Titel

Student*

Advisor: Mitarbeiter†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*student@student.kit.edu
†mitarbeiter@teco.edu

Abstract. TBD

Keywords: TBD

1 Introduction

motivation

2 Fundamental Principles

short introduction to EBSE,..

- **Evidence-based approach**: integrate all available research (evidence) in decision making process
- **Aim**: "EBSE aims to improve decision making related to software development and maintenance by integrating current best evidence from research with practical experience and human values." [7]
- **Five steps** of practising EBSE [14]:
  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.

  → important tool: Systematic Literature Review (SLR)
- **SLR** [12]: identify and interpret all available literature regarding a research question → papers should be written for synthesis (TODO requirements for this, common mistakes/problems?)

3 Related Work

SEED, "a preliminary empirical investigation of the use of EBSE by undergraduate students"

4 Our Approach/Guidelines

4.1 Setting up our design guidelines

The checklist (TODO name?) is meant to implicitly guide the user's approach to experimenting. By guiding the user, typical mistakes might be prevented. To create guidelines that help preventing typical users' mistakes, these mistakes first need to be identified. In this section, experiences and guidelines found in related work are discussed. The conclusions are used as basis for design of our guidelines. The first set of guidelines is based on the report of Rainer et al. [15]:

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)	TODO
"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 possibly simple 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 Give a hint/indication?
"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. To limit this paper's scope, we focus on scientific evidence.

4.2 Research Question, Hypothesis, and Objectives

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 [9]. These three 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."* [9, p. 280]

Research Question The question the later study is designed to answer is called research question [6]. It should be an answerable question and address a relevant issue in the research area [8]. 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 [9].
There are two general classes of research questions: qualitative and quantitative questions. Qualitative research states questions which report, describe, or explore a subject [5, p. 139-141]. In computer science as the research field matures these questions become more and more rare (find source). Therefore focus is on quantitative research questions in this paper. *"Quantitative research questions inquire about the relationships among variables"* [5, 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 [17] (maybe cut out).
To design a good research question Haynes coined the acronym PICO: Population, Intervention, Comparison group, and Outcome [1]. 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 [9]. Before PICOT Sackett and colleagues suggested that good research questions consist out of three components: Intervention, Context and Outcome [16], 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?"* [8, p. 60] Here the intervention is pair programming, the context of interest are professional software developers, and the outcome is improved code quality [8]. 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 [9]. A more detailed view of

the FINER criteria can be seen in Table 1. (TODO specify more tips for writing a good question. Creswell2014)

Table 1. FINER criteria for a good research question [9]

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

P	Population	• What specific patient 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?

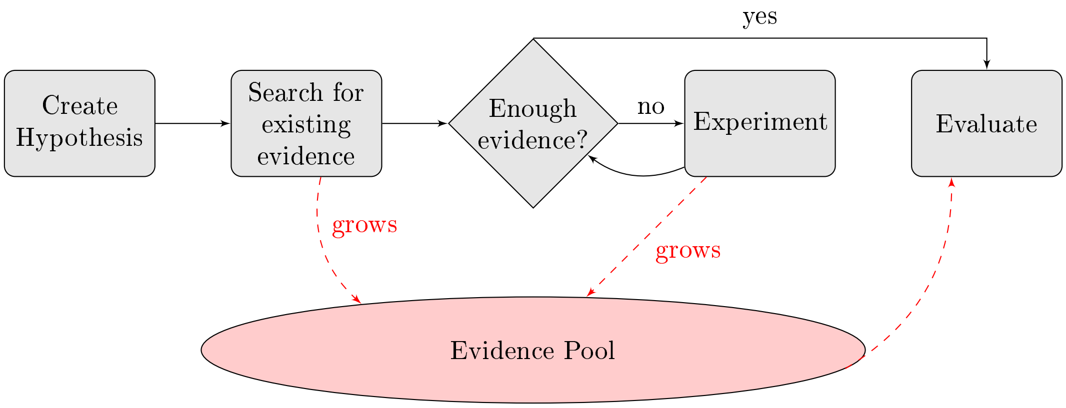
Hypothesis

- What is it?
- For what is it?
- educated guess (Science Buddies, Prasad)
- is testable statement (Prasad)

- is a prediction (Prasad)
- no interpretation (Prasad)
- relationship between two or more variables (Vickers)
- Only in Quantitative research (Creswell)
- two-sided/one-sided hypothesis -> use only two-sided (Farrugia et al.)
- Null hypothesis in empirical work (Farrugia et al.)
- contains varaibales/population/relationship (Vickers)
- How to hypothesis:
  - (Prasad)
  - from websites (Science Buddies)
  - (Creswell)

Objectives Sometimes researchers define objectives to their hypotheses. They are active statements that *"define specific aims of the study and should be clearly stated"* [9, p. 280] at the beginning of research. Objectives help to define the study (e.g. helping to calculate sample size). [9,6]

4.3 Workflow



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

4.4 Checklist

Question

Contains ***technology*** in a ***context*** showing an ***effect***. TODO Kitchenham Quote (practitioners)?
"Does pair programming in professional software development teams increases code quality?"

