.e. participant is not aware of the observation) or overt (i.e. participant is aware of the observation) manner. In case study research, observation can be either covert or overt. In action research, observation tends to be overt. Runeson and Höst (2009) classify the researcher-participant interaction as high degree vs. low degree interaction and participants' awareness as high or low level of awareness. The authors point out that in action research researcher-participant interaction tends to be high with high or low level participant's awareness.

Archival Research Method

Archival research is the investigation of hard historical data that are archived by someone or by organisations. It may require permission to access the data or it can be publically available. The method involves locating the data, systematically collecting the data, analysing and interpreting the data.

Archival data can be qualitative or quantitative. Examples of qualitative data include meeting minutes, e.g. Scrum meeting minutes, and organizational data, e.g. policy documents. Examples of quantitative data include measurements that are collected from, for example, project databases either in a company or from a website for an open source software project or from SourceForge.net or Githup.com

Archival data is also used in both case study research and action research. It tends to be combined with other data collection methods as the data may have some missing components (Runeson and Höst, 2009).

Survey Research

Survey research (a.k.a. questionnaire) involves examination of a phenomenon in a wide variety of natural settings (Pinsonneault and Kraemer, 1993). It is a quantitative research method that is commonly used in both empirical software engineering and information systems research. It may elicit data from a variety of sources including individuals, groups or organizations. According to Pinsonneault and Kraemer (1993), the three most important aspects of survey research are: purpose of survey research, unit of analysis, and representative sample. Kitchenham and Pfleeger (2002) point out that understanding the objective of the survey is the starting point and this must be clearly expressed by the researcher. Furthermore, understanding survey participants, asking an appropriate number of questions, standardizing the participants' response format are also critical factors that need to be considered (Kitchenham and Pfleeger, 2002).

Easterbrook et al. (2008) discuss the importance of research questions in survey research and also address certain characteristics of surveys, including selection of a representative sample from a well-defined population. In survey research, the researcher must have clearly defined dependent and independent variables before testing a specific model against observations of phenomenon (Pinsonneault and Kraemer, 1993).

Survey research can be exploratory, descriptive or explanatory. Pinsonneault and Kraemer, (1993) argue that exploratory surveys are generally conducted to identify concepts to be measured and used in developing descriptive concepts for explanatory surveys. A general guideline for conducting survey research can be found in Kitchenham and Pfleeger (2002).

Experiments

Experimental research is characterized by random assignment of participants to experimental conditions in a controlled environment. Quasi-experiments involve non-randomized assignment of participants to experimental conditions. Experiments are often conducted to compare two or more treatments, for example, specific methods, techniques or tools. Subjects (participants) are assigned to use one or more treatments, and the objective of the researcher is to be able to analyse the data collected with statistical methods.

Experiments can be laboratory experiment or field experiment. Laboratory experiments are conducted in controlled environments whereas field experiments are conducted in a real-world setting. Both cases usually involve having different groups being assigned to different treatments to be able to contrast the precise relationships among variables through collection of data during the experiment. Laboratory experiments suit well to research projects that involve relatively limited and well-defined concepts and research questions that involve individuals or small groups (Pinsonneault and Kraemer, 1993).

Experiments can be used within research methodologies as case studies, action research as well as design science research, or experiments can be run stand-alone. Although controlled experiments are used quite often in empirical software engineering research, several researchers report lack of rigour, poor experimental design, and inappropriate use of statistical techniques and conclusions that did not follow from the reported results generate major problems in experimental software engineering research (Shaw, 2003; Kitchenham et al., 2002). Easterbrook et al. (2008) argue that researchers must really think hard before conducting experiments to ensure that experiments are effectively used to improve software development.

Further information on experiments can be found from Wohlin et al. (2012) and Kitchenham et al. (2002).

Simulation

Simulation is the use of a model of some real world entity. In software engineering, it may involve creating a model of a system that describes processes, products or people. Simulation could, for example, be used to evaluate alternative implementations or different ways of working. The simulation model reproduces the actual operations of the real components of the system with varying degrees of accuracy. Simulations help answering "what if" questions.

Simulation can use a deductive or an inductive approach. In the deductive approach, a model is generated based on a theory and the model is run, output from the simulation is compared with observations where the output follows directly from the assumptions made (Gilbert, 1995; Harrison et al., 2007). In the inductive approach, a model is generated based on gathered data; the model is run to examine its implication, output relationships between variables may be inferred from analysing the output data (Gilbert, 1995; Harrison et al., 2007).

Moreover, according to Law (2007), simulations could be classified in three dimensions: static vs. dynamic, discrete vs. continuous, and deterministic vs. stochastic simulation models. Static vs. dynamic simulations refer to whether or not the system under study is time-dependent or not. Discrete vs. continuous simulations relates to whether events happen at specific discrete points in time or the simulation is continuous in time. Finally, deterministic simulations are simulations without any probabilistic behaviour, while stochastic simulations depend on probabilities for different events and outcomes.

Müller and Pfahl (2008) discuss how simulation can provide insights to the software development design process and project before significant time and cost have been invested.

4.3.2 Data Analysis Methods (Decision Point 8)

Through the research methods, a lot of data may have been collected in qualitative or quantitative form. The data provide insight and evidence into the phenomenon studied. Once the data are collected, the researcher needs to analyse the data by using qualitative or quantitative data analysis techniques.

Qualitative analysis methods, e.g. thematic analysis, and hermeneutics may use qualitative data such as text. The grounded theory method, also sometimes referred to as a research method, may use both qualitative and quantitative data. The emphasis on qualitative analysis methods is "sense-making" or understanding the phenomenon (Bhattacherejee, 2012). These methods heavily depend on the researcher's analytical skills and her or his creativity. In some cases data analysis is conducted in parallel to the data collection process (Runeson and Höst, 2009). During the data analysis, new insights can be identified which may require gathering more data. Quantitative analysis methods, e.g. statistical analysis and mathematical modelling approaches use numeric data. These methods heavily depend on the researcher's technical and analytical skills. In this article we provide a brief overview of some of the frequently used data analysis methods in empirical software engineering research.

