# A structured approach to evidence-based software engineering in empirical software engineering research for students


M. Danz, T. Gräf, C. Michel*

Advisor: Andrei Miclaus†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*danz@teco.edu,tobias.graef@student.kit.edu,michel@teco.edu
†miclaus@teco.edu

**Abstract.** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
TBD -> in the structured style we propose.

**Keywords:** TBD

# Table of Contents

A structured approach to evidence-based software engineering in empirical software engineering research for students . . . . . . . . . . . . . . . . . . . 1
*M. Danz, T. Gräf, C. Michel* Advisor: Andrei Miclaus†*

1 Introduction . . . . . . . . . . 5
2 Fundamental Principles . . . . . . . . . . 6
2.1 EBSE - Evidence-Based Software Engineering . . . . . . . . . . 6
2.2 Structured Abstract . . . . . . . . . . 7
3 Related Work . . . . . . . . . . 9
4 Research Process - Checklist . . . . . . . . . . 10
4.1 Test . . . . . . . . . . 10
4.2 Scientific Method . . . . . . . . . . 11
4.3 Formulating Research Question and Hypothesis . . . . . . . . . . 12
Research Question . . . . . . . . . . 12
Hypothesis . . . . . . . . . . 13
Objectives . . . . . . . . . . 14
4.4 Search for Existing Evidence . . . . . . . . . . 14
4.5 Designing, Conducting And Interpreting Experiment . . . . . . . . . . 16
4.6 Answer Question . . . . . . . . . . 16
4.7 Discussion . . . . . . . . . . 16
4.8 Evaluate Process . . . . . . . . . . 17
5 Briefing Form - Briefing Form . . . . . . . . . . 18
5.1 Research Question and Hypothesis . . . . . . . . . . 18
5.2 Experiment . . . . . . . . . . 19
Variables . . . . . . . . . . 19
Research Techniques . . . . . . . . . . 20
Statistical Results . . . . . . . . . . 20
5.3 Conclusion . . . . . . . . . . 20
6 Discussion . . . . . . . . . . 21

Final TODOs:

TODO: check for broken quotes
TODO: check figure references
TODO: check section references
TODO: fix ToC chapter
TODO: add appendix and reference to it
TODO: check for first term occurence with using italic text

# 1 Introduction

Our advisor observed that most software build in a bachelor's or master's thesis is poorly evaluated or not evaluated at all. Controlled studies are needed to attain valid measurements that allow comparison. Students often struggle with obtaining this *empirical evidence* (see table 1) leading to unusable results making a proper comparison difficult or not possible. Furthermore, large pieces of software are hard to evaluate because effects can be hard to isolate.
Therefore, the aim of this work is to create a supporting system for students helping to improve the quality (in particular the substantiality) of their studies in the domain of software engineering.

The final system intended is a database containing a digitalized collection of experiments. The system is meant to simplify searching, scanning and comparing experiments. Allowing to quickly find existing evidence that can be used for software design decisions. To populate that collection, a process to create evidence correctly and a compact representation of the results are needed. In this work, two documents are proposed: *Checklist* to guide students through the scientific process and *Briefing Form* for a compact and structure resume of experiments.

First we align the proposed process with related work in section 3. In section 2 *evidence-based software engineering* (EBSE) and structured abstracts are introduced. EBSE is the foundation of the process introduced later in section 4. Therefore, the idea behind EBSE is very fundamental for this paper. Structured abstracts are used as a base for the form introduced in section 5.

# 2 Fundamental Principles

This chapter introduces *evidence-based software engineering* (EBSE) as well as *structured abstracts*.

EBSE introduced in 2004 as an adoption of the evidence-based approach in medicine. Kitchenham et al. thought that software engineering could profit from it in a similar manner as medicine did [KDJ04]. Finding evidence involves browsing through a lot of publications. Brereton et al. TODO: ref found that the quality of abstracts is often not sufficient to decide whether a paper is relevant in a specific context. Reading the conclusion solves this issue but increases the time needed to identify relevant studies. Structured abstracts can be used to improve the quality of abstracts, which is particularly important in the context of EBSE.

## 2.1 EBSE - Evidence-Based Software Engineering

The aim of EBSE is "to improve decision making related to software development and maintenance by integrating current best evidence from research with practical experience and human values" [DKJ05]. Practising EBSE includes five steps:

1. Ask an answerable question.
2. Find the best evidence that answers that question.
3. Critically appraise this evidence.
4. Apply the evidence (and critical appraisal).
5. Evaluate the performance in previous steps.

Formulating the question precisely is important for the success of the process. The question should be formulated broad enough, so important studies are not missed, but must be precise enough to cope with the amount of studies (see section 4.3).

In medicine, researchers rely heavily on already published *systematic literature reviews* (SLR) to find relevant studies. SLRs try to identify and interpret all available literature regarding a specific research question [Kee07]. There are several organisations dedicated to conduct such reviews in medicine. The lack of this infrastructure makes applying the evidence-based approach in software engineering more difficult, but the number of existing SLRs increases steadily. For further reading on SLR, see Wohlin et al. [WRH+12].

It is important to check the quality of the identified studies, because being published is not a guarantee for absence of errors. Sometimes the integrity of results is compromised, because the research method has weaknesses or the researcher has vested interest connected to the outcome of the study.

