Chapter 3

Research Methods in Software Engineering

3.1 Introduction

Different streams of basic sciences have good explanations i.e. detailed guidance for researchers and simplified views for the stakeholders for their research strategies. While, Software Engineering (SE) is a field without too much historic background and well understood guidance since it is less than four decades old. Software engineering researchers rarely write explicitly about their paradigms of research and their standards for judging quality of results. In general, software engineering researchers seek better ways to develop and evaluate software. They are motivated by practical problems and key objectives of the research are often *quality*, *cost* and *timeliness* of software products. Many unfruitful attempts had been made to characterize software engineering research. The data suggested by Redwine-Riddle (37, 38) recommends that around 10 years of the 15–20 years of evolution are spent in concept formation, development and extension. As a result, full understanding of research strategy must account for the accumulation of evidence over times as well as for the form and content of individual projects and research papers. Software engineering will benefit from a better understanding of the research strategies that have been most successful. The model of software engineering research reflects the character of the discipline. It identifies the *types of questions* software engineers find interesting, the *types of results* we produce in answering those questions, and the *types of evidence* that we use to evaluate the results.

3.2 What Research IS and What NOT

3.2.1 What research IS?

Research is a general term which covers all kinds of studies designed to find responses to worthwhile questions by means of a systematic and scientific approach. Research could include *synthesis* and *analysis* of previous research to the extent that it leads to new and creative outcomes. Thus research is:
- *a process of investigation* i.e. a systematic inquiry that investigates hypotheses, suggests new interpretations of data or texts and poses new questions for future research to explore
- *an examination* of a subject from different points of view. It is getting to know a subject by reading up on it, reflecting, playing with the ideas, choosing the areas that interest you and following up on them
- *a hunt for the truth* i.e. a systematic inquiry to describe, explain, predict and control the observed phenomenon involving both inductive and deductive methods (39)
- *to search purposely and methodically* for new knowledge and practical solutions in the form of answers to questions formulated beforehand
- *the way you educate yourself*

Figure 3.1: Research (Analogy)

Research is defined as the creation of new knowledge and/or the use of existing knowledge in a new and creative way so as to generate new concepts, methodologies and understandings. This definition of research is consistent with a broad notion of research and experimental development as comprising of creative work undertaken on a systematic basis in order to increase the stock of knowledge, including knowledge of humanity, culture and society, and the use of this stock of knowledge to devise new applications (40).
DICTIONARY DEFINITION: RESEARCH

- *To search or investigate* exhaustively
- *Studious inquiry or examination*; especially investigation or experimentation aimed at the discovery and interpretation of facts, revision of accepted theories or laws in the light of new facts, or practical application of such new or revised theories or laws
- The *collecting of information* about a particular subject.

3.2.2 What research IS NOT?

There exist several misconception about research. In this section we justify some these misconceptions i.e. what are NOT research but often misunderstood as to be research.

- **Research Isn't Teaching:** Teaching itself is generally regarded as the synthesis and transfer of existing knowledge. Generally, the knowledge has to exist before you can teach it. Most of the time, you aren't creating new knowledge as you teach. Some lecturers may find that their students create strange new knowledge in their assignments, but making stuff up doesn't count as research either.
- **Research Isn't Scholarship:** A literature search is an important aspect of the research process but it isn't research in and of itself. The process of being a scholar generally describes surveying existing knowledge. One might be looking for new results that he/she hadn't read before, or one might be synthesizing the information for his/her teaching practice. Either way, one aren't creating new knowledge, but are reviewing what already exists.
- **Research Isn't Encyclopedic:** Encyclopedias, by and large seek to present a synthesis of existing knowledge. Collecting and publishing existing knowledge isn't research, as it doesn't create new knowledge.
- **Research Isn't Just Data Gathering:** Data gathering is a vital part of research, but it doesn't lead to new knowledge without some analysis and some further work. Just collecting the data doesn't count, unless one does something else with it.
- **Research Isn't Just About Methodology:** Just because one are using mice, or interviewing people, or using a High Performance Liquid Chromatograph (HPLC) doesn't mean one are doing research. One might be, if one are using a new data set or using the method in a new way or testing a new hypothesis. However, if one

is using the same method, on the same data, exploring the same question, then one will almost certainly get the same results, and that is repetition, not research.

