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
*\todo{Mails}student@student.kit.edu
†miclaus@teco.edu

Table of Contents

Abstract. TBD -> in the structured style we propose.

Keywords: TBD

1 Introduction

motivation

- observation: most software build in a bachelor or master thesis is poorly evaluated (if it is done at all)
- Initial aim: Find/create a method to evaluate and compare software developed by students.
- But: 1. Large pieces of software are hard to evaluate because effects can be hard to isolate.
- 2. To find out wich software is more suitable for a given problem, measurements are needed for a quantitative comparison that can be treated as empirical evidence.
- To attain valid measurements that allow comparison, a controlled study is needed. Students sometimes struggle with making that kind of controlled study TODO: tabellenverweis table 1 leading to unusable results making a proper comparison difficult or not possible at all.
- So we revised the question to: How 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 we intend is an electronical database containing a collection of experiments. The system is meant to simplify searching, scanning and comparison of 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. We propose two documents: *Checklist* to guide students through the scientific process and *Briefing Form* for a compact, structured and complete resume of the conducted experiment.

- experimentation/empiricism in CS?
- "should computer scientists experiment more?, Tichy 1997"
- Experimental evaluation in CS: A quantitative study, 1994: 40% of papers requiring quant evaluation have none (other disciplins: 15%)
- "A survey of Controlled Experiments in SE, 2005" from 1993 to 2002: 1,9% of software engineering paper report controlled experiments
- "Belief & Evidence in Empirical Software Engineering", 2016: programmers have strong beliefs, primarily formed by personal opinion (research papers just above "other", below peer opinion, mentor opinion and trade journals); beliefs vary between projects, but do not necessarily correspond with evidence in that project

– "Incorrect results in software engineering experiments: How to improve research practices", 2016: research and publication bias quite common, need to improve research practices (a key: avoid experiments with low statistical power)

2 Fundamental Principles

TODO: text

2.1 EBSE

TODO: short introduction to EBSE,...
In 2004 evidence-based software engineering (EBSE) was proposed as an adoption of the evidence-based approach in medicine [KDJ04]. 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 4.2).

In medicine researcher heavily rely 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 organisation dedicated to conduct such reviews in medicine. The lack of this infrastructure makes applying the evidence-based approach in software engineering more difficult. TODO: problems with SLRs? abstracts, papers should be written for synthesis,requirements for this, common mistakes/problems?

Kitchenham et al. [KDJ04] 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 subect and experimenter bias.
2. The lifecycle issue: Prediction of behaviour TODO: long time? of deployed technology is difficult and it is hard to isolate efffects because of interaction with other methods and technologies.

TODO: 2 approaches to reduce each of these effects in [KDJ04]

TODO: SLR in SE: lack of systematic reviews (still correct? source?), lack of replication studies (source?), problems regarding SLR TODO "Identifying relevant studies in software engineering", Zhang et al 2010

TODO: next paragraphs correct / useful?

The process of practising EBSE and the *Software Development Life Cycle* (SDLC) share characteristics. The SDLC starts with analysing a system and requirements that result in questions. These questions must be answered to continue the SDLC and design software that meets the requirements. EBSE can be used to improve this step. Decisions made using EBSE rely on evidence, decreasing the risk of making bad decisions (compare to the goal of EBSE).

maybe better: SDLC and EBSE are the same: questions regarding software are answered via prototypes (building software). EBSE reuses the created evidence to have shorter SDLC cycles (or fewer cycles for product of similar quality).

2.2 Structured Abstract

TODO: importance of abstracts: Abstracts, together with the title, are used to identify relevant research, not only in SLRs. Often the abstract is the only part of the paper that can be accessed for free. Therefore abstract and title should contain all necessary information to decide whether a paper (in case of SLR: primary study) is relevant in this context. (source: "Lessons from applying the systematic literature review process within the software engineering domain") → quality of abstract crucial for research, how to support researcher in writing useful abstracts? Structured Abstracts provide guidance for writer and reader. end TODO

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. Although a variety of different elements is used (see for example "Adoption of structured abstracts by general medical journals and format for a structured abstract "), the most common elements of structured abstracts are *Background* (*or Context*), *Objective* (or *Aim*), *Methods*, *Results* and *Conclusion* (or *Discussion*). Jedlitschka et al. [JCP08] suggest the use of a 6th (TODO: or sixth? heading called *limitations*. This information is necessary to decide whether a result can be transfered to another context. Others TODO: often? mostly? (e.g. 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** or **Context**: Explain briefly what the motivation for conducting the study was and refers to previous research.