Applying the evaluated evidence means integrating it with personal experience and other requirements. This step highly depends on the context and type of technology under evaluation.

At last the performance in previous steps is evaluated to improve applications of the EBSE process in the future.

Kitchenham et al. also identify two major problems inherent to software engineering:

1. The skill factor: Performing software engineering methods and techniques often require skilled practitioners. This prevents blinding and can therefore cause problems related to subject and experimenter bias.
2. The lifecycle issue: Prediction of behaviour of deployed technology is difficult and it is hard to isolate effects because of interaction with other methods and technologies.

Furthermore, they also state two approaches to reduce each of these effects [KDJ04].

### 2.2 Structured Abstract

In their guidelines for reporting experiments in software engineering Jedlitschka et al. [JCP08] propose the use of *structured abstracts*. They adopted the idea from medicine and psychology, where structured abstracts were introduced to increase the quality of abstracts. Structured abstracts guide the writer as well as the reader by using headings. Although a variety of different elements is used (see for example TODO: "Adoption of structured abstracts by general medical journals and format for a structured abstract "), the most common elements of structured abstracts are *Background/Context*, *Objective/Aim*, *Methods*, *Results* and *Conclusion/Discussion*. Jedlitschka et al. [JCP08] suggest the use of a sixth heading called *limitations*. This information is necessary to decide whether a result can be transfered to another context. Kitchenham et al. [KBO+08] include this information in the *conclusion* section.

The list and description of elements below closely follows the suggestion of Jedlitschka et al. [JCP08]:

1. *Background/Context*: Briefly explains the motivation for conducting the study and refers to previous research.
2. *Objective/Aim*: Describes the purpose of the study, including the object that is studied as well as focus and perspective. This part should cover the research question.
3. *Methods*: Sums up used research methods. For example experimental design, setting, participants and selection criteria, intervention and measurement and analyzing technique.
4. *Results*: The key findings are described here in form of numerical values. Do not include interpretations here. See section 5.2 (Statistical Results) for further details.
5. *Limitations*: Describes the scope of the study to point out the limits of generalization. This element might be incorporated in the *conclusion*-element.
6. *Conclusion*: Contains the interpretation of results and puts them into larger context.

There are several studies comparing structured and unstructured abstracts in the domain of software engineering with regard to completeness and clarity:

- Structured abstracts include more relevant information and are easier to read than conventional abstracts [BKC+07,BKC+08].
- Inexperienced authors are likely to produce clearer and more complete abstracts when using a structured form [BBK11] .
- On average structured abstracts are longer (limitations in conclusion is a good idea to prevent lengthy abstracts) and have better readability than unstructured abstracts [KBO+08].

These findings are consistent with the ones in other disciplines. It is important to mention that there are critics of structured abstracts that are supported by studies, but in general structured abstracts are considered advantageous [Har04,Har14].

A downside to structured abstracts is their length compared to unstructured ones. If the size of abstracts is limited, the abstract should still be in a structured form traditional elements should be prioritized: background (one sentence), objective, methods, result and conclusion [JCP08].

If it is not possible to structure an abstract (e.g. due to standard of journal or supervisor, length limitations) the elements of a structured abstracts should be contained in the unstructured abstract to make sure no important information is missing. See the common structure of the clearest abstracts as found by Shaw [Sha03]. Moreover the reader should be able to quickly identify each element by reading through the abstract.

For further information and examples of structured abstracts see the guide of C. Andrade [And11] and Kitchenham et al. [KDJ04].

## 3 Related Work

Rainer et al. [RHB06] released a paper about the use of EBSE by 15 undergraduate students. We have used observations listed in the paper to create design guidelines for the documents introduced in section 4 and 5 . There are seven main issues we tried to address:

1. "Students had problems constructing well-formulated EBSE questions."
2. "Students used limited criteria for identifying the best or better evidence [...]"
3. "Students used a very limited number of search terms."
4. "Students provided poor explanation in their reports of how their searches were conducted."
5. "Students varied in their use of the EBSE checklist."
6. "Some students critically appraised the technologies rather than the publications (evidence) on the technologies."
7. "But we also think that the kinds of problems students were tackling [...] are not the kinds of problems researchers commonly investigate."

**Table 1.** Issues with EBSE found by Rainer et al. [RHB06]

There are already databases that are meant to ease the access to studies: SEED [JR08] and the *Evidence Map* [EBS].
SEED is a community-driven online database that has been created during graduate courses. It lists summaries of studies in a common format and grouped by topics. The studies are added by researchers, whereas users create unstructured summaries, ratings and comparison grids. There only exists a prototype [SEE] that has not been updated since 2009.
The Evidence Map is primarily concerned with classifying systematic literature reviews (secondary studies, see sections 2.1 and 4.4), but also provides lists of primary and tertiary studies as well as information and guidelines about EBSE. A major disadvantage is the lack of study summaries. It has not been updated since 2012.

Rainer et al. [RB08, p. 7] introduced a flowchart to support the use of EBSE. The chart is rather fine grained and therefore might be unsuited for students novice in EBSE and scientific working in general.