In both qualitative and quantitative analysis, data preparation is an important step for the data analysis. Some of the phases of data preparation are similar in these analyses. For example, in qualitative data analysis the first step is coding the data.

Coding quantitative data is the process of systematically capturing and scoring each data point in order to make the data analysis simpler. The researcher tend the decide how to code the data before data is collected and analyse data using one of the quantitative data analysis techniques e.g. regression or structural equation modelling.

Coding qualitative data is the process of identifying themes in the text and assigning labels to each theme where the emerging themes become categories of analysis. A theme is something important about the data in relation to the research questions. For example, if the researcher aims to investigate risk management in agile projects, the themes from transcribed interview scripts may be "time pressure" and "frequent release". Themes show the patterns in the data. The coding process enables the researcher to group text based on themes identified from the text so that the themes can be studied and compared.

There are three types of coding process: open coding, axial coding and selective coding (Strauss and Corbin, 1998).

- Open coding is the process of identifying themes within the data by reading and then assigning codes to each sentence. The codes may be revised or merged with other codes as the researcher continues the coding.
- Axial coding is the process of categorizing and ordering codes and identifying causal relationships between the codes or forming hypothesis that can explain the phenomenon.
- Selective coding is the selection of one code as the main code and relating the rest of the codes to the main code.

Thematic Analysis

Thematic analysis is widely used as a qualitative data analysis technique in empirical software engineering research as it provides deeper understanding about the data content.

Braun and Clarke (2006) describe thematic analysis as a method for identifying, analysing and reporting themes within data. The authors describe six phases of thematic analysis process: familiarising yourself with the data, generating initial codes, searching for themes, reviewing themes, defining and naming themes, and producing the report. Thematic analysis generally involves open coding where the codes are used to organise themes. Braun and Clark (2006) classify thematic analysis as "semantic" and "latent" themes. Semantic themes are identified based on explicit meaning in the data. The researcher searches for patterns using semantic themes. On the other hand in latent themes, the researcher searches for underlying ideas within the data that is theorised. Braun and Clark, (2006) suggest that semantic themes tend to be used in the positivist research approach paradigm whereas latent themes tend to be used in interpretivist research approach. Researchers identify themes and assign codes to themes manually or using software packages, e.g. NVivo. A guideline for thematic analysis can be found in Braun and Clark (2006). Ideally thematic analysis should involve several researchers with themes being developed by using discussions with interview subjects or other researchers to capture the multiple perspectives from various people.

Grounded Theory

The grounded theory method (GTM), developed by Glaser and Strauss (1967), is a data analysis method that involves analysing data by coding, constantly comparing and categorizing, and interpreting data to build theories and interrelated hypotheses. GTM is a systematic way of generating theory based on observed patterns and involves open, axial and selective coding where the coding can be handled in parallel rather in sequential order. GTM is heavily dependent on the observational and interpretive abilities of the researcher (Glaser, 1992). The resulting theory may be subjective and non-confirmable.

GTM is a process oriented, and systematic data analysis method (Birks et al., 2013). It encompasses collecting, coding and categorising data until saturation is reached, writing memos to identify potential relationships between the codes, and illustrating the relationships between the concepts that lead to emergent theory (Glaser and Strauss, 1967). In this method data collection and data analysis is an iterative process (Birks et al., 2013). The design of interview questions is critical importance as the researcher can reach saturation in one question if not done properly. GTM is appropriate for topics that have been little explored (Walker and Myrick, 2006; Urquhart, 2002; Dunne, 2011).

Although GTM is originally developed by Glaser and Strauss (1967), over the years, the way the coding and data interpretation process is carried out differently by Glaser (1992) (a.k.a. Glaserian approach) and Strauss (Strauss and Corbin 1998) (a.k.a. Straussian approach) hence creating two version of the grounded theory approach (Walker and Myrick, 2006; Urquhart, 2002; Birks et al., 2013). While the Glaserian approach promotes identifying participants' differing perspectives from the data at an abstract level, and conceptualizes them to find a latent pattern, the Straussian approach encourages, during axial coding, development of categories under headings, e.g. conditions, context, action/interaction strategies or headings that come from literature or the researcher's experience (Strauss and Corbin, 1998). The main difference between the two approaches occurs between the forcing and emergence of categories (Birks et al., 2013). Both approaches emphasize the importance constant comparison of concepts that leads to conceptualization (Glaserian approach) or descriptions (Straussian approach).

GTM is increasingly used in empirical software engineering research in particular in the last few years as it helps researchers to build theories and hypotheses from the data and test them in another study (Adolph et al., 2011; Coleman and O'Connor, 2007a; Coleman and O'Connor, 2007b). For example, Coleman and O'Connor (2007a, 2007b) used the Straussian approach to examine software process improvement in an industrial setting. Ghanam et al. (2012) also applied the Straussian approach when investigating issues and challenges organizations face when adopting software platform as a reuse strategy. Adolph et al. (2011), on the other hand, used the Glaserian approach when investigating the effect of social processes on team performance and the effect of software methods. Their findings showed that GTM is an effective data analysis method for examining behavioural aspects of people in empirical software engineering research.

There is no consensus among researchers about how GTM should be used; hence the analysis of GTM is open to interpretation. On one hand, the absence of clear guidelines to GTM brings flexibility to the researchers if they wish to apply GTM, on the other hand, it generates a dilemma to those who do not have enough experience and knowledge in this specific data analysis methods; hence it makes it hard to novice researchers to understand and conduct a proper GTM. We argue in this article that the researchers must opt for the approach that is most appropriate for their own research. It is important to clearly describe the key steps of GTM and to provide evidence for data analysis e.g. illustrating the evidence from the coding process, constant comparison of data and theoretical sampling.

Hermeneutics

Hermeneutics is a data analysis method for understanding and interpretation of qualitative data. The method requires reading data, understanding the small parts of the data that ultimately leads to understanding of a larger whole. The hermeneutic circle refers to the dialectic between the understanding of the text as a whole and interpretation of its part (Myers, 1995). Although the method is originally applied for interpretation of religious text in the 19th century, modern hermeneutics involve all types of data including verbal and non-verbal data (Boell and Cecez-

Kecmanovic, 2010) and is used in many disciplines that involve human behaviour including information systems and software engineering research.

