Empirical Standards

Torben Brennecke

University of Kiel, Christian-Albrechts-Platz 4, 24118 Kiel, Germany

Abstract. In the field of writing and reviewing software engineering studies there is a wide variety of approaches and no real guide to structure and build a paper. Therefore researchers are dependent on the reviewers and their experience in reviewing papers. To address this problem, some empirical standards for peer reviewing papers in the field of software engineering studies are pointed out. With these standards, reviewers do not have to create their own review criteria and researchers know the criteria. This allows them to build their work to fully meet these criteria and have a basis for discussion with the reviewers in case of doubt. In addition, advantages arise for all members of the software engineering scientific community.

Keywords: empirical standards · review

1 Introduction

In 2004 BA. Kitchenham et al. published the paper *Evidence-based Software Engineering* [20]. In this paper the necessity of *Empirical Standards* was already emphasised at that time. The objective was "to describe how software engineering might benefit from an evidence-based approach and to identify the potential difficulties associated with the approach". They tried to map the "organisation and technical infrastructure supporting evidence-based medicine" onto the topic of software engineering.

Sixteen years later, in 2020, Paul Ralph et al. published the paper *Empirical Standards* [25] in which they introduced some ground rules in the context of reviewing paper. These ground rules can be applied on both sides of the reviewing process - by the reviewer and bye the researcher. It contains a general standard which can be applied to any research paper and methods for engineering, qualitative and quantitative research. Additionally, it contains interesting examples and suggested readings for each method presented.

Unfortunately, the paper has not been finished yet and some methods are still under construction. This process is visible and available on the GitHub page [8] of this project. This paper is the first to declare methods in a standardized in order to easily find methods for any kind of scientific research on software engineering.

In the mean time of both paper, several other papers have been published on the topic of reviews and standards in software development. As an example consider Barbara Kitchenham (a co-author of the Empirical Standards Paper)

and some of her colleagues [16], [17], [19], [18] Jorge Biolchini et al. [11], [9], P. Mian et al. [24]. However, they have never published structured methods to review papers.

Moreover, the necessity of empirical standards and how they are created will be discussed.

2 Foundations

In this part the methods used in software engineering research and for which an empirical standard, according to the work of Paul Ralph et al. [25], would be necessary will be explained.

2.1 Engineering methods

Engineering Research is used when new technological artifacts are introduced and evaluated through research. The research is limited to the detailed description of the artifact and possible guidance.

2.2 Qualitative Methods

Qualitative Methods focuses on evaluating non-standardized data to get a deeper understanding on the topic. [22]

Action Research - Seeks to examine how changing a process, such as through a new or changed algorithm, affects real life. The focus is on a detailed description of the changes that happen in real life.

Case Study - An empirical inquiry that investigates a contemporary phenomenon (the "case") in depth and within its real world context, especially when the boundaries between phenomenon and context [are unclear]" [26].

Grounded Theory - Tries to get the key patterns about a specific topic. The patterns emerge from iterative and nested rounds of data collection and elicitation

Qualitative Surveys (Interview Studies) - Studies which use interviews with open-ended questions to come to a result. Here, a subjective and individual opinion can be expressed and respected.

2.3 Quantitative Methods

Quantitative Methods are methods which "emphasize objective measurements and the statistical, mathematical, or numerical analysis of data collected through polls, questionnaires, and surveys, or by manipulating pre-existing statistical data using computational techniques. Quantitative research focuses on gathering numerical data and generalizing it across groups of people or to explain a particular phenomenon." [23]

Experiments (with Human Participants) - A study in which a change in a process is introduced in order for the change to be observed in reality. In contrast to the qualitative method case study, the environment is controlled and observed.

Questionnaire Surveys - A study which uses questionnaires or manuscripts for interviews. Thereby the questions are closed and can only allow scaleable answers.

Systematic Reviews - Such reviews are based on primary and secondary literature, which is evaluated, analyzed, and summarize to get a result.

3 Introduced methods

This part will show and discuss the introduced methods as shown in Fig 1 to review papers. The methods are divided into three parts. For engineering, qualitative, and quantitative research methods are introduced and summarised. For detailed information, good exemplars and suggested readings the paper from Paul Ralph et al. [25] is to be used.