## 4 Research Process - Checklist

TODO: Motivation

Rainer et al. found, that "[s]tudents varied in their use of the EBSE checklist" [RHB06] (see issue 5 in table 1). Therefore, Checklist is designed to support students during the process and with doing TODO: rephrase EBSE rather than cluttering the process even more.

The process of Checklist TODO: rephrase contains seven steps and one branch point: "Formulate question", "formulate hypothesis", "search for existing evidence", "experiment", "answer question", "discussion", "evaluate process", and "substantial evidence?". The whole graph can be seen in figure 1. And Checklist is appended in TODO: ref appendix checklist.

In the left column of the document is a flow chart of the proposed process depicted. For computer science students this should be a fast way to navigate through the process. To further assist navigation visually each process step has been assigned a color to.

In the right column additional information is given about the process steps. Each process step in Checklist contains a short description, some guidelines, and a checklist with acceptance criteria. Whereas the guidelines contain methods and tool on how to process the current step. And the acceptance criteria give students a hint on when the current step is completed and move on to the next step in the process.

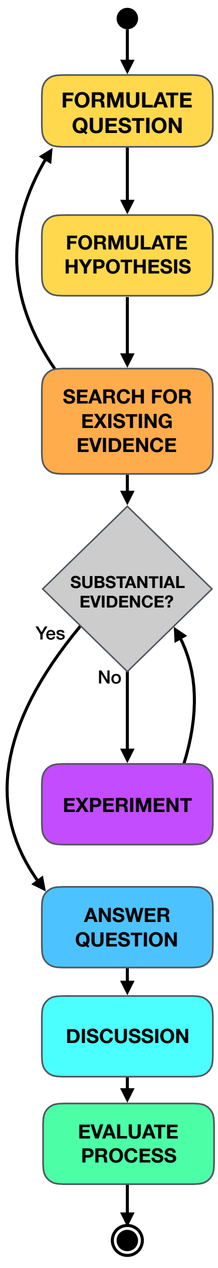


**Fig. 1.** Workflow Graph

### 4.1 Test

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

TODO: remove

### 4.2 Scientific Method

mapping of our process to scientific method
our states mapped to the states given in the graph of scientific method using color of our workflow graph. no additional inserts done.
the mapping shows, that we have more fine grained resolotion in the parts around experimenting. this foccus is motivated by the students as target group. since a advisor usually gives students a research question,our graph lacks the part with observations and thinking of questions and maps 3 states and two transitions to just the two yellow ones of the Checklist . We do not fully agree with refining, altering or expanding the hyptothesis based on experiment results. From our point of view, these findings should be reported and put in larger context like other findings and evoke a new supgraph with the new arising questions.

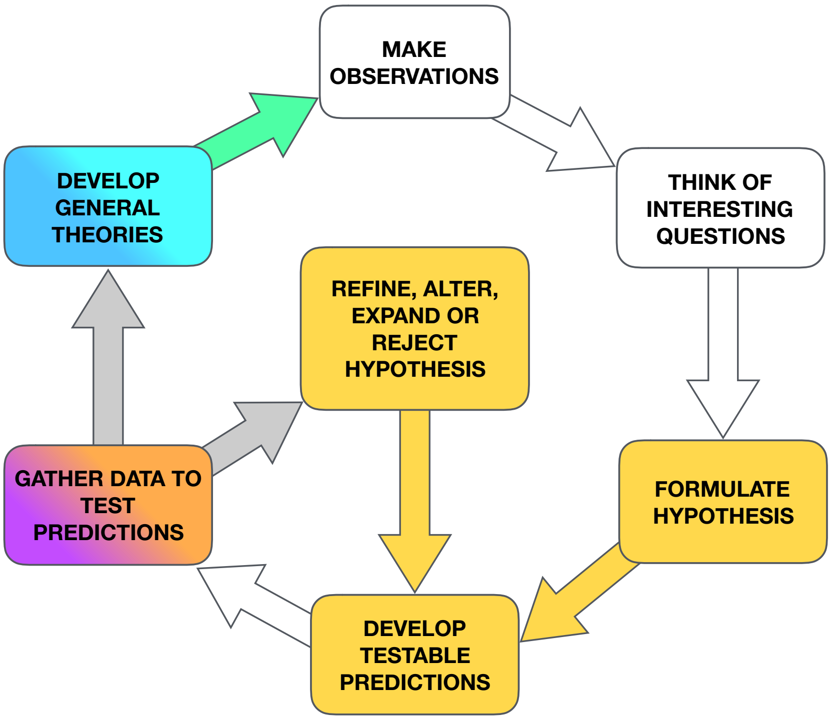


**Fig. 2.** Mapping of our process to the scientific method. TODO: source for graph

### 4.3 Formulating Research Question and Hypothesis

For researchers to produce relevant results and understand their research domain fully, the step of developing a good research question, with a supporting hypothesis and sometimes objectives is integral [FPFB09]. These components should be carefully designed *before* conducting the study that tries to answer the question. Otherwise it is more likely to produce questions that are already answered, or "could potentially lead to spuriously positive findings of association through chance alone" [FPFB09, p. 280]. This seems to be especially true for students [RHB06]. Therefore, we will address issue 1 from table 1 with this section. TODO: is this too blunt?