Hypothesis

Needs to contain a ***prediction*** and needs to be ***testable***.
"If you do x, then y will happen"

Experiment

Context

Dependent Variables
Variables that are ***measured*** during the study.

Independent Variables
Variables that are ***changed*** during the study.

Method
Lab-/Field study, number of participants, metrics, ...

Results
Experiment's outcome
No interpretation or conclusion!

Conclusion

Interpretation of experiment's results.
Verifying or Falsifying Hypothesis.
Scope of generalization.

4.5 Structured abstracts in software engineering

- Suggestion of elements: Background or Context, Objective or Aim, Method, Results, Limitations and Conclusion. [10] [11]
- About completeness and clarity of structured abstracts:
  1. Structured abstracts include more relevant information and are easier to read than conventional abstracts. [4] [3]
  2. Inexperienced authors are likely to produce clearer and more complete abstracts when using a structured form.[2]
- On average structured abstracts are longer and have better readability than unstructured abstracts. [13]

5 Discussion

i just cleaned this mess up..

References

1. Brian Haynes, R.: Forming research questions. Journal of Clinical Epidemiology 59(9), 881–886 (2006)
2. Budgen, D., Burn, A.J., Kitchenham, B.: Reporting computing projects through structured abstracts: A quasi-experiment. Empirical Software Engineering 16(2), 244–277 (2011)
3. Budgen, D., Kitchenham, B., Charters, S., Turner, M., Brereton, P., Linkman, S.: 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 pp. 64–72 (2007)
4. Budgen, D., Kitchenham, B.A., Charters, S.M., Turner, M., Brereton, P., Linkman, S.G.: Presenting software engineering results using structured abstracts: A randomised experiment. Empirical Software Engineering 13(4), 435–468 (2008)
5. Creswell, J.W.: Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (2014)
6. Dr. Peter Vickers, D.M.O.: Developing a Healthcare Research Proposal: An Interactive Student Guide, `http://www.health.herts.ac.uk/immunology/Webprogramme-Researchhealthprofessionals/hypothesisresearch_question.htm`
7. Dyba, T., Kitchenham, B.A., Jorgensen, M.: Evidence-based software engineering for practitioners. IEEE software 22(1), 58–65 (2005)
8. Dybå, T., Kitchenham, B.A., Jorgensen, M.: Evidence-based software engineering for practitioners. IEEE Software 22(1), 58–65 (2005)
9. Farrugia, P., Petrisor, B.A., Farrokhyar, F., Bhandari, M.: Practical tips for surgical research: Research questions, hypotheses and objectives. Canadian journal of surgery. Journal canadien de chirurgie 53(4), 278–281 (2009)
10. Jedlitschka, A., Ciolkowski, M., Pfahl, D.: Reporting experiments in software engineering. In: Guide to advanced empirical software engineering, pp. 201–228. Springer (2008)
11. Jedlitschka, A., Pfahl, D.: Reporting guidelines for controlled experiments in software engineering. 2005 International Symposium on Empirical Software Engineering, ISESE 2005 pp. 95–104 (2005)

12. Keele, S.: Guidelines for performing systematic literature reviews in software engineering. In: Technical report, Ver. 2.3 EBSE Technical Report. EBSE (2007)
13. Kitchenham, B.A., Brereton, O.P., Owen, S., Butcher, J., Jefferies, C.: Length and readability of structured software engineering abstracts. IET Software 2, 37 – 45 (2008), http://www.redi-bw.de/db/ebsco.php/search.ebscohost.com/login.aspx%3fdirect%3dtrue%26db%3daph%26AN%3d30193038%26site%3dehost-live
14. Kitchenham, B.A., Dyba, T., Jorgensen, M.: Evidence-based software engineering. In: Proceedings of the 26th international conference on software engineering. pp. 273–281. IEEE Computer Society (2004)
15. Rainer, A., Hall, T., Baddoo, N.: A preliminary empirical investigation of the use of evidence based software engineering by under-graduate students. 10th International Conference on Evaluation and Assessment in Software Engineering (EASE 2006) (2006)
16. Sackett, D.: Evidence-based medicine : how to practice and teach EBM (2000), http://www.ncbi.nlm.nih.gov/pubmed/12037026
17. Shaw, M.: What makes good research in software engineering? International Journal on Software Tools for Technology ... 4(1), 1–7 (2002), http://link.springer.com/article/10.1007/s10009-002-0083-4