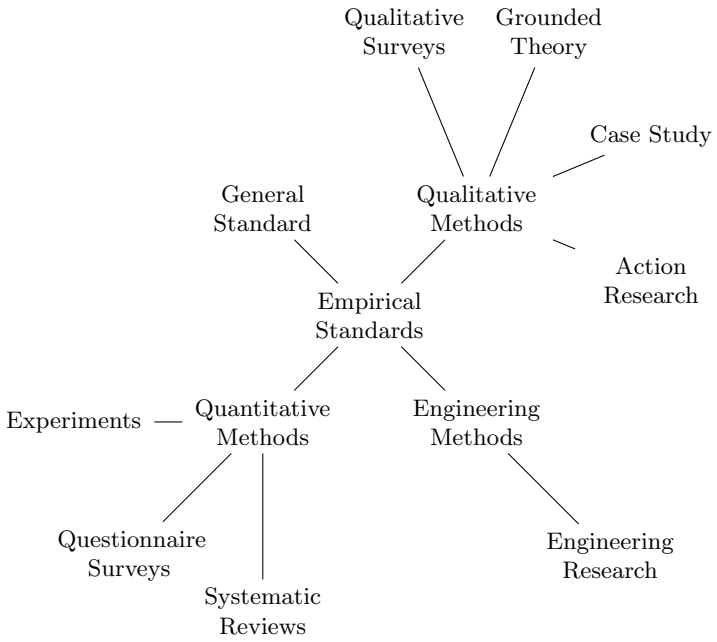

Fig. 1. Introduced methods

3.1 General

The General Standard "applies to all software engineering studies that collect and analyze data" [25]. This standard is the basis for all reviews and should be supplemented by a specific standard where possible.

This checks the general information and criteria of the paper, which should always be met. It lists not only criteria for the paper, but also to the reviewer, e.g., sufficient expertise and objectivity. The listed *Invalid Criticisms and Reviewing Antipatterns* try to prevent unfair reviewing and maintain professionalism of the reviewer. E.g. the critique should be objective and directed at the work and not at the researcher, there should be no examples of methods that might fit better, and he should read the work carefully with his full attention.

Some expectations of the researcher are also important. The researcher should know the expectations and guidelines of his study, the work should be necessary and self-contained, and all results of tests should be published.

3.2 Engineering methods

The first specific method introduced is *Engineering Research* which is used when new technological artifacts are introduced and evaluated through research. Specific features include the need for a detailed and consistent description, competitiveness with the state-of-art, and the use of an evaluation method. These are explained below.

A general quality criteria is the "Relationship of innovativeness to rigorousness" [25]. It says that if an artefact is less innovative, it requires an unbending assessment. From this relationship a problem for criticisms derivatives.

An introduced artefact may not be reasonably reviewed because there are no examples from the source code and only a small evaluation. This can happen when an introduced artefact is very advanced but there are practical or ethical reasons that prevent full disclosure of the source code.

3.3 Qualitative methods

The following section summarises methods from the field of qualitative methods. The summarised methods are *Action Research*, *Case Study*, *Grounded Theory*, and *Qualitative Surveys (Interview Studies)*.

Action Research The review standard for action research seeks to ensure that the study is detailed, the implications of the proposed method or tool are clear, and the relationship between the researcher and a potential founder are explained. Also, it is important to know that it is okay to hide details about participants or that a negative impact is justifiable if the reason is given. It is very important for researchers to keep a professional distance from the study. If this is not done the discovery could be overestimated because the opinion of the participants, which may be negative, is ignored.

The reviewer must be aware that it is *Action Research*, so it is not usual to publish quantitative data, transcripts or protocols

Case Study The case study standard presented ensures that the case study and type of case study is the appropriate method for the research question and that its use is justified. It is also important that the context, unit of analysis, and "chain of evidence" are well defined.

As mentioned in section 2.2, a case study only examines one specific case, so it should not matter to the reviewer whether there is quantitative data or whether the results can be generalised to comparable cases.

Grounded Theory The method presented emphasises the detailed explanation of the iterative and nested process of data collection and the chain of evidence from the collected data to the key patterns. To make this process easier the research should use a common grounded theory version from Glaser [14], Strauss-Corbin [13], Charmaz [12].

As with the other qualitative methods, the lack of quantitative data, the lack of reproducibility due to lack of transcripts or lack of representativeness should not be a point of criticism.

3.4 Quantitative methods

The following section summarises methods from the field of quantitative methods. The summarised methods are *Experiments (with Human Participants)*, *Questionnaire Surveys*, and *Systematic Reviews*.

Experiments (with Human Participants) The presented standard for experiments characterises it as essential that the study contains a previously established hypothesis that is to be proven or disproven with acquired knowledge. This requires a detailed description of the entire setting, procedure and recording of the variables of the experiment. The reviewer should not criticise the selection of participants without knowing the reason for the selection or a possible repetition of the experiment.

Questionnaire Surveys The standard for questionnaire surveys tries to ensure a valid survey. A good explanation of the target group is inescapable. This contains a description of the selection of this group and a description of the participants in the context of the research question. The data which arises during the survey must be fully documented. In the process of the review the reviewer must hide his own opinion about the questions in the questionnaire and should accept also surprising or controversial results.