Boell and Cecez-Kecmanovic (2010) argue that hermeneutics provide a framework for ad hoc literature review because an ad hoc literature review (as opposed to a systematic literature review) is an open ended process of reading and understanding, it is an iterative process and has a cyclic nature. Myers (1995) offers using dialectic hermeneutics as a framework to understand the social and political process in information systems development.

In the software engineering discipline, the application of hermeneutics goes back to Checkland's (1981) systems thinking approach in soft systems applications. Hansen and Rennecker (2010), in their explorative study, propose hermeneutics as a method when investigating how software teams arrive at a common understanding in their daily activities. Allison and Merali (2007) applied dialectic hermeneutics when investigating the dynamics of software process improvement in a case study.

Klein and Myers (1999) point out the lack of principles for evaluation of hermeneutics data in case studies and provided a set of principles as a guideline for using the hermeneutics method. Butler (1998) discusses the fundamental differences between different perspectives that can be applied in hermeneutics including conservative, constructivist, critical and radical. The author emphasizes that researchers who apply hermeneutics must clearly explain which perspective that they use in their studies.

Statistics

Statistics is used for analysing quantitative data. The data can be collected in quantitative form and statistics is applied to analyse the data. Alternatively, the data is collected in qualitative form and then converted to quantitative form, e.g. by using frequency of words, events, people, themes and then statistics is applied to analyse the data. Data can be analysed using descriptive or inferential analysis.

The descriptive analysis involves summarizing data by describing, and aggregating data and presenting association between the constructs. Mean, median, mode, average, deviation and variance are examples of methods used in descriptive analysis as well as different types of plots such as box plots.

The inferential analysis involves, for example, statistical testing of hypotheses, regression analysis and estimation using data mining techniques. Hypothesis testing is used to make inferences about a population. Regression analysis refers to methods that help to understand how changes in one variable affect another variable. Data mining is an automatic or semi-automatic approach to discern interesting data patterns using different statistical techniques such as cluster analysis.

Moreover, statistical methods may be parametric or non-parametric. Parametric methods make assumptions on the distribution of data, while non-parametric are more general. If the assumptions of the parametric methods are correct then they can normally provide more accurate and precise estimates than non-parametric methods.

Further information about statistics can be found in the numerous books on statistics, for example, Marascuilo and Serlin (1988), and books focused on specific areas of statistics such as Kachigan's (1986) book on univariate and multivariate methods, Siegel and Castellan's (1988) book on non-parametric statistics and Kline's (2011) book on structural equation modelling.

5. Decision Space and Limitations

In the previous section we provided brief explanation of each decision point with some alternatives.

There are several alternatives available at each decision point. Ideally, the researcher is expected to start the research from the first step, i.e. after identification of the research question, and then follow the decision process from decision point one to decision point eight. However, as can be seen from the examples provided below (Section 6), it is very common to make the decisions in other orders. Based on the decision alternatives available in Figure 2, it is possible to create several research decision paths. In Figure 2, there are 2*2*4*3*3*3*6*4 =10368 paths available, although not all combinations are viable. Furthermore, the researcher's selection of decision alternative at each decision point, in Figure 2, is influenced by the availability of alternatives to the researcher. In practice, several factors such as the availability/accessibility of data sources, researcher's knowledge and experience about the research methods and methodology would influence the study design. Hence, the number of research paths and number of decision points are fewer than what we provide in Figure 2. The following items are some examples of why not all paths are viable:

(1) For example, Figure 2 shows eight decision points, in some cases this may be reduced to 7 decision points. Once the researcher decides which research approach (Decision point 4) to use in the strategy phase, he/she can move to the tactical and operational phase using one of the following paths:

- Research approach (decision point 4) →research process (decision point 5) →data collection methods (decision point 7), or
- Research approach (decision point 4) → research methodology (decision point 6) → data collection methods (decision point 7), or
- Research approach (decision point 4) → research process (decision point 5) →research methodology (decision point 6) → data collection methods (decision point 7)

(2) The research question may imply a certain combination in decision points. For example, if the researcher wishes to examine "factors that influence decisions of advertisers to employ search engine advertising", he/she is likely to conduct basic research, which will use a deductive approach and will be explorative research (e.g. Jafarzadeh et al., 2011 and 2013). The researcher will develop a model from the literature, set up some hypotheses, conduct a survey to test the hypotheses and apply statistics as a data analysis method.

(3) There are dependencies between the options available at each decision points. Selection of certain alternatives may have some implications in the following decision points. For example, if the researcher uses simulation as a research method, grounded theory would not be used as a data analysis technique.

(4) Availability of resources may affect the selection of decision alternatives. For example, the researcher who has lack of access to human subjects may decide to choose archival data as a research method. This type of selection would force the researcher to revise the research question and chose appropriate research methodology and research process that will go with archival data analysis. Another example includes the researcher who already decided to conduct interviews and observation as a data collection method (decision point 7). This decision eliminates quantitative and mixed approaches from research process (decision point 5), most probably action and design science from research methodology (decision point 6), positivist approach from research approach (decision point 4), exploratory and evaluation from research purpose (decision point 3), inductive logic from (research logic) and applied science from (research outcome)

6. Examples of Using the Decision-making Structure

This section presents three examples of research that represent three different research paths through the research decision-making structure. These examples are chosen because the authors were involved in supervising the students and the research findings have been published in good journals. Although these three examples cannot represent the whole universe of possible research paths through the structure in empirical software engineering, they are useful for suggesting how the trade-offs between the decision points, can lead to different research paths. In Figures 3-5, the direction of arrow shows the direction of the decisions made. Arrows that point in both directions show the iterative approach between the decision points. Each example is illustrated in a figure followed by a short description of the decisions made.