- **Research Isn't Repetition, Except in Some Special Circumstances:** If one is doing the same thing that someone else has already done, then generally that isn't research unless he/she are specifically trying to prove or disprove their work. What's the difference? Repeating an experiment from 1400 isn't research. One know what the result will be before he/she start it implies it has already been verified many times before. Repeating an experiment reported earlier probably is research because the original result can't be relied upon until it is verified.

3.2.3 Science Vs. Engineering

3.2.3.1 Science and Engineering

Engineering problems deal with the creation of new artifacts and *scientific problems* deal with the study of existing ones.
Scientific research :

- is about the study phenomena and try to find the truth
- what kinds of questions are interesting?
- what kinds of results help to answer these questions, and what research methods can produce these results?
- what kinds of evidence can demonstrate the validity of a result, and how to distinguish good results from bad ones?

Engineering research :

- is about the study methods, tools, etc. that can be used to solve practical problems
- may include invention of new methods, tools, etc. or improvement of existing ones, but invention is neither necessary nor sufficient

Thus, scientific and engineering research fields can be characterized by identifying what they value:

- kinds of questions are interesting
- kinds of results to answer the interested questions

- kind of research methods that can produce these results
- kinds of evidence can demonstrate the validity of a result
- way of distinguishing good and bad results

3.2.3.2 Difference between the Objectives of Science and Engineering Study

Science pays attention to natural aspects while engineering is concerned with artificial aspects. While science deals with the study of what things are like, engineering is concerned with what they should be like in order to make it possible to construct new objects. While sciences deal with the study of existing objects and phenomena, be it physically, metaphysically or conceptually, but engineering is based on how to do things, how to create new objects. The difference between science and engineering lies in the modes of knowledge and action that they develop, not in one of them knowing and the other applying (41). Nevertheless, both science and engineering are knowledge and action since in the same way science is also action, not just knowledge; correlatively, we may say that engineering is also knowledge and not just application. In other words, science and engineering will differ in the research process which is used in each one (42). An engineering problem becomes a scientific problem once the object has been created; when a new artifact is created for example a new model, by an engineering research process, this new artifact becomes an existing one being an object of study by a scientific research process for example, studying its correctness, its quality, etc.

3.2.3.3 Software Engineering : Science or Engineering?

There are several non-conclusive opinions by several authors (43, 44, 45) regarding the issues whether software engineering can be a real engineering. To answer this question, we take as reference point other fields with a bigger historic background and also with more maturity such as Electronics, Chemistry or Geology– in such fields may we talk as science as engineering. So, in the same way electronic physics and electronic engineering, chemistry and chemical engineering, geology and geological engineering coexist and making a simile with these fields, we can say that SE has a double nature of science and engineering, depending on the object of study; this fact determines the research process. The nature of Software Engineering (SE) research, basically science or engineering, depends on its object of study. Seen this way, and according to the object of study, the research process will be different so that the kinds of problems can be tackled by means of different research methods and even by means of different paradigms. A classification of the issues of the SE discipline in:

- *those with a scientific nature :* focuses on theoretical bases of the SE
- *those with an engineering nature :* deals with the problem of building new software artifacts.

We try to justify this hypothesis on the basis of the paradigms and the research process which is in general used for the resolution of these kinds of problems.
Therefore, using Blum terminology (46), software science comes from computer science and pays attention to the aspects that have to do with the study of built artifacts, like code or other kind of artifacts such as models, documents, etc. This science deals with problems such as algorithmic complexity, software metrics, testing techniques, etc. On the other hand, we can find other kind of engineering research problems, supported by software science, concerned with the creation of software artifacts and we could define it as the study of transforming ideas into operations (46).
Although there are other classifications of problems in SE, these classifications are not appropriate since they are not focused on the research process. Thus, for instance, the SE Body of Knowledge project (SWEBOK) (47) sets out areas as: Software design, Software construction, SE tools and methods, etc. This schema is accurate enough but it is mainly centered on the creation of a body of knowledge with educative aims. Then, it provides invalid areas of knowledge for a classification of research problems because we can find scientific and engineering problems in each area, for example in SE tools and methods area. It is not the same trying to create a new method than testing a previously created method. In Software design, you can either create new models to improve the design process or study existing models, and analyze their implantation and use in a company. We notice the former case is an engineering problem while the latter case is an empirical or socio-cultural problem but all of them are included in the same area of the SWEBOK. This inclusion into the same area makes its difference in the research field impossible.
Hence, it is well justified that software engineering is an engineering discipline.