Systematic Reviews A systematic review must contain a description of the whole process of search, gather, and extract data from the primary or secondary literature. Moreover a type of the systematic review [15] must be identified.

3.5 Methods under construction

According to the current state the paper is unfortunately not finished. The paper mentioned Artifact Evaluation, Case Survey, Discourse Analysis, Protocol Analysis, Exploratory Data Science (Repository Mining), Longitudinal Studies, Simulation, Multi-Methodology (Supplement), and Replications (Supplement). For every mentioned method there are placeholders in the paper. The missing parts are created by the community behind the project (mentioned in 7).

4 Similarity to EBSM (Evidence-based Software Engineering)

BA. Kitchenham et al. formulated in their paper *Evidence-based Software Engineering* [20], among other things, the goal of evidence-based software engineering would provide "A common goal for individual researchers and research groups to ensure that their research is directed to the requirements of industry and other stakeholder groups.". In the paper there are no real guidelines on methods to do so, but necessity was underlined and affirmed. But it recommends to "adopt as much of the evidence-based approach as is possible, to target systematic reviews (or summaries of such reviews)".

The methods mentioned above aim more or less at the same goal and fulfilled it. One can therefore say that the target, although sixteen years later and in a different paper, has been fulfilled.

5 Benefits of Empirical Standards

The benefits of Empirical Standards are far reaching. Everyone who comes in contact with scientific paper will benefit. The main benefit for program chairs, reviewers and researchers is a constant base for discussion when in doubt.

If Empirical Standards reach an general validity, program chairs can use those as acceptance criteria for their program. In this case, they do not need to create their own review criteria or have reviewers create them, in which case they may vary. This leads in higher consistency and quality in the review process, higher acceptance rate of papers. The recruitment of reviewers is simplified, since the methodological knowledge is negligible and only the technical expertise is relevant and the workload of creating criteria is eliminated. In addition, as mentioned above, it also prevents possible discussions regarding unfair punishment of scientists by the reviewer. Unfair in this context means, a different reviewer would use lower criteria than the one who reviewed. It is still possible for a reviewer to place too much or too little emphasis on a criterion, but in this case there is a basis for discussion. Since the criterion is used by other reviewers and has a value which is recognized by the public (reviewers and researchers).

For reviewers, the work is immensely simplified if they use Empirical Standards. They do not have to create their own criteria and defend them to researchers or program chairs, resulting in more time for the actual review process.

Reviewers only have to work through the "checklist" (empirical standard for the used method). Moreover universally valid criteria resulting in an consistent review process, so its only in a few cases necessary that a third reviewer must review a paper. The first two reviews should be similar and differ only in details, not in a fundamental difference of opinion about the approach of the work.

The benefits for researchers are easy identified. From beginning of the paper they have "research objective" but also a "methodological objective", presented as the empirical standard of the method they want to use. More specifically, they can review their own work before submission and check for themselves whether the paper meets the requirements or not. This leads to a higher acceptance rate and faster publication time, as there is less need for revisions if the quality of the paper is already good at submission. There are also benefits after the submission of a paper. They are less dependent on the "mood" of the reviewer, the possible criticism is more objective and focused, and if the criticism is justified, it is very clear where the problem lies.

All this benefits results in benefits for the company of all persons who come into contact with papers (program chairs, reviewer, researcher, companies which commission paper, employees in those companies, students, etc.). First, the quality of all published papers is better because all papers meet the same high standards, and second, the financial expenditures to produce a paper are lower because the review and revision process is much shorter.

6 Challenges of Empirical Standards

Upon the many and far reaching benefits of empirical standards there are some challenges.

First of all, the overview must be completed. As long as it is unfinished, it is not possible to use the *Empirical Standard* as a complete and generally accepted basic paper. A good thing at this point is the support of the *ACM Special Interest Group on Software Engineering* [7]. With this support and the further development with the state-of-the art tool GitHub, the number of participants and the reach of the paper is higher than with a small publication.

Another challenge is to reach a general acceptance in the software engineering research community. Again the support of the *ACM Special Interest Group on Software Engineering* is a good point but not sufficient. The *Institute of Electrical and Electronics Engineers (IEEE)* [5], as a host of big and important conferences like the *International Conference on Software Engineering (ICSE)* [6] should use the *Empirical Standard* [25] as their review guideline.

Furthermore the *Empirical Standard* does not replaces experts of the respective topic of paper to be reviewed. You can't review a paper just on his methodical context. In a technical investigation, for example, the programming language must be known in order to be able to assess the level of detail of the description, or know the techniques used in the environment, such as Docker

or Kubernetes, in order to be able to correctly assess the functionality of the investigation.