**Research Question** The question the later study is designed to answer is called research question [VO]. It should be an answerable question and address a relevant issue in the research area [DKJ05]. Preceding a research question is the need for a deep understanding of the topics that have already been studied, in order to produce questions which drive knowledge further. The questions that arise during the acquisition of knowledge, and cannot be answered by means of EBSE, are likely appropriate questions for further research [FPFB09].

There are two general classes of research questions: qualitative and quantitative questions. Qualitative research states questions which report, describe, or explore a subject [Cre14, p. 139-141]. The issues computer science students are confronted with are often of quantitative nature than of qualitative (e.g. Is databese engine X faster than database engine Y?). These problems are different from what researchers commonly investigate (see issue 7 in table 1) [RHB06]. Therefore, focus is on quantitative research questions in this paper. "Quantitative research questions inquire about the relationships among variables" [Cre14, p. 143], and from them emerge quantitative hypotheses.

To understand the structure of research questions Shaw provides a model where she categorizes research questions from software engineering papers in five types [Sha02] TODO: (maybe cut out).

To design a good research question Haynes coined the acronym PICO: Population, Intervention, Comparison group, and Outcome [Bri06]. Sometimes Time is added as fifth component, when it is important over what time frame the study is conducted, see left box in figure 3. A research question structured with the PICOT approach supports in restricting the research question and steers thereby hypotheses and study. By restricting the research question researchers can limit bias and increase the internal validity of the study, but a too narrow question may also lead to decreased external validity [FPFB09].

Before PICOT Sackett and colleagues suggested that good research questions consist out of three components: Intervention, Context and Outcome [Sac00], which is a more coarse grained decomposition than PICOT. Dybå et al. displayed a fitting example for this template in software engineering: "Does pair programming lead to improved code quality when practiced by professional software developers?" [DKJ05, p. 60] Here the intervention (technology) is pair programming, the context of interest are professional software developers, and the outcome (effect) is improved code quality [DKJ05]. To verify the quality of a freshly designed research question Hulley et al. suggest the use of the FINER criteria. It highlights key aspects of the question and provides thereby new angles to view the proposed study from. The FINER criteria consists of: Feasible, Interesting, Novel, Ethical, and Relevant [FPFB09]. A more detailed view of the FINER criteria can be seen in the right box of figure 3. TODO: specify more tips for writing a good question. Creswell2014)

| | |
|---|---|
| **P**opulation | What specific population are you interested in? |
| **I**ntervention (Technology) | What is the investigational technology / intervention? |
| **C**omparison Group | What is the main alternative / baseline to compare with the intervention |
| **O**utcome | What do you intend to accomplish, measure, improve or affect? |
| **T**ime | What is the appropriate follow-up time to assess outcome? |

| | |
|---|---|
| **F**easible | ‣ Adequate number of subjects<br>‣ Adequate technical expertise<br>‣ Affordable in time and money<br>‣ Manageable in scope |
| **I**nteresting | ‣ Getting the answer intrigues investigator, peers and community |
| **N**ovel | ‣ Confirms, refutes or extends previous findings |
| **E**thical | ‣ Amendable to a study that institutional review board will approve |
| **R**elevant | ‣ To scientific knowledge<br>‣ To clinical and health policy<br>‣ To future research |

**Fig. 3.** PICOT criteria adjusted to fit better in computer science research [FPFB09] and FINER criteria for a good research question [FPFB09].

**Hypothesis** For each quantitative research question there should be a hypothesis - an educated guess about the outcome of the research question [Bud10,FPFB09]. A good hypothesis needs to be a testable, prediction of the studies outcome, but it is important that it does not contain any interpretation [PRR01]. A simple template for writing a hypothesis would be:

> *If [I do X], then [Y] will happen.* [Bud10]

Vickers et al. propose a more refined structure, whereas a good hypothesis needs to include three components: Two or more variables, population/context, and the relationship between the variables [VO]. TODO: specify the thing with the variables more. For example a good hypothesis in software engineering research could be:

> Pair programming used by professional software developers improves code quality, in comparison to teams that use conventional techniques.TODO: revise this.

Furthermore, when conducting empirical research - as we propose in this paperTODO: revise this. - the hypothesis should be formulated as a *null hypothesis* $H_0$, and be accompanied by an *alternative hypothesis* $H_1$ [FPFB09]. The null hypothesis is a theory that is believed to be true but not proven jet. The alternative hypothesis is the opposite prediction of the null hypothesis [PRR01]. At the end of the study the null hypothesis is empirically tested, and only if it is rejected (i.e., there is a significant difference between groups) the alternative hypothesis is taken as true. This confirms that effects did not show by chance alone [FPFB09]. A null hypothesis to the example above would be:

> Pair programming used by professional software developers does not affect code quality. TODO: revise this.

To support the validity of the study even more, the hypotheses should be formulated as 2-sided hypothesis. "A 2-sided hypothesis states that there is a difference between [groups, but without specifying the direction of the outcome]."[FPFB09, p.280] 1-sided hypotheses should only be used when there is a strong justification for one direction of the outcome [FPFB09]. A 2-sided revision of the $H_1$ from above would be:

> Pair programming used by professional software developers does affect code quality. TODO: revise this.