3.3 Broader View of Software Engineering Research Paradigm

To carry out research in any field, it is important to set out a paradigm for any research to realize.

3.3.1 Types of Research Paradigms

In general, we may divide the research paradigm into the following two categories (48):

- **Descriptive Paradigms:** The descriptive paradigms is evaluative–deductive or positivist paradigm, valuative interpretive or interpretive paradigm, evaluative-critical or critical paradigm etc.
- **Formulative Paradigms:** The formulative paradigms are formulative model, formulative process, method, algorithm etc.

3.3.2 Types of Research Problem Domain

From our point of view, depending on the nature of problems, we may broadly classify the problem domains in to the following two types:

- **Scientific Problems Domain :** In scientific research problems, *evaluative* paradigms are the used in most cases. *Positivist paradigms* as used in empirical sciences and either *interpretive* or *constructive* paradigms as used in social and cultural problems. In this regard, for instance, we could apply positivist paradigms to testing or interpretive paradigms to organization processes which are necessary for the implantation of a tool. Although there are other paradigms, even combinations of paradigms which can give rise to mixed paradigms, they always present characteristics which allow us to include them in a behavioral science research paradigm (49).
- **Engineering Problems Domain :** To engineering research problems, *descriptive paradigms* are used and these paradigms interact with positivist and interpretive paradigms (50). Thus, for instance, by means of literature reviews the researcher can try to establish the weak spots of a model and its respective technique of creation and afterwards can try to establish a description of a new technique and the new built model.

Brooks (51) reflected on the tension in human computer interaction research between :

- "narrow truths" proved convincingly by statistically sound experiments that satisfy the gold standard of science, and
- "broad truths", generally applicable, but supported only by possibly unrepresentative observations that provide pragmatic guidance, but at risk of over generalization.

Brooks (51) proposes a certainty shell structure – to recognize *three* nested classes of results :

- **Findings:** well established scientific truths, judged by truthfulness and rigor
- **Observations:** reports on actual phenomena, judged by interestingness
- **Rules of thumb:** generalizations, signed by their author but perhaps incompletely supported by data, judged by usefulness with freshness as a criterion for all three.

This same problem is also in software engineering. Observations and rules of thumb provide valuable guidance for practice when findings are not available. They also help to understand the area and lay the groundwork for the research that will yield findings in due time.
Further, Newman (52) compared research in human computer interaction (HCI) to research in engineering. He characterized engineering practice, identified three main types of research contributions, and performed a preliminary survey of publications in five engineering fields. He found that over 90% of the contributions was of three kinds:

- EM-Enhanced analytical modeling techniques, based on relevant theory, that can be used to tell whether the design is practicable or to make performance predictions
- ES-Enhanced solutions that overcome otherwise insoluble aspects of problems or that are easier to analyze with existing modeling techniques
- ET-Enhanced tools and methods for applying analytical models and for building functional models or prototypes.

3.4 Prior Reflections on Software Engineering Research

In the 1950s there was certain research but it was covered of confusion and without any significant publication. This idea can be supported by means of the fact that its first publications and conferences were held in the late 1960s (53). Till the beginning of 1980s, the existence of computer science and software engineering were cohesive– almost non-distinguishable.

From early 1980s onwards the academic presence of software engineering begins to separate from computer science. This influenced a lot in SE research developing (48). This youth of the discipline of SE is resulted in an immaturity of this research field. This immaturity is verified because research which is carried out in this discipline has several deficiencies (54, 55, 56) like the lack of systematic rigorous method, the lack of the evident methods of validation, etc. We can say SE research still lacks suitable scientific precision.

Each scientific discipline has a certain object of study which distinguishes the process to be followed in the research. The problem of SE research basically takes root in the fact that it is not so evident which the object of study is or rather the problem is that there are several objects of study with different nature and so, there are also different research and validation processes. This problem is motivating a rising concern with research methods and the validation in SE (41, 48, 50, 57, 58).