2. **Objective** or **Aim**: Describes the purpose of the study, including the object that is studied as well as focus and perspective. TODO: research question or hypotheses?
3. **Methods**: Sums up which research methods were used, 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. TODO: name stat. significance, confidece interval, standard error? (Do not include interpretations here!).
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 context.

TODO: examples? keywords as part of abstract?

There are several studies comparing structured and unstructured abstracts in the domain of software engineering with regard to completeness and clarity TODO: noting the similar authors (weakening the statement)?:

- 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 found by Shaw [Sha03]). Moreover the reader should be able to quickly identify the elements by reading through the abstract.

TODO: notes and ideas

- Elements similar to “IMRAD”-structure (see paper “Adoption of structured abstracts by general medical journals and format for a structured abstract”)

– guidelines for construcing structured abstracts (from unstructered ones) in [KBO+08] → used to verify descr of elements
– Guide with examples (psychology): "How to write a good abstract for scientific paper.", C. Andrade
ANSI/NISO Z39.14.1997 (R2015) Guidelines for Abstracts
– use standard terminology (commonly used industry terms) [JCP08], no abbreviations [KBO+08]

end TODO

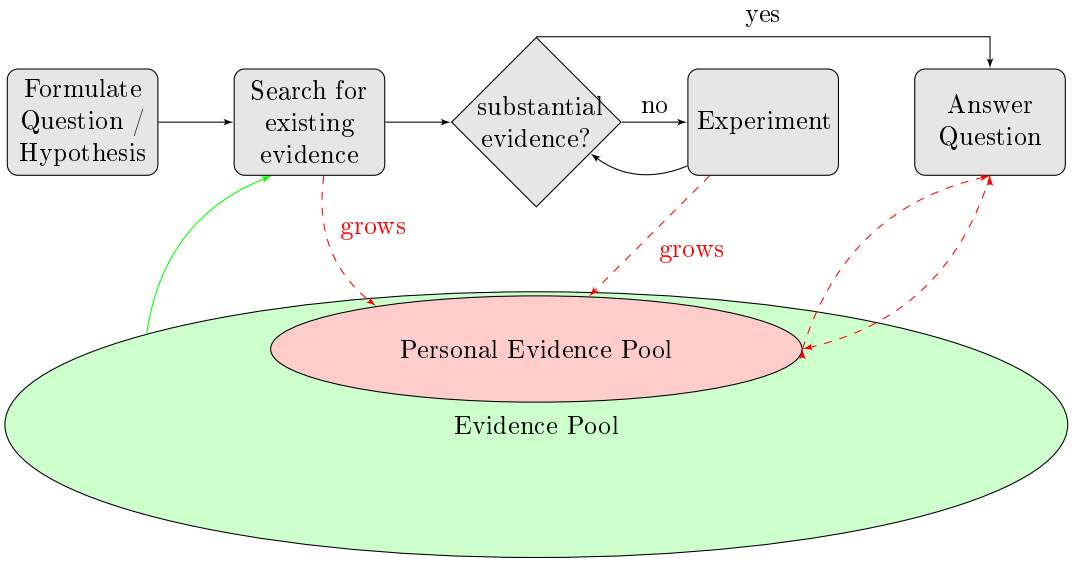
3 Related Work

TODO: SEED

Rainer et al. [RHB06] released a paper about the reported use of EBSE by 15 under-graduate students. We have used observations listed in the paper to create design guidelines for the documents introduced in section 4 and 5 . The main issues we tried to address are listed in Table 1.

4 Research Process - Checklist

TODO: Motivation



TODO:
Add numbers to each node, explain each node
Missing node: "Make Decision" at the end?
Layout/Style/Color

Observation	Conclusion/Guideline
"Students had problems constructing well-formulated EBSE questions." (p. 6)	Give examples for good questions to make sure the user understands a good question's scope of information. Also, explicitly list which building blocks should be contained in the question.
"Students used limited criteria for identifying the best or better evidence[...]" (p. 6)	Support decision-making to get a decision as unbiased and suited as possible. Since a decision's quality is highly dependent on the individual case, we only give a very general hint to the user. The idea is to sensitize the user to consciously prevent bias as good as possible.
"Students used a very limited number of search terms." (p. 6)	If users look for something very specific without knowing the technical term, search engines might yield better results when used with more detailed search terms. Also, synonyms or similar words might widen the search's scope to find more related work. Encourage more search terms by providing examples containing enough search terms.
"Students provided poor explanation in their reports of how their searches were conducted." (p. 7)	Recommend users to write down their search terms for a more structured search approach.
"Students varied in their use of the EBSE checklist." (p. 7)	Design the checklist in a way to support the user's workflow instead of hindering it. Keep it as simple as possible and provide enough examples to make the user never guess an item's meaning.
"Some students critically appraise the technologies rather than the publications (evidence) on the technologies" (p. 7)	TODO: link ciritcal appraisal checklist
"But we also think that the kinds of problems students were tackling [...] are not the kinds of problems researchers commonly investigate." (p. 8)	Scientific and practical evidence can have very different requirements regarding content and other aspects such as duration of evaluation. Due to the large differences, we kept the forms and checklists rather general to make the useful for both sides.