TODO: specify more tips for writing a good hypothesis. Creswell2014

**Objectives** Sometimes researchers define objectives to their hypotheses. They are active statements that "define specific aims of the study and should be clearly stated" [FPFB09, p. 280] at the beginning of research. Objectives help to define the study (e.g. helping to calculate sample size). [FPFB09,VO] Although we do not include objectives in our Briefing Form we would like to mention them for reasons of completeness.

### 4.4 Search for Existing Evidence

After formulating a research question it is important to know which scientific results exist that help answer this question. Students normally perform an "ad-hoc" literature review that is prone to missing evidence and getting biased results (experimenter bias).

For that reason EBSE proposes the use of *systematic literature reviews* (SLR), a type of secondary study, to search for studies related to the research question. The aim of SLRs is to "identify, assess and combine the evidence from primary research studies using an explicit and rigorous method" [ZBT11]. Using this well-defined method might prevent biased results of the literature review, but requires more effort (both in time and skill) than traditional literature reviews. There exist guidelines [Kee07,Woh14,ZBT11] and reports of experiences with conducting SLRs [BKB+07] that provide the means necessary to conduct a SLR.

In any case published SLRs are very useful, even if conducting one is not feasible, as they provide a summary of multiple studies concerning one research question. Currently there are at least two projects concerned with making SLRs and finding relevant studies easier by collecting and indexing them (SEED [JR08] and and the Evidence Map [EBS])TODO: no brackets, but both have not been updated recently. TODO: more disadv?

Students voted for the SLR conducted during the EBSE process as one of the hardest steps [Kee07]. Therefore we suggest a less formal search process to save time and effort. But it should be noted, that the search should still be planned and structured to get the desired resultTODO: state it?. If it is possible conducting a SLR is always preferable.

Several guidelines from SLRs are still useful TODO: rephrase:

- Think about your search strings and engines beforehand and write them down. This also addresses issue 4 (see section 3).
- It is unlikely to find all relevant literature using only a single search engine [BKB+07]. TODO: GS as good starting point (publisher bias)?
- *Snowballing* is useful, both in a forward and backward manner [Woh14]. TODO: ref students using few keywords. keywords found in other papers can be used as search terms.

  *Backward*-snowballing refers to looking at the references of a publication an therefore going back in time. This can be done by using the "cited by" functionality of Google Scholar.

  *Forward*-snowballing means going forward in time by looking at the papers citing the current paper.
- The search strategy depends on the research question. If the research domain provides a vast amount of studies the search can TODO: should, must? be restricted, for example the exclusion of older studies [BKB+07].
- Even for more experienced students it might be necessary to refine the research question based on the search result, because their understanding of the research domain increases [BKB+07]. For example having only a few search result can be caused by research questions that are too narrow. Research Questions that are too general can lead to an abundance of search result that would take too long to examine TODO: or can't be handled?.
- Categorize your research question using classification systems like the ACM Computing Classification System and search within these categories. This is especially helpful if it is problematic to limit the scope with in the research area or there are uncertainties about wording and namingTODO: rephrase

TODO: issue 3?

TODO: mention? systematic mapping studies (broader, but not as deep as SLRs (deep: quantitative analysis and quality assessment)) "Systematic Mapping Studies in Software Engineering", Petersen et al. 2008

### 4.5 Designing, Conducting And Interpreting Experiment

To generate evidence, experiments are conducted. To keep the complexity of the study manageable and prevent side effects, it is recommended to answer only one question per experiment. If the question is too extensive for a single experiment, split up the question in sub-questions and recursively start this process with each sub-question.
For guidance and documentation through the experiment use Briefing Form introduced in section 5. Since experimenting is a very complex topic and can not be fully covered in this paper, see [WRH+12,TA13] for further reading.

### 4.6 Answer Question

Before you can answer a question the existing studies must be critically appraised. This means checking the study design, study quality, relevance for the research question, as well as consistency between different studies.

Rainer et al. found that "students use limited criteria for identifying the best or better evidence" [RHB06] (see issue 2 in table 1). They suggest sensitizing students for biases to solve this issue. We also suggest guiding the user through the process, because students had additional problems (see issue 6). TODO: rephrase

The *GRADE approach* [ABB+04] is a grading system for studies that goes beyond simple hierarchies of study types. It was introduced for medicine and has been used in the software engineering domain [Woh13,TD08]. It is a well-defined method and differentiates between the quality of evidence and the strength of recommendation. Additionally Figure 4 shows a checklist containing the important factors to consider when appraising a study and we suggest following it when appraising studies.TODO: rework this paragraph This could help easing issue 6 in table 1.

One important aspect of appraising studies is checking whether they might be biased. Jørgensen et al. [JDLS16] reported indications of research and publication bias being quite common in the domain of software engineering. Shepperd makes similar findings and gives a good and short overview of the problem [She15].

The hypothesis is accepted or rejected on the basis of the critical appraisal and the research question is answered accordingly.

TODO: - maybe reference to different types of bias

### 4.7 Discussion

In this step the whole study should be discussed, with respect to limitations and scope of generalization. The study should be put in larger context to the concerning field of research. Additionally future approaches and increments can be discussed here. For guidance in analyzing the own study it is recommended to use a checklist as proposed by Dybå et al. (see fig. 4).