In 1980, Mary Shaw (43) examined the relation of engineering disciplines to their underlying craft and technology and laid out expectations for an engineering discipline for software. In 1984–85, Redwine, Riddle, and others (37, 38) proposed a model for the way software engineering technology evolves from research ideas to widespread practice. More recently, software engineering researchers have criticized common practice in the field for failing to collect, analyze and report experimental measurements in research reports (54, 55, 56, 59). Redwine and Riddle (37) presented timelines for several software technologies as they progressed through these phases up until the mid 1980s. Mary Shaw (60) presented a similar analysis for the maturation of software architecture in the 1990s. In 2001, Mary Shaw (60) presented preliminary sketches of some of the successful paradigms for software engineering research drawing heavily on examples from software architecture. Redwine and Riddle (37, 38) reviewed a number of software technologies to see how they develop and propagate and found that it typically takes 15–20 years for a technology to evolve from concept formulation to the point where it's ready for popularization.

Software engineering research is targeted to improve the practice of software development, so research planning should make provisions for the transition. The IMPACT project (61) is tracing the path from research into practice. The objectives of the project include identifying the kinds of contributions that have substantial impact and the types of research that are successful.

3.5 Types of Software Engineering Research

3.5.1 Types of General Research

In general, research may be categorized in to different types as mentioned below:

- **Action Research :** Action research is a methodology that combines action and research to examine specific questions, issues or phenomena through observation and reflection, and deliberate intervention to improve practice.
- **Applied Research :** Applied research is research undertaken to solve practical problems rather than to acquire knowledge for knowledge sake.
- **Basic Research :** Basic research is experimental and theoretical work undertaken to acquire new knowledge without looking for long term benefits other than the advancement of knowledge.
- **Clinical Trials :** Clinical trials are research studies undertaken to determine better ways to prevent, screen for, diagnose or treat diseases.
- **Epidemiological Research :** Epidemiological research is concerned with the description of health and welfare in populations through the collection of data related to health and the frequency, distribution and determinants of disease in populations, with the aim of improving health.
- **Evaluation Research :** Evaluation research is research conducted to measure the effectiveness or performance of a program, concept or campaign in achieving its objectives.
- **Literature Review :** Literature review is a critical examination, summarization, interpretation or evaluation of existing literature in order to establish current knowledge on a subject.

- **Qualitative Review :** Qualitative research is research undertaken to gain insights concerning attitudes, beliefs, motivations and behaviors of individuals to explore a social or human problem and include methods such as focus groups, in-depth interviews, observation research and case studies.
- **Quantitative Review :** Quantitative research is research concerned with the measurement of attitudes, behaviors and perceptions and includes interviewing methods such as telephone, intercept, *door-to-door* interviews as well as self-completion methods such as mail outs and online surveys.
- **Service or Program Monitoring and Evaluation :** Service or program monitoring and evaluation involves collecting and analyzing a range of processes and outcome data in order to assess the performance of a service or program, and to determine if the intended or expected results have been achieved.

Typically our research falls under the category of applied research where we undertook to solve some issues related to software crisis following a typical process model proposed by us – named BRIDGE (62).

3.5.2 Types of Software Engineering Research

Software engineering (SE) research maybe one of the following types:

1. Method or means of development
2. Method for analysis or evaluation
3. Design, evaluation, or analysis of a particular instance
4. Generalization or characterization
5. Feasibility study or exploration

The *first two types of SE research* i.e. method or means of development and Method for analysis or evaluation, produce methods of development or of analysis that the authors investigated in one setting, but that can presumably be applied in other settings.
The *third type of SE research* i.e. Design, evaluation, or analysis of a particular instance, deals explicitly with some particular system, practice, design or other instance of a system or method; these may range from narratives about industrial practice to analytic comparisons of alternative designs. For this type of research the instance itself should

have some broad appeal — an evaluation of Java is more likely to be accepted than a simple evaluation of the toy language you developed last summer.
The *fourth type of SE research* i.e. generalizations or characterizations explicitly rise above the examples presented in the paper.
Finally, the *fifth type of SE research* i.e. feasibility study or exploration types of research deal with an issue in a completely new way are sometimes treated differently from papers that improve on prior art.
The most common kind software engineering research reports an improved method or means of developing software—-that is, of designing, implementing, evolving, maintaining, or otherwise operating on the software system itself. Also fairly common kinds of software engineering research are about methods for reasoning about software systems, principally analysis of correctness i.e. testing, verification and validation (63).

This research work, in particular, from the software engineering perspective falls under the first category i.e. method or means of development. We proposed a software development method of in the form of a typical process model - named BRIDGE (62).