Example 1

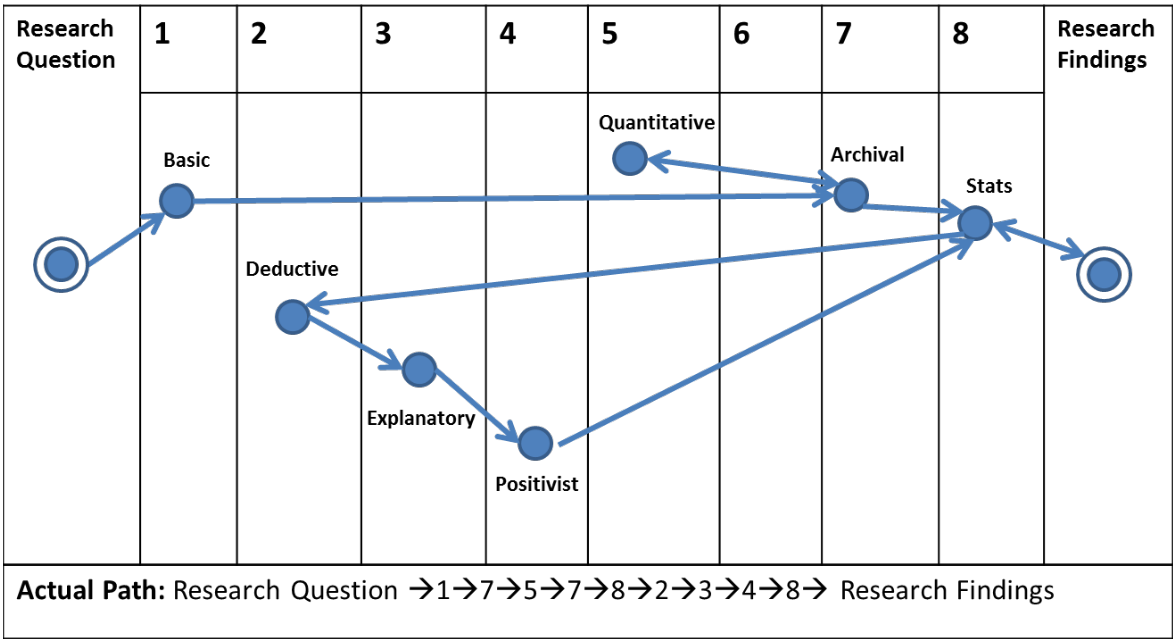

Figure 3. Research path for Example 1

Student A decides to conduct research on project management using quantitative research methods because Student A has a strong background on both topics. The research aims to examine the factors that affect the success of open source projects. Once completing the literature review, identifying the gaps, and defining the research question(s), the student decides to conduct basic research (decision point 1). The student realises that it is difficult to reach the users of open source software projects; hence it is better to use archival data as a data collection method (decision point 7). The student decides to apply a quantitative approach as research process (decision point 5) because he has a strong background on this topic. He also decides to apply statistical analysis as a data analysis technique (decision point 8). Once he determines how his research will be operationalized, the student determines that this research will be deductive research (decision point 2) as he will develop a model and a set of hypothesis to draw inferences about the factors that impact the success of open source software projects. Deductive research fits well to an explanatory study as a research purpose (decision point 3) and positivist approach as research approach (decision point 4).

Figure 3 illustrates the path that this student took in this example. Further information about this research can be found from Ghapanchi (2011). In this example, the student's background on the research area, research methods and the availability of data sources were the "underlying factors" that affected the research path.

Example 2

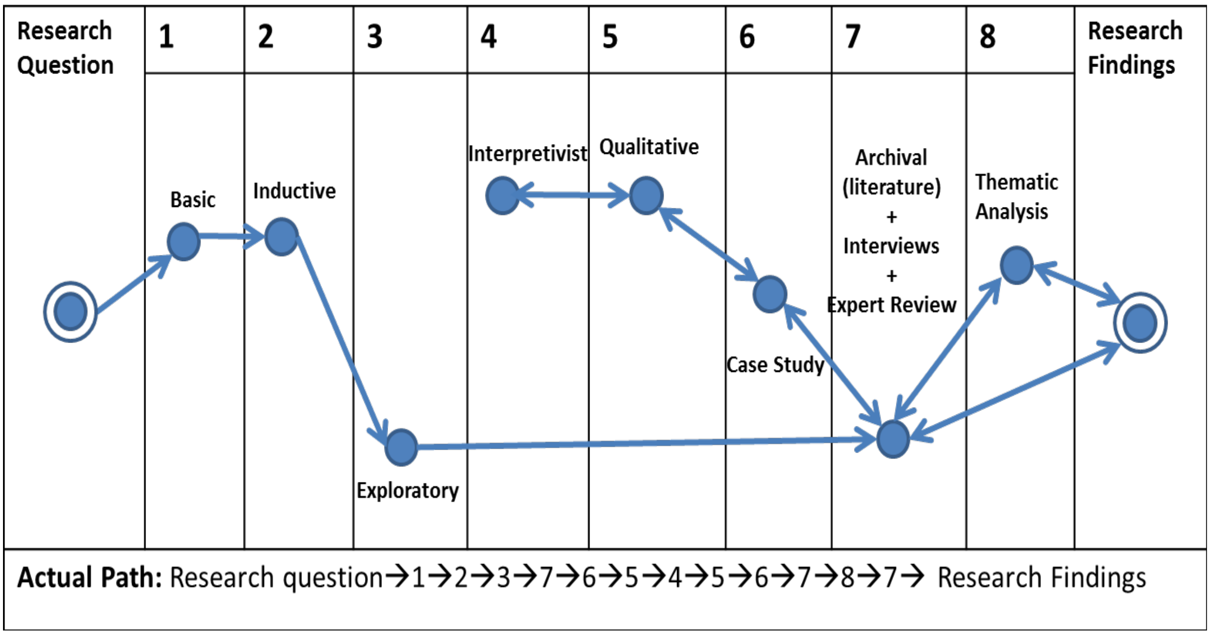

Figure 4. Research path for Example 2

Student B wants to conduct research on technical debt, as this phenomenon has not been addressed well in academic literature even though practitioners have emphasized its importance.