Make sure, to only discuss the content of the study. For reflection on the process, see section 4.8. TODO: Text

**Study Appraisal Checklist**

1. Is there any vested interest?
   - Who sponsored the study?
   - Do the researchers have any vested interest in the results?
2. Is the evidence valid?
   - Was the study's design appropriate to answer the question?
   - How were the tasks, subjects, and setting selected?
   - What data was collected, and what were the methods for collecting the data?
   - Which methods of data analysis were used, and were they appropriate?
3. Is the evidence important?
   - What were the study's results?
   - Are the results credible, and, if so, how accurate are they?
   - What conclusions were drawn, and are they justified by the results?
   - Are the results of practical and statistical significance?
4. Can the evidence be used in practice?
   - Are the study's findings transferable to other industrial settings?
   - Did the study evaluate all the important outcome measures?
   - Does the study provide guidelines for practice based on the results?
   - Are the guidelines well described and easy to use?
   - Will the benefits of using the guidelines outweigh the costs?
5. Is the evidence in this study consistent with the evidence in other available studies?
   - Are there good reasons for any apparent inconsistencies?
   - Have the reasons for any disagreements been investigated?

**Fig. 4.** Checklist for critical appraisal of studies compiled by Dybå et al. [DKJ05].

### 4.8 Evaluate Process

The last step of the process proposed here is to reflect on the process itself. After discussing the study and finishing it in the step before (see section 4.7), Dybå et al. recommend to reflect on how well each step was performed and what could be improved in the next iteration. To support this they recommend using *after-action reviews* (AAR) and *postmortem analysis* (PMA) [DKJ05].

AAR is a short meeting of 10 to 20 minutes answering the four questions in figure 5, left box. This method can also be used at any other point the process when need is to reflect on or rethink the current situation[DKJ05].

PMA is similar to AAR but with a deeper insight. Therefore it lasts for several hours up to a full day. It answers the questions in figure 5, right box [DKJ05]. Here conclusions can be drawn to improve upcoming EBSE cycles and further research in general. This method can give insight issues like: which methods worked out as planned and which did not. Where was time lost and deadlines missed. Which technique proves to be used in upcoming studies.

After this step the whole process starts over again with the next research question.

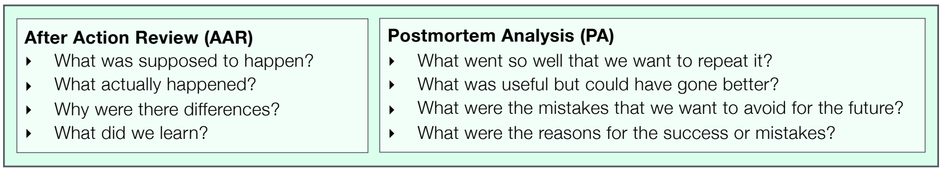


**Fig. 5.** After-action reviews (AAR) left box, and postmortem analysis (PMA) right box [DKJ05].

TODO: remove repetition in heading if we do not find better name for briefing form.

## 5 Briefing Form - Briefing Form

In this section, the *Briefing Form* is introduced. The Briefing Form is a one page sheet designed for two purposes: Guide users through the design of an experiment and make conducted experiments easier skimmable and searchable. It is similar to SEED but has a more detailed structure to support the search for existing evidence even more.

Experiments are used to obtain scientific evidence. To make the evidence as reliable as possible, the experiment needs to be designed and conducted with minimal flaws. That can be a very difficult task because experiments and their interpretation can be tremendously prone to errors or mistakes. Especially for people new to experimenting, an awareness for common mistakes and best practices can be very beneficial. By supporting the researcher to understand the study thoroughly, mistakes can be discovered early in the process.
On a small scale, the Briefing Form is also meant to be a supporting framework for a systematic workflow.

For software practitioners, it is important to find solutions for a problem quickly. Reading through papers can be very time consuming. The Briefing Form can help speed up the search by providing a clear structure for a quick overview for an experiment.

Since the Briefing Form is meant to implicitly guide the user's approach to experimenting and prevent mistakes, we tried to address problems observed by Rainer et al. [RHB06]. The issues and their respective conclusions are listed in Table 1. The Briefing Form consists of three parts: Research Question/Hypothesis, Experiment and Conclusion. These are explained in the following.

TODO: supports researcher in understanding the scope of research?

### 5.1 Research Question and Hypothesis

We decided to include both research question and hypothesis in the briefing form. Despite them being rather redundant in many cases. The research question is included due to the method researchers use to search for existing evidence, by entering their question into search engines. On the other side the hypothesis has its right to exist in the briefing form to support evaluation of validity and relevance of found evidence.

Therefore the fields in the form should be filled in with the research question and hypothesis constructed earlier (see section 4.3). The research question in a asking form which contains technology, context and effect. Or it is constructed according to the PICOT criteria (see fig. **??**). And the hypothesis in form of a testable prediction (e.g., “If [I do X], then [Y] will happen.” [Bud10]).

### 5.2 Experiment

To find cause and effect relationships between variables experiments are conducted [Bud]. An experiment consists of: Variables, techniques, and statistical results. These are integral components to describe a study in brief, and enhance the ability to search for, and evaluate the relevance and validity of a given study. For further reading see section 4.5.