3.6 Typical Phases of Research

Redwine and Riddle (37, 38) identify *six* typical phases of research:

- **Phase 1: Basic Research :** Investigate basic ideas and concepts, put initial structure on the problem, and frame critical research questions.
- **Phase 2: Concept Formulation :** Circulate ideas informally, develop a research community, converge on a compatible set of ideas, and publish solutions to specific sub-problems.
- **Phase 3: Development and Extension :** Make preliminary use of the technology, clarify underlying ideas, and generalize the approach.
- **Phase 4: Internal Enhancement and Exploration :** Extend approach to another domain, use technology for real problems, stabilize technologies, develop training materials, show value in results.
- **Phase 5: External Enhancement and Exploration :** Similar to internal, but involving a broader community of people who weren't developers, show substantial evidence of value and applicability.

- **Phase 6: Popularization :** Develop production quality, supported versions of the technology, commercialize and market technologies and expand user community.

3.7 Research Strategies

3.7.1 Creating Research Strategies

The spectrum of good research strategies includes experimental computer science (54, 55, 56, 59) and this spectrum is much broader than just experimental research. Of course, not all the combinations of question, result and validation make sense, but often many of such combinations do.

3.7.2 Building Good Research Results

This discussion on research results has focused on individual results as reported in conference and journal papers. Major results, however, gain credibility over time as successive papers provide incremental improvement of the result and progressively stronger credibility. Assessing the significance of software engineering results should be done in this larger context. As increments of progress appear, they offer assurance that continued investment in research will pay off. Thus initial reports in an area may be informal and qualitative, but it presents a persuasive case for exploratory research. But reports afterwards in this area present empirical case, and later formal models that justify larger investment. This pattern of growth is consistent with the Redwine-Riddle's (37, 38) model of technology maturation.

3.8 Classifications of Research Design Strategies in Software Engineering

Research design strategies in software engineering may be classified into the following categories:

- Empirical
- Observational
- Correlational
- Quasi Experimental

The above research design strategies are discussed in the following section in brief.

3.8.1 Empirical research methods in software engineering

3.8.1.1 Controlled Experiments – high level of control and repeatable

A true experiment is defined as an experiment conducted where an effort is made to impose control over all other variables except the one under study. In controlled experiments, researchers control independent variable(s) and observe dependent variables(s). It is often easier to impose this sort of control in a laboratory setting. Thus, true experiments have often been erroneously identified as laboratory studies. This type of research design are useful for studying isolated activities in controlled environments i.e. comparing the use of two programming languages using students as subjects. These types of research are also called as *research in-the-small*.

3.8.1.2 Surveys – Sampling and Statistically Valid

In survey method research, participants answer questions administered through interviews or questionnaires. After participants answer the questions, researchers describe the responses given. In order for the survey to be both reliable and valid it is important that the questions are constructed properly. Another consideration when designing questions is whether to include open-ended, closed-ended, partially open-ended, or rating-scale questions (64). *Open-ended questions* allow for a greater variety of responses from participants but are difficult to analyze statistically because the data must be coded or reduced in some manner. *Closed-ended* questions are easy to analyze statistically, but they seriously limit the responses that participants can give. Many researchers prefer to use a *partiall open-ended or Likert-type* scale because it's very easy to analyze statistically. In this method, researchers observe a phenomenon in a representative subset of some population i.e. collecting data about a number of projects in different organization through questionnaires and/or interviews. This types of research is also called *research in-the-large* (61).

3.8.1.3 Case Studies

Case study research involves an in-depth study of an individual or group of individuals. Case studies often lead to testable hypotheses and allow us to study rare phenomena. Case studies should not be used to determine cause and effect, and they have limited use for making accurate predictions. Case studies needs careful documentation, multiple case studies can strengthen results. But, there are two serious problems with case studies i.e. expectancy effects and atypical individuals. *Expectancy effects* include the experimenters'underlying biases that might affect the actions taken while conducting research. These biases can lead to misrepresenting participants descriptions. Describing *atypical individuals* may lead to poor generalizations and detract from external validity. In case study, researchers observe phenomena in a *real-life context* i.e. collecting data about an ongoing project. Possible methods for data collection include interviews, project participation, documents, artifacts (software). This type of research is also called *research in-the-typical* (61).