Once the research question(s) is defined, Student B decides that this research will be basic research (decision point 1), and will apply inductive logic (decision point 2). Student B believes that both academia and practitioners will benefit from having a framework that addresses what constitutes technical debt, why it occurs and its detrimental implications in projects. Practitioners will evaluate the findings of this research. Hence the purpose of this research is explanatory (decision point 3). Student B conducts a systematic literature (decision point 7), using thematic analysis (decision point 8) to identify to what extent technical debt is covered in academic literature. The outcome of the systematic literature is a framework that illustrates reasons and outcomes of technical debt. Then Student B carries out a multi-vocal literature review to revise the framework (decision point 7). The multi-vocal literature review considers the practitioner's writings whereby each document (i.e. practitioner's writing) is treated as a separate data point of a case study (decision point 6). Both the systematic literature review and the multi-vocal literature review apply a qualitative approach as research process (decision point 5), which will involve an interpretivist research approach (decision point 4), and thematic analysis as data analysis method (decision point 8). The framework is evaluated by interviewing several practitioners (decision point 7). The interviews are transcribed and thematic analysis is applied (decision point 8). Then, experts evaluate the framework (decision point 7) and reviews are integrated to research findings (decision point 8).

Figure 4 illustrates the path that this student took in this example. Further information on this research can be found in Tom et al. (2013). The "underlying factors" that influenced this student's research design were the student's interest in technical debt as well as her wish to bring a practitioners' view to her research (decision point 7).

Example 3

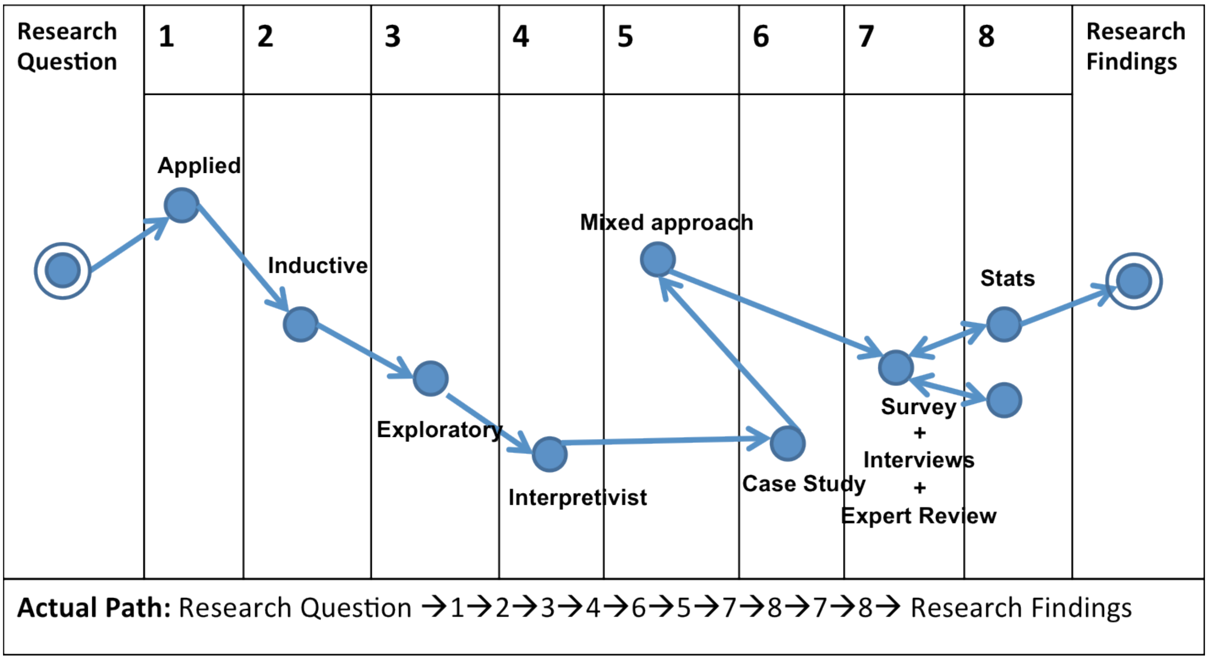

Figure 5. Research path for Example 3

Student C conducts research together with a company, and hence the research is applied (decision point 1). The research is focused on understanding company internal alignment in relation to software quality attributes. From the company point of view, it is perceived that the alignment is not optimal, for example the perception is that different roles (people) within the organisation prioritize different software quality attributes differently. Thus, the research question is formulated in relation to understanding, and hence the research is inductive (decision point 2). The objective is to explore potential differences in prioritization (decision point 3).

The exploration is done from the perspective of the people in the organization, and hence the research is conducted from an interpretivist research approach (decision point 4). It is decided that the research should be conducted as a case study (decision point 6). Eventually the plan is to conduct a series of case studies capturing different products and different development setups. It is decided to collect both qualitative and quantitative data (decision point 5). The qualitative data is in the form of interviews and an expert review (an expert reference group is assigned to the collaborative project between the company and academia), and the quantitative data is collected using of a questionnaire (survey tool) for prioritizing the different software quality attributes (decision point 7).

The quantitative data is analysed using statistical methods, and the interviews are transcribed and thematic analysis is applied (decision point 8). The outcomes of the analysis are discussed and evaluated by experts at several reference group meetings. Thus, the findings are iterated between data collection methods (decision point 7) and data analysis method (decision point 8).

Figure 5 illustrates the path that this student took in this example. Further information on this research can be found in Barney et al. (2013). The main “underlying factor” that influenced this student’s research design was the collaborative project with an industrial partner and their perceived challenges when it comes to alignment in relation to software quality attributes.

7. Discussion

This article introduces a research decision-making structure, provides a comprehensive overview of decisions that needs to be made during research design. The structure constitutes of several decision points that outlines decision space for a researcher. The article provides an introduction to each decision point and its constituents, as well as to the relationships between decisions points in the decision-making structure. These decisions points should not be viewed as totally different decision options; instead they represent different ends in a continuum. Typical scenarios of research in the three examples provided in the previous section can illustrate how these decision points combine into a research design.

Although the formulation of research questions is important in research design, including research strategy, research tactics and research implementation, there are some "underlying factors" that influence the research design e.g. previous experience, skills, knowledge about decision alternatives, motivation, and access to data resources. The three examples from the previous section show plausible research designs from published articles.