**Variables** Variables are operationalizations of concepts, and there are usually three kinds of variables: Dependent, independent and controlled variables [Bud,Sel15]. A good variable must be measurable by any means [Bud]. Seltman additionally defines several qualities that make a good variable: “high reliability, absence of bias, low cost, practicability, objectivity, high acceptance, and high concept validity”[Sel15, p. 10].

TODO: include classification of statistical type. has influence on choice of statistical method in experiment.

We decided to include the different variables in the briefing form to make the decomposition of hypotheses easier while searching for related work. In order to make the search more fertile. Additionally the variables make it easier to evaluate the validity of the evidence.

*Independent Variables* The variables that are changed by the researcher during the experiment are called independent variables. In the pair programming example from section 4.3, under Hypothesis the independent variable is the technique utilized by the software developers (e.g., pair programming and conventional techniques). Unless there is justifiable evidence that two or more independent variables are not having an effect on the same dependent variable, it is advisable to only measure one independent variable during one experiment [Bud]. Otherwise it is not clear which of the independent variables caused the observed behavior of the measured dependent variable.

*Dependent Variables* An experiment focuses on the effect of independent variables on dependent variables. Therefore dependent variables are the variables that are measured. For example the code quality of a software project is the dependent variable used in the pair programming example from 4.3, under Hypothesis.

*Controlled Variables* The last role a variable can play during an experiment is the role of a controlled variable. These are variable which have or may have an effect on the dependent variables, but are not focus of the study. Therefore they are controlled by the researcher and kept constant throughout the experiment [Bud]. The test environment of a software, the gender of participants, or the number of trials per group are good examples for controlled variables in software engineering research.

**Research Techniques** For researchers that sift existing evidence it is sometimes important to know how or by what means a hypothesis was tested. Therefore asection for research techniques was included in Briefing Form . Here the conducting researcher gives here a brief resume about which techniques she uses. For example that the experiment is conducted as double blind experiment, using A/B Testing, with a Think aloud session and data is also measured via EEG (Electroencephalography). We use the term techniques here for everything ranging from methods that tell how an experiment is conducted, over measurement techniques, through to the hardware that is used. For further reading and more details on research techniques, measurement methods and study design see [AT08,WRH+12].

**Statistical Results** The outcome of an experiment is important for researchers who search for related work. Additionally the results help validating the solidity of evidence found. The statistic result of an experiment should never contain any interpretation. Interpretations should be given in a conclusion or discussion section separately from the data.

This field should at least contain the used statistic measurement, method, or model and their resulting parameters (e.g. Analysis of Variance: $F(2.57) = 211.496, p < .001$). It is advisable to use established methods to ensure comparability and reproducibility. More on statistical analysis can be found in books like [WRH+12], or [AT08] as already mentioned in Section 4.5.

### 5.3 Conclusion

This is the interesting part for searching researchers: What was found during the study? When searching for evidence researchers often like to see what question the study is about and what it finds. The conclusion indicates whether the question is answered or if more research has to be done on this issue. This includes an interpretation of the experiment results, and the verification or rejection of $H_0$ (and acceptance of $H_1$). Also a short statement on the scope of generalization should be given. Finally it must be clear if the given research question is answered and in which direction.

# 6 Discussion

TODO: TEXT!
aim to help students write more scientific papers/thesis
content to be validated/tested
propose a digital version for full potential
practitioner vs science, difficult for broad target audience

# References


[ABB+04] David Atkins, Dana Best, Peter A Briss, Martin Eccles, Yngve Falck-Ytter, Signe Flottorp, Gordon H Guyatt, Robin T Harbour, Margaret C Haugh, David Henry, Suzanne Hill, Roman Jaeschke, Gillian Leng, Alessandro Liberati, Nicola Magrini, James Mason, Philippa Middleton, Jacek Mrukowicz, Dianne O'Connell, Andrew D Oxman, Bob Phillips, Holger J Schünemann, Tessa Tan-Torres Edejer, Helena Varonen, Gunn E Vist, John W Williams, Stephanie Zaza, and GRADE Working Group. Grading quality of evidence and strength of recommendations. *BMJ (Clinical research ed.)*, 328(7454):1490, 2004.

[And11] Chittaranjan Andrade. How to write a good abstract for a scientific paper or conference presentation. *Indian Journal of Psychiatry*, 53(2):172, 2011.

[AT08] Bill Albert and Tom Tullis. Measuring the user experience. *Collecting, Analyzing, and Presenting Usability . . .*, pages 1–17, 2008.

[BBK11] David Budgen, Andy J. Burn, and Barbara Kitchenham. Reporting computing projects through structured abstracts: A quasi-experiment. *Empirical Software Engineering*, 16(2):244–277, 2011.

[BKB+07] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571–583, 2007.

[BKC+07] David Budgen, Barbara Kitchenham, Stuart Charters, Mark Turner, Pearl Brereton, and Stephen Linkman. Preliminary results of a study of the completeness and clarity of structured abstracts. *Proc. of the 11th Int. Conf. on Evaluation and Assessment in Software Engineering*, pages 64–72, 2007.