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

4.1 Scientific Method

TODO: map our checklist cycle to 'scientifc method' graph; text

4.2 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]

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]. In computer science as the research field matures these questions become more and more rare TODO: (find source). 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 Table 2. 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 Table 3. TODO: specify more tips for writing a good question. Creswell2014)

P	Population	What specific population are you interested in?
I	Intervention (technology)	What is the investigational technology/ intervention?
C	Comparison group	What is the main alternative/baseline to compare with the intervention
O	Outcome	What do you intend to accomplish, measure, improve or affect?
T	Time	What is the appropriate follow-up time to assess outcome?

Table 2. PICOT criteria adjusted to fit better in computer science research.

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 this], then [this] 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. (revise this)

Furthermore, when conducting empirical research - as we propose in this paper - (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

F	Feasible	• Adequate number of subjects • Adequate technical expertise • Affordable in time and money • Manageable in scope
I	Interesting	• Getting the answer intrigues investigator, peers and community
N	Novel	• Confirms, refutes or extends previous findings
E	Ethical	• Amendable to a study that institutional review board will approve
R	Relevant	• To scientific knowledge • To clinical and health policy • To future research

Table 3. FINER criteria for a good research question [FPFB09]

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. (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. (revise tis)

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 Formwe would like to mention them for reasons of completeness.

4.3 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] TODO: cite EBSE guidelines? 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 TODO: ref SEED and Software Engineering Evidence Map, but both have not been updated recently TODO: (SEED: prototype, 2009; Ev Map 2012). 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 usefulTODO: rephrase:

- Think about your search strings and engines beforehand and write them down.
- It is unlikely to find all relevant literature using only a single search engine [BKB+07]. TODO: how good is Google Scholar today?, GS as good starting point (publisher bias)?
- *Snowballing* is useful, both in a forward and backward manner [Woh14].
  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 within the research area or there are uncertainties about wording and namingTODO: rephrase

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

4.4 Designing, Conducting And Interpreting Experiment

- propose our Briefing Formthat is described later.
- set in context of evidence generation
- reference to books on how to design, conduct and analyse experiments. (Experimentation in SE, How to measure UX)

4.5 Answer Question

TODO

4.6 Discussion

- use EBSE checklist described in "Search for existing evidence"

4.7 Evaluate Process

- describe AAR
- describe PMA
- describe why to reflect on research process?
- distinguish from study discussion.

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 searchability 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. It is supposed to shift the focus to the important aspects in each step and sensitize for critical errors.TODO: does this sentence suit better for the checklist?
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 Formconsists of three parts: Research Question/Hypothesis, Experiment and Conclusion. These are explained in the following.

TODO: supports researcher in understanding the field of research? TODO: supports EBSE?

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 4.2). The research question in a asking form which contains technology, context and effect. Or it is constructed according to the PICOT criteria. And the hypothesis in form of a testable prediction (e.g., *"If [I do this], then [this] will happen."* [Bud10]).

5.2 Experiment

To find cause and effect relationships between variables experiments are conducted [Bud]. A more detailed view on experiments in our EBSE workflow TODO: (how do we call this?) is given in section 5.2TODO: check if this is actually the case. 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.

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 choise 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.

TODO: maybe put this whole description part of experiment already in workflow and just state here that it is vital for the briefing.

Independent Variables The variables that are changed by the researcher during the experiment are called independent variables. In the pair programming example from above TODO: (link to example above) 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 above TODO: (link to pair programming example).

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 we included a section for research techniques in the briefing form. The conducting researcher gives here a briefing 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. TODO: Find some sources. Maybe in Experimentation in Softwareengineering or how to measure user experience books

Statistic 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 4.4.

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. In the end it must be clear if the given research question is answered and in what direction. TODO: last sentence is not so nice.

6 Discussion

propose a digital version

References

[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 Jorgensen. Evidence-based software engineering for practitioners. *IEEE Software*, 22(1):58–65, 2005.

[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.

[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 Jorgensen. 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.

[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.

[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.

[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.

[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.