It is important to note that there is no one single path to investigate research questions/problems. Although having multiple paths in the decision space provides flexibility to the researcher during the research design, some paths are not feasible because not all combinations of decision options are valid. The research question(s) may evolve during the research and the researcher may need to adjust the research question(s) several times to fit them to the research methodology, research methods or their findings. The mapping of other research studies to the decision-making structure to ensure its validity beyond the examples provided is an area for future research, and it will be a direct consequence of the usage of the structure.

There are some decisions that are not addressed in this article such as deciding on unit of analysis, sampling, reliability and validity. In empirical research the decision on these topics are also important.

- Unit of analysis is the key object that is being analysed in empirical research. It can be an artefact, a project, a specific role of an individual, a group of people or an organization. Decision on unit of analysis is critical as it dictates the research question and tends to be embedded in the research methods.
- Sampling is the process of selecting which portion of the population that will be analysed. There are several sampling strategies that a researcher can adopt, such as random sampling, and snowballing. Researchers must have a good understanding about the sampling process before starting to collect their data.
- Reliability and validity have to be considered every step of the decision points in Figure 2 in a research design. It may include the design of a questionnaire or interviews, experimental design, sampling strategy or the way that the data will be analysed. Conducting empirical research without considering its reliability and validity is pointless, because the researcher will not be able to generalize from the results. The researcher must have knowledge about the potential threats to his/her study to be able to substantiate their findings.

As well as contributing to improving decision making in research design, this article contributes to empirical software engineering in other ways.

- The introduction of decision points and decision alternatives should help researchers to better appreciate the nature of decisions that they make during research design. We hope that researchers have a better understanding about their research design decisions and that the decision-making structure brings awareness and systematize the decisions that researchers have to go through during research design.
- Secondly, the research decision-making structure provides an important stimulus for advancement of empirical software engineering research. The objective is that the structure should help researchers to make informed decisions, better describe their research design and the reasons behind the selection of decision alternatives in each decision point.

The issues discussed in this article are by no means the only ones under discussion for the near and far future in empirical software engineering research. We hope that this article will stimulate further reflection and debate concerning how the quality of empirical software engineering research can be achieved and assessed.

8. Conclusions

This article has put forth a decision-making structure pertaining to the impacts of the research design decisions in empirical software engineering research. The discussion has been illustrated with three examples from the authors' experience. It has examined the structure of research design decisions that support researchers in doing their research design in empirical software engineering research.

The objective of this article is to highlight the implications of the research design decisions, raise some issues for consideration when conducting empirical research, for example that decision points are not independent; one decision may impact other decisions. Furthermore, it aims to help researchers (i) to better understand the interrelationships of the decision points in their research design. Awareness of the research design decisions will help researchers to design their research with less flaws/weaknesses and may facilitate informed choices; (ii) to be able to present their research design and research results with confidence. Awareness of the research design decisions will bring put the researcher in a more confident position when discussing their research results; and (iii) to be able to conform to research standards.

We hope that this article will lead to empirical software engineering research being better understood and assist researchers to articulate more carefully how they conduct their work, and how they organize and justify their research contribution. This article is one input to this continuing process and it may be useful not only to empirical software engineering researchers, but also to those working in the fields of information systems and computer science.

References

Adolp A., Hall W., Kruchten P. (2011) Using Grounded Theory to Study the Experience of Software Development. Journal of Empirical Software Engineering, 16(4):487-513.

Allison I., Merali Y. (2007) Software Process Improvement as Emergent Change: A Structurational Analysis. Information and Software Technology 49(6):668-681.

Barney S., Mohankumar V., Chatzipetrou P., Aurum A., Wohlin C., Angelis L. (2014) Software Quality across Borders: Three Case Studies on Company Internal Alignment. Information and Software Technology, 56(1):20-38.

Basili V. (1993) The Experimental Paradigm in Software Engineering. Proceedings of the International Workshop on Experimental Software Engineering Issues: Critical Assessment and Future Directions. Pages 3-12, Springer-Verlag, London, UK.

Baskerville R. (2008) What Design Science is Not. European Journal of Information Systems, 17:441–443.

Benbasat I., Goldstein D.K., Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. MIS Quarterly 11(3):369-386.

Bertelsen, O.W. (1997) Towards a Unified Field of SE Research and Practice, IEEE Software, November/December, pp 87-88

Bhattacherjee A. (2012), Social Science Research: Principles, Methods, and Practices. USF Open Access Textbooks Collection. Book 3.

Birks D.F., Fernandez W., Levina N., Nasirin S. (2013) Grounded Theory Method in Information Systems Research: Its Nature, Diversity and Opportunities. Guest Editorial. European Journal of Information Systems, 22:1-8.

Boell S., Cecez-Kecmanivic D. (2010) Literature Reviews and the Hermeneutic Circle. Australian Academic and Research Libraries, 41(2):129-144.

Braun V., Clarke V. (2006) Using Thematic Analysis in Psychology. Qualitative Research in Psychology, 3(2):77-101.

Brooke, C. (2002). What Does It Mean to be 'Critical' in IS Research? Journal of Information Technology, 17(2), 49-57.

Butler T. (1998) Towards a Hermeneutic Method for Interpretive Research in Information Systems. Journal of Information Technology, 13:285-300.

Carver J., Seaman C., Jeffery R. (2004) Using Qualitative Methods in Software Engineering. International Advanced School of Empirical Software Engineering (IASESE04), August 18, 2004, LA CA.

Cecez-Kecmanovic D. (2011): Doing Critical Information Systems Research - Arguments for a Critical Research Methodology. European Journal on Information Systems, 20(4): 440-455.

Checkland P. (1981) Systems Thinking, Systems Practice. John Wiley, UK.

Chen W.S., Hirschheim R. (2004) A Paradigmatic and Methodological Examination of Information Systems Research from 1991 to 2001. Information Systems Journal, 14(3):197-235.

Coleman G., O'Connor R. (2007a) Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software, 81:772-784.

Coleman G., O'Connor, R. (2007b). Using Grounded Theory to Understand Software Process Improvement: A Study of Irish Software Product Companies. Information and Software Technology, 49(6):654–667.

Collis J., Hussey R. (2009) Business Research. Palgrave MacMillan, UK.

Creswell J. (2013) Research Design: Qualitative, Quantitative and Mixed Methods Approach. Sage Publication.