[BKC+08] David Budgen, Barbara A. Kitchenham, Stuart M. Charters, Mark Turner, Pearl Brereton, and Stephen G. Linkman. Presenting software engineering results using structured abstracts: A randomised experiment. *Empirical Software Engineering*, 13(4):435–468, 2008.

[Bri06] R. Brian Haynes. Forming research questions. *Journal of Clinical Epidemiology*, 59(9):881–886, 2006.

[Bud] Science Buddies. Variables in your science fair project. `http://www.sciencebuddies.org/science-fair-projects/project_variables.shtml`. accessed 2017-01-06.

[Bud10] Science Buddies. A Strong Hypothesis. `http://www.sciencebuddies.org/blog/2010/02/a-strong-hypothesis.php`, 2010. accessed 2017-01-02.

[Cre14] J W Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 2014.

[DKJ05] Tore Dybå, Barbara A. Kitchenham, and Magne Jørgensen. Evidence-based software engineering for practitioners. *IEEE Software*, 22(1):58–65, 2005.

[EBS] Evidence Based Software Engineering. `https://community.dur.ac.uk/ebse`. accessed 2017-02-25.

[FPFB09] Patricia Farrugia, Bradley A. Petrisor, Forough Farrokhyar, and Mohit Bhandari. Practical tips for surgical research: Research questions, hypotheses and objectives. *Canadian journal of surgery. Journal canadien de chirurgie*, 53(4):278–281, 2009.

[Har04] James Hartley. Current findings from research on structured abstracts. *Journal of the Medical Library Association*, 92(3):368, 2004.

[Har14] James Hartley. Current findings from research on structured abstracts: an update. *Journal of the Medical Library Association: JMLA*, 102(3):146, 2014.

[JCP08] Andreas Jedlitschka, Marcus Ciolkowski, and Dietmar Pfahl. Reporting experiments in software engineering. In *Guide to advanced empirical software engineering*, pages 201–228. Springer, 2008.

[JDLS16] Magne Jørgensen, Tore Dybå, Knut Liestøl, and Dag I K Sjøberg. Incorrect results in software engineering experiments: How to improve research practices. *Journal of Systems and Software*, 116:133–145, 2016.

[JR08] David S. Janzen and Jungwoo Ryoo. Seeds of Evidence: Integrating Evidence-Based Software Engineering. In *Software Engineering Education Conference, Proceedings*, pages 223–232, 2008.

[KBO+08] B. A. Kitchenham, O. Pearl Brereton, S. Owen, J. Butcher, and C. Jefferies. Length and readability of structured software engineering abstracts. *IET Software*, 2:37 – 45, 2008.

[KDJ04] Barbara A Kitchenham, Tore Dyba, and Magne Jørgensen. Evidence-based software engineering. In *Proceedings of the 26th international conference on software engineering*, pages 273–281. IEEE Computer Society, 2004.

[Kee07] Staffs Keele. Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. 2007.

[PRR01] Shalini Prasad, Ajith Rao, and Eeshoo Rehani. Developing hypothesis and research question. *500 Research Methods*, pages 1–30, 2001.

[RB08] Austen Rainer and Sarah Beecham. Supplementary Guidelines , Assessment Scheme and evidence-based evaluations of the use of Evidence Based Software Engineering, Version 2. (February 2008):1–27, 2008.

[RHB06] Austen Rainer, Tracy Hall, and Nathan Baddoo. A preliminary empirical investigation of the use of evidence based software engineering by undergraduate students. *10th International Conference on Evaluation and Assessment in Software Engineering (EASE 2006)*, 2006.

[Sac00] David Sackett. Evidence-based medicine : how to practice and teach EBM, 2000.

[SEE] SEED. `http://evidencebasedse.com`. 2017-2-25.

[Sel15] Howard Seltman. Experimental Design and Analysis. *Online Book*, page 428, 2015.

[Sha02] Mary Shaw. What makes good research in software engineering? *International Journal on Software Tools for Technology ...*, 4(1):1–7, 2002.

[Sha03] Mary Shaw. Writing good software engineering research papers: minitutorial. In *Proceedings of the 25th international conference on software engineering*, pages 726–736. IEEE Computer Society, 2003.

[She15] Martin Shepperd. How Do I Know Whether to Trust a Research Result? *IEEE Software*, 32(1):106–109, jan 2015.

[TA13] Tom Tullis and Bill Albert. *Measuring the user experience*. 2013.

[TD08] Dybå Tore and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10):833–859, 2008.

[VO] Peter Vickers and Maxine Offredy. Developing a Healthcare Research Proposal: An Interactive Student Guide. `http://www.health.herts.ac.uk/immunology/Web%20programme%20-%20Researchhealthprofessionals/hypothesisresearch_question.htm`. accessed 2017-01-06.

[Woh13] C Wohlin. An Evidence Profile for Software Engineering Research and Practice. *Perspectives on the Future of Software Engineering*, 2013.

[Woh14] Claes Wohlin. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. *18th International Conference on Evaluation and Assessment in Software Engineering (EASE 2014)*, pages 1–10, 2014.

[WRH+12] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*, volume 9783642290. 2012.

[ZBT11] He Zhang, Muhammad Ali Babar, and Paolo Tell. Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637, 2011.

TODO: append Briefing Form and Checklist