3.8.2 Observational Method/Field Observation

The primary characteristic of each of observational method (61) is that phenomena are being observed and recorded. A detailed report with analysis would be written and reported constituting the study of this individual case. These studies may also be qualitative in nature or include qualitative components in the research. There are two main categories of observations i.e. Quantitative and Qualitative. *Quantitative observational method* is generally based on measurement. While, *qualitative observational method* is generally based on no measurement.

There are two main categories of the observational method i.e. naturalistic observation and laboratory observation. The biggest advantage of the naturalistic method of research is that researchers view participants in their natural environments. Proponents of laboratory observation often suggest that due to more control in the laboratory, the results found when using laboratory observation are more meaningful than those obtained with naturalistic observation. Laboratory observations are usually less time consuming and cheaper than naturalistic observations. Of course, both naturalistic and laboratory observation are important in regard to the advancement of scientific knowledge.

When the science has an empirical nature, quantitative research methods can be applied (65); these methods try to solve problems like: *what model method is more efficient?* But, when the science has a social and cultural nature, qualitative research methods can be applied (66) and these methods can seek to answer questions like: *what factors make a given software process unacceptable to the company?* or *why is one information systems development tool more acceptable than another?*

3.8.3 Correlational Research

In general, correlational research examines the co-variation of two or more variables. Correlational research can be accomplished by a variety of techniques which include the collection of empirical data. Often, correlational research is considered type of observational research as nothing is manipulated by the experimenter or individual conducting the research. It is important to note that correlational research is not causal research. In other words, we cannot make statements concerning cause and effect on the basis of this type of research. Correlational research is often conducted as exploratory or beginning research. Once variables have been identified and defined, experiments are conductible.

3.8.4 Quasi-Experimental

Quasi-experiments are very similar to true experiments but use naturally formed or pre-existing groups. Therefore, this cannot be a true experiment. When one has naturally formed groups, the variable under study is a subject variable as opposed to an independent variable. As such, it also limits the conclusions we can draw from such a research study. There are many differences between the groups that we cannot control and those could account for differences in our dependent measures. Thus, we must be careful concerning making statement of causality with quasi-experimental designs. However, there are also instances when a researcher designs a study as a traditional experiment only to discover that random assignment to groups is restricted by outside factors. The researcher is forced to divide groups according to some pre-existing criteria. The results are again restricted due to the quasi-correlational nature of the study. As the study has pre-existing groups, there may be other differences between those groups than just the presence or absence of a wellness program. Hence, quasi-experiments may result from either studying naturally formed groups or use of pre-existing groups. When the study includes naturally formed groups, the variable under study is a subject variable. When a study uses pre-existing groups that are not naturally formed, the variable that is manipulated between the two groups is an independent variable. As no random assignment exists in a quasi-experiment, no causal statements can be made based on the results of the study.

3.9 Software Engineering Research Model: Questions, Results and Validation

Over several years a model was evolved, that explains software engineering research papers by classifying:

- the types of research questions they ask
- the types of results they produce, and
- the character of the validation they provide

3.9.1 Software Engineering Research Questions and Types

Generally speaking, software engineering researchers seek better ways to develop and evaluate software. *Development* includes all the synthetic activities that involve creating and modifying the software, including the code, design documents, documentation, etc. *Evaluation* includes all the analytic activities associated with predicting, determining, and estimating properties of the software systems including both functionality and extra-functional properties such as performance or reliability.
Software engineering research answers questions (63):
a. About method or means of development
- how can we do/create (or automate doing) X?
- what is a better way to do/create X?
b. About method for analysis
- how can I evaluate the quality/correctness of X?
- how do I choose between X and Y?
c. About details of design, evaluation, or analysis of a particular instance
- what is a (better) design or implementation for application X?
- what is property X of artifact/method Y?
- how does X compare to Y?
- What is the current state of X / practice of Y?
d. About Generalization or characterization over whole class of systems or techniques
- given X, what will Y (necessarily) be?
- what, exactly, do we mean by X?
- what are the important characteristics of X?

- what is a good formal/empirical model for X?
- what are the varieties of X, how are they related?

e. About exploratory issues concerning existence or Feasibility

- feasibility Does X even exist, and if so what is it like?
- is it possible to accomplish X at all?

In this work, we answer question about method or means of development.