Davison R.M., Martinsons M.G., Kock N. (2004) Principles of Canonical Action Research. Information Systems Journal, 14(1):65–86.

Dunne C. (2011) The Place of the Literature Review in Grounded Theory Research. International Journal of Social Research Methodology, 14(2):111-124.

Dybå T., Prikladnicki R., Rönkkö K., Seaman C.B., Sillito J. (2011) Qualitative Research in Software Engineering. Empirical Software Engineering 16(4):425-429.

Easterbrook S., Singer J., Storey M.A., Damian D. (2008) Selecting Empirical Methods for Software Engineering Research. In Guide to Advanced Empirical Software Engineering,Edited by Shull F., Singer J., Sjoberg D.I.K., pp. 285-311.

Eisenhardt K.M. (1989) Building Theories from Case Study Research. Academy of Management. The Academy of Management Review, 14(4):532-550.

Engel R.J., Schutt, R.K. (2010) The Practice of Research in Social Work. Sage Publication.

Ghanam Y., Maurer F., Abrahamsson P. (2012) Making a Leap to a Software Platform Strategy: Issues and Challenges. Information and Software Technology, 54(9):968-984.

Ghapanchi A.H. (2011) Dynamic Capabilities and Project Characteristics Contributing to the Success of Open Source Software Projects. PhD Dissertation Thesis. The University of New South Wales, Sydney, NSW Australia.

Gilbert G.N. (1995) Dagstuhl Seminar on Social Science Microsimulation: A Challenge to Computer Science, SchjoB, May 1-5, 1995.

Glaser B.G., Strauss A.L. (1967) The Discovery of Grounded Theory: Strategies for Qualitative Research. New York Aldine.

Glaser, B.G. (1992) Emergence vs. Forcing: Basics of Grounded Theory Analysts. California, Sociology Press.

Gregg D.G., Kulkarni U.R., Vinze A.S. (2001) Understanding the Philosophical Underpinnings of Software Engineering Research In Information Systems. Information Systems Frontiers 3(2):169-183.

Grix J. (2002) Introducing Students to the Generic Terminology of Social Research. Politics, 22(3):175-186.

Hannah D.R., Lautsch B.A. (2011) Counting in Qualitative Research: Why to Conduct it, When to Avoid it, and When to Closet it. Journal of Management Inquiry, 20(1):14-22.

Hansen S., Rennecker J. (2010) Getting on the Same Page: Collective Hermeneutics in a Systems Development Team. Information and Organization, 20(1):44-63.

Harrison R.J., Lin Z., Caroll G.R., Carley K.M. (2007) Simulation Modeling in Organization and Management Research. Academy of Management Review, 32(4):1229-1245.

Hevner A.R., March S.T., Park J., Ram S. (2004) Design Science In Information Systems Research. MIS Quarterly, 28(1):75-105.

Jafarzadeh H., Aurum A., D'Ambra J. (2011) Factors Affecting the Success of Businesses in Effective Utilization of Search Engine Advertising. ICIS 3-7 December, Shanghai, China, 2011, SIGIQ Workshop.

Jafarzadeh, H., Aurum, A., D'ambra J., Abedin, B. (2013) Determinant of Intention to use Search Engine Advertising: A Conceptual Model. International Journal of Enterprise Information Systems (IJEIS) 9 (3):22-38.

Johnson CF (1996) Deductive versus Inductive Reasoning: A Closer Look at Economics. The Social Science Journal, 33(3):287-299.

Johnson R.B., Onwuegbuzie A.J. (2004) Mixed Methods Research: A Research Paradigm Whose Time Has Come. Educational Researcher, 33(7):14-26.

Kachigan S.K., (1986) Statistical Analysis: An Interdisciplinary Introduction to Univariate & Multivariate Methods. Radius Press, New York.

Kitchenham, B.A., Pfleeger, S.A. (2002): Principles of Survey Research Part 2: Designing a Survey. ACM SIGSOFT Software Engineering Notes 27(1):18-20.

Kitchenham B.A., Pfleeger S.L., Pickard L.M., Jones P.W., Hoaglin D.C., El Emam K., Rosenberg J. (2002) Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering, 28(8):721-734.

Kitchenham B.A., Pickard L.M. and Pfleeger S.L. (1995) Case Studies for Method and Tool Evaluation, IEEE Software, 12(4):52–62.

Klein H.K., Myers M.D. (1999) A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly, 23(1):67-89.

Kline R.B. (2011) Principles and Practice of Structural Equation Modeling. Guilford Press, New York.

Kontio J., Lehtola L., Bragge J. (2004) Using the Focus Group Method in Software Engineering: Obtaining Practitioner and User Experience. Proceedings of International Symposium of Empirical Software Engineering, pp. 271-280.

R. Kyburz-Graber (2007) Does case-study methodology lack rigour? The Need for Quality Criteria for Sound Case-study Research, as Illustrated by a Recent Case in Secondary and Higher Education. Environmental Education Research, 10(1):53-65.

Law A.M. (2007) Simulation Modeling and Analysis, Volume 4. McGraw-Hill New York.

Lee A.S. (1989) A Scientific Methodology for MIS Case Studies. MIS Quarterly 13(1):33–54.

Lethbridge T.C., Sim S.E., Singer J. (2005) Studying Software Engineers: Data Collection Techniques for Software Field Studies. Journal of Empirical Software Engineering, 10(3):311–341.

Lyons,H. (2009) Case Study research methodology for publishing developments in ICT-facilitated learning in higher education – a perspective approach. Innovations in Education and Teaching International, 46(1): 27-39.

March S.T., Smith G.F. (1995) Design and Natural Science Research on Information Technology. Decision Support Systems, 15(4):251-266.

Mack N., Woodsong C., MacQueen K., Guest G., Namey E. (2005) Qualitative Research Methods: A Data Collector's Field Guide. Research Triangle Park, NC: Family Health International.

Marascuilo L.A., Serlin, R.C. (1988) Statistical Methods for the Social and Behavioral Sciences. W.H. Freeman and Company, New York.

McKernan J. (1996) Curriculum Action Research: A Handbook of Methods and Resources for the Reflective Practitioner. London, Kogan Page.

McLeod L., MacDonell S.G., Doolin B. (2011) Qualitative Research on Software Development: A Longitudinal Case Study Methodology. Journal of Empirical Software Engineering, 16(4):430-459.

Mingers J. (2001) Combining IS Research Methods: Towards a Pluralist Methodology. Information Systems Research, 12(3):240-259

Mkansi M., Acheampong E.A. (2012) Research Philosophy Debates and Classification: Students' Dilemma. Electronic Journal of Business Research Methods, 10(2): 132-140.

Moe N. B., Aurum A., Dybå T. (2012) Challenges of Shared Decision-making: A Multiple Case Study of Agile Software Development. Information & Software Technology 54(8):853-865.

Müller M., Pfahl D. (2008) Simulation Methods. In Guide to Advanced Empirical Software Engineering. Ed Shull F., Singer J., Sjøberg D.I.K. Springer-Verlag, Germany

Myers M.D. (1995) Dialectical Hermeneutics: A Theoretical Framework for the Implementation of Information Systems, Information Systems Journal, 5(1): 51-70.

Myers M.D. (1997) Qualitative Research in Information Systems. MIS Quarterly 21(2):241-242.

Myers, M. D., Klein, H. K. (2011). A Set of Principles for Conducting Critical Research in Information Systems. MIS Quarterly, 35(1):17-36.

Nunamaker J.F. Jr., Chen M., Purdin T.D.M. (1991) Systems Development in Information Systems Research. Journal of Management Information Systems, 7(3):89-106.

Orlikowski W.J., Baroudi J.J. (1991) Studying Information Technology in Organizations: Research Approaches and Assumptions. Information Systems Research 2(1):1-28.

Ostrowski L., Helfert M. (2011) Commonality in Various Design Science Methodologies. Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 317-320. ISBN 978-83-60810-39-2.

Perry C. (1998) Processes of a Case study Methodology for Postgraduate Research in Marketing. European Journal of Marketing 32.9/10: 785-802.

Perry D.E., Porter A.A., Votta L.G. (2000) Empirical Studies of Software Engineering: A Roadmap. In "The Future of Software Engineering", Anthony Finkelstein (Ed.), ACM Press 2000.

Pinsonneault A., Kraemer K. (1993) Survey Research Methodology in Management Information Systems: An Assessment. Journal of Management Information Systems 10(2):75-105.

Rossi P., Freeman H.F. (1993) Evaluation: A Systematic Approach. Sage Publication, USA.

Runeson P., Höst M. (2009) Guidelines for Conducting and Reporting Case Study Research in Software Engineering. Journal of Empirical Software Engineering, 14(2):131-164.

Runeson P., Höst M., Rainer A., Regnell B. (2012) Case Study Research in Software Engineering: Guidelines and Examples. John Wiley, USA.

Seaman C.B. (1999) Qualitative Methods in Empirical Studies of Software Engineering. IEEE Transactions on Software Engineering, 25(4):557-173.

Shanks G, Parr A (2003) Positivist, Single Case Study Research in Information Systems: A Critical Analysis. Proceedings of the 11th European Conference on Information Systems, 1760-1774.

Shaw M. (2002) What Makes Good Research in Software Engineering? International Journal of Software Tools for Technology Transfer, 2002, 4(1):1-7.

Shaw M. (2003) Writing Good Software Engineering Research Papers. Proceedings of 25th Int Conference on Software Engineering, pp. 726-736.

Shull F., Singer J., Sjøberg D.I.K. (2008) Guide to Advanced Empirical Software Engineering. Springer, Germany.

Shye, S (1988) Inductive and Deductive Reasoning: A Structural Reanalysis of Ability Tests. Journal of Applied Psychology, 73(2):308-311.

Siegel S., Castellan, N.J. Jr. (1988) Nonparametric Statistics for the Behavioral Sciences. Mcgraw-Hill Book Company, New York.

Sjøberg D.I.K., Dybå T., Jørgensen M. (2007) The Future of Empirical Methods in Software Engineering. IEEE Proceedings of Future of Software Engineering (FOSE).

Smite D., Wohlin C., Gorschek, T., Feldt R. (2010) Empirical Evidence in Global Software Engineering: A Systematic Review. Empirical Software Engineering: An International Journal, 15(1):91-118.

Smite D., Wohlin, C., Aurum, A., Jabangwe, R., Numminen, E. (2013) Offshore Insourcing in Software Development: Structuring the Decision-Making Process. Journal of Systems and Software, 86(4):1054-1067.

Staron M., Kuzniarz L., Wohlin, C. (2006) Empirical Assessment of Using Stereotypes to Improve Comprehension of UML Models: A Set of Experiments. Journal of Systems and Software, 79(5):727-742.

Susman G., Evered R. (1978) An Assessment of the Scientific Merits of Action Research. Administrative Science Quarterly, 23(4):582–603.

Strauss A.L., Corbin J.M. (1998) Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publication.

Tichy W.F., Lukowicz P., Prechelt L., Heinz E.A. (1995) Experimental Evaluation in Computer Science: A Quantitative Study. Journal of Systems and Software 28 (1), 9-18

Tom E., Aurum A., Vidgen R. (2013) An Exploration of Technical Debt. Journal of Systems and Software, 86(6): 1498-1516

Urquhart C. (2002) Regrounding Grounded Theory – Or Reinforcing Old Prejudices? A Brief Reply to

Bryant. Journal of Information Technology Theory and Application, 4(3):43-54.

Walker D., Myrick F. (2006) Grounded Theory: An Exploration of Process and Procedure. Qualitative Health Research, 16(4):547-559.

Wynekoop J.L., Russo N.L. (1997) Studying System Development Methodologies: An Examination of Research Methods. Information Systems Journal 7(1):47-65.

Wohlin C., Runeson P., Höst M., Ohlsson M.C., Regnell B., Wesslén A. (2012) Experimentation in Software Engineering, Springer, ISBN 978-3-642-29043-5.

Yin R.K. (2002) Case Study Research. Sage Publication.

Zelkowitz M.V., Wallace D. (1997) Experimental Validation in Software Engineering. Information and Software Technology, 31(5):23-31.