3.9.2 Software Engineering Research Results and Types

The tangible contributions of software engineering research may be procedures or techniques for development or analysis; they may be models that generalize from specific examples, or they may be specific tools, solutions, or results about particular systems. Software engineering research results may take one of the following form:

a. Procedure or Technique

- New or better way to do some task, such as design, implementation, measurement, evaluation, selection from alternatives,
- Techniques for implementation, representation, management, and analysis, but not advice or guidelines.

b. Qualitative or Descriptive model

- Structure or taxonomy for a problem area; architectural style, framework, or design pattern; informal domain analysis
- Well-grounded checklists, well-argued informal generalizations, guidance for integrating other results.

c. Empirical model

- Empirical predictive model based on observed data

d. Analytic model

- Structural model precise enough to support formal analysis or automatic manipulation

e. Notation or tool

- Formal language to support technique or model (should have a calculus, semantics, or other basis for computing or inference)
- Implemented tool that embodies a technique

f. Specific solution

- Solution to application problem that shows use of software engineering principles – may be design, rather than implementation
- Careful analysis of a system or its development

- Running system that embodies a result; it may be the carrier of the result, or its implementation may illustrate a principle that can be applied elsewhere

g. Answer or judgment

- Result of a specific analysis, evaluation, or comparison

h. Report

- Interesting observations, rules of thumb

The result may be a specific procedure or technique for software development or for analysis. It may be more general, capturing a number of specific results in a model; such models are of many degrees of precision and formality. Sometimes, the result is the solution to a specific problem or the outcome of a specific analysis. Finally, as Brooks observed (51), observations and rules of thumb may be good preliminary results.

By far the most common kind of software engineering research result reports a new procedure or technique for development or analysis either described in narration or embodied in a tool. Analytic and descriptive models are also common. The analytic models support predictive analysis, whereas descriptive models explain the structure of a problem area or expose important design decisions. Models of various degrees of precision and formality are also common, with better success rates for quantitative than for qualitative models. Tools and notations were well represented, usually as auxiliary results in combination with a procedure or technique. But, empirical models backed up by good statistics are uncommon.

Our research result took the form of a descriptive model.

3.9.3 Research Result Validation Techniques

A key concept relevant to a discussion of research methodology is that of validity. The validity is the question about the truthfulness of a study under consideration. There are four types of validity that can be discussed in relation to research and statistics. Thus, when discussing the validity of a study, one must be specific as to which type of validity is under discussion. Good research requires not only a result, but also clear and convincing evidence that the result is sound. This evidence should be based on experience or systematic analysis, not simply persuasive argument or textbook examples.

3.9.3.1 Foundations for Acceptance of Research Results

The acceptance of any research results depends heavily on:

- the process of obtaining the results

- analysis of the results themselves

Research yields new knowledge. This knowledge is expressed in the form of a particular result.

3.9.3.2 Types of Validity of Empirical Studies

There are four types of validity. Each type of validity has many threats which can pose a problem in a research study. For a comprehensive discussion of the four types of validity, the threats associated with each type of validity, and additional validity issues one may go through Cook and Campbell (67). Each of the four types of validity will be briefly defined and described below.

a. Statistical Conclusion Validity

According to Cook and Campbell (67), "statistical conclusion validity refers to inferences about whether it is reasonable to presume covariation given a specified alpha level and the obtained variances". Essentially, the question that is being asked is:
- are the variables under study related? or
- is variable A correlated (does it co-vary) with Variable B?

If a study has good statistical conclusion validity, we should be relatively certain that the answer to these questions is "yes". Examples of issues or problems that would threaten statistical conclusion validity would be random heterogeneity of the research subjects and small sample size.

b. Construct Validity

One is examining the issue of construct validity when one is asking the questions:
- am I really measuring the construct that I want to study? or
- Is my study confounded (Am I confusing constructs)?

Threats to construct validity include subject apprehension about being evaluated, hypothesis guessing on the part of subjects and bias introduced in a study by expectancies on the part of the experimenter.

c. Internal Validity

Once it has been determined that the two variables (A & B) are related, the next issue to be determined is one of causality. One is examining the issue of internal validity when one is asking the questions:
- does A cause B?

If a study is lacking internal validity, one cannot make cause and effect statements based on the research; the study would be descriptive but not causal. There are many potential threats to internal validity.

d. External Validity

External validity addresses the issue of being able to generalize the results of your study to other times, places, and persons. Therefore, one needs to ask the following questions to determine if a threat to the external validity exists:
- would I find these same results with a difference sample?,
- would I get these same results if I conducted my study in a different setting?, and
- would I get these same results if I had conducted this study in the past or if I redo this study in the future?

If I cannot answer "yes" to each of these questions, then the external validity of my study is threatened.

3.9.3.3 Software Engineering Research Validation Techniques

The most common kinds of validation are experience in actual use and systematic analysis. The other result validation methods applied in software engineering based on assertion, demonstration, examples, evaluation, persuasion, or offer no evidence at all.

- **Experience**
  My result has been used on real examples by someone other than me, and the evidence of its correctness / usefulness / effectiveness is:
  - ... narrative
  - ... data, usually statistical, on practice
  - ... comparison of this with similar results in actual use
- **Analysis**
  I have analyzed my result and find it satisfactory through:
  - ... rigorous derivation and proof
  - ... data on controlled use
  - ... experiment
- **Example**
  Here's an example of how it works on:

  - ... a toy example, perhaps motivated by reality
  - ... a system that I have been developing

- **Persuasion**
  If the original question was about feasibility, a working system, even without analysis, can be persuasive:
  - ... if you do it the following way ...
  - ... a system constructed like this would ...
  - ... this model seems reasonable

- **Evaluation**
  Given the stated criteria, my result:
  - ... adequately describes the phenomena of interest
  - ... accounts for the phenomena of interest
  - ... is able to predict ... because ..., or ... gives results that fit real data ...
  - Feasibility studies, pilot projects

- **Blatant assertion**
  In case of blatant assertion generally no serious attempt to evaluate result is taken.

3.10 Software Engineering Research: The Road-map

3.10.1 Software Engineering Research Methods

Software engineering research includes, but is not limited to, experimental research. Depending on the kind of problem to solve and the context of the problem, science or engineering, different research methods are used (68). Moreover, scientific research methods cannot always be applied to engineering research problems (69). Scientific research problems are similar to problems broached in traditional sciences and can have either an empirical or a cultural and social nature.
When the science has an empirical nature, quantitative research methods can be applied (65); and when the science has a social and cultural nature, qualitative research methods can be applied (66). In both, it is necessary certain knowledge of the reality: the object of study is an existing object in the world. Thus, this kind of problems use the research methods proposed by traditional sciences, as they study phenomena and objects of the

world regardless of how they were created. However, there is not any precise method to broach engineering research problems and the search for a method appropriate to this field is becoming a research field in its own right (48, 50, 57, 58, 66). The solution of problems purely concerning engineering requires methods of a different kind since in these cases it is directly possible to apply neither empirical methods nor methods which have to do with social and cultural component as the object of study does not yet exist (70). Furthermore, in the case of engineering, it is necessary a major component of creativity, which makes it difficult to draw up a universal method for solving problems within this field. For instance, *what research method would be valid for the specification of a new methodology for software development?* It would be necessary to study existing methodologies, reflecting on them to determine their advantages and disadvantages and proposing a new one, which, while retaining the advantages of the methodologies studied, would, as far as possible, lack their shortcomings. Arriving at a better final proposition would largely depend on the creativity and common sense applied to the construction of the new method. This method is applied in engineering consist in the formulation of experiences and the identification of the best practices (45). In 1993, Basili laid out experimental research paradigms appropriate for software engineering (71).

3.10.2 Critiques of Experimental Software Engineering

During 1995–88, Later, Tichy (54, 55) and colleagues criticized the lack of quantitative experimental validation reported in conference papers:
"Computer scientists publish relatively few papers with experimentally validated results... The low ratio of validated results appears to be a serious weakness in CS research... This weakness should be rectified (54)".
They classified 246 papers in computer science and, for comparison, 147 papers in two other disciplines, according to the type of contribution in the article. The majority of the papers produced design and modeling results. Then, they assess each papers'evaluation of its results on the basis of the fraction of the article's text devoted to evaluation. They found, that hypothesis testing was rare in all samples, that a large fraction (43%) of computer science design and modeling papers lacked any experimental evaluation, and that software engineering samples were worse than computer science in general (72).
Zelkowitz and Wallace (56, 59) built on Basili's description of experimental paradigms and evaluated over 600 computer science papers and over 100 papers from other disciplines published over a 10-year period. Again, they found that too many papers have no experimental validation or only informal validation, though they did notice some progress

over the 10-year period covered by their study. These critiques start from the premise that software engineering research should follow a classical experimental paradigm.