Also updates or changes during a delivery period on the *Empirical Standards* used for a programm or conference can lead into discussion. Such changes are possible when a programme or conference reweights some parts of the standards, adds or removes some parts. The researcher works with the old standard and applies the paper with a good conscience to meet all requirements, while the reviewer uses a newer version at the review process. Such possible situations need a fair predefined solution, e.g. an extension of the deadline or an exception.

7 Researcher/Team

The initiative for the paper *Empirical Standards* was created by the ACM SIGSOFT. ACM is a scientific society for computer science and stands for Association for Computing Machinery [2]. This association have some "Special Interest Groups (SIG)" for many topics in the field of computer science. The Special Interest Group on Software Engineering (SIGSOFT) [7],[1] works in the field of software engineering. This group started the *Paper and Peer Review Quality* initiative. The work stated by requesting volunteers by mailing lists and social media to contribute input and some people to co-chair the initiative and the paper. Paul Ralph (Dalhousie University) and Romain Robbes (Free University of Bozen-Bolzano) were found to co-chair the paper.

At the start of the initiative they want to introduce and improve empirical standards and peer reviewing. After some research, however, they focused on the topic of empirical standards, as there is already recent work, e.g. by Tine Köhler et al. [21] and Lonni Besançon et al. [10], on improving peer reviewing.

In the process of creating the paper *Empirical Standards*, more than forty volunteers worked on the paper respectively the including standards. They reviewed the plan for drafting the standards, discussed which methods should have a standard, created a template illustrating the sections each standard should have, suggested additional researchers who might want to volunteer and nominated subject matter experts to draft each standard. Under the supervision of one to three *subject matter experts*, the work results were then inserted into the in section 3 mentioned methods.

The created drafts were put into the paper and was sent to all volunteers to get feedback. This feedback was then incorporated into the paper. Then the paper was uploaded to GitHub to present the work to the public.

For comparsion, the paper *Evidence-based Software Engineering* [20] was admittedly presented at the ICSE'04 [4] but the team behind the paper consists only of three people - Barbara A. Kitchenham, Tore Dybå, and Magne Jørgensen. For this reason, and because of the significantly lower level of networking via the internet, the possibilities much smaller.

After the release of the version 0.1.0 the continuing work consist of maintaining the developed standards and finishing the unfinished ones. This work will be done by some *maintainers* for each standard, who are responsible to

reviewing issues and change requests from the GitHub page. This maintainers have three requirements to meet. Have a PhD or equivalent terminal degree in software engineering, computer science or a related discipline, have published one or more studies related to the standard in a prestigious software engineering journal or conference in the past six years, and be approved by the steering committee. The steering committee consist of the director of the empirical standards project, the editors-in-chief (or their designated representatives) of all the DBLP-indexed [3] journals that adopt the standards, a designated representative of the steering committee of each DBLP-indexed conference that adopts the standards and two, elected, early-career members-at-large.

8 Conclusion

In summary, I can say Empirical Standards for paper in software engineering research are necessary. The benefits which i mentioned in section 5 would lead to higher quality of all published papers and perhaps motivate more researchers to submit their papers to some programs or congresses. This leads to the fact that a larger number of people would read a paper that contains a perhaps outstanding invention than if the paper is only published on a small company or university website.

Moreover the complete process of reviewing a paper is less personal. All involving parties have much less reason to criticise the person of the reviewer, researcher, or program chair because the process of reviewing is complete guides by the Empirical Standard.

The whole process of build an Empirical Standard is not even finished. The *Empirical Standards* from Paul Ralph et al. [25] is a good start, but guidelines for methods are still under construction and for some there are already suggestions for improvement on the GitHub page [8] of mentioned paper.

I think it's good that it's quite easy to participate in the project. Git is a state-of-art version control system, mainly for program code but also for working with issues. For any method it is a very good way of working for this paper. This makes it possible, even as a "freshman", to contribute ideas to the community behind this project.

References

1. Acm - Special Interest Group on Software, https://www.acm.org/special-interest-groups/sigs/sigsoft, visited 2021-01-02
2. Association for Computing Machinery, https://www.acm.org/, visited 2021-01-02
3. dblp: computer science bibliography, https://dblp.org/, visited 2021-01-02
4. ICSE'04, http://www.icse-conferences.org/2004/, visited 2021-01-02
5. Institute of Electrical and Electronics Engineers, https://www.ieee.org/, visited 2021-01-02
6. International Conference on Software Engineering, https://conf.researchr.org/home/icse-2021, visited 2021-01-02

7. SIGSOFT, https://www.sigsoft.org/index.html, visited 2021-01-02
8. EmpiricalStandards on GitHub (Dec 2020), https://github.com/acmsigsoft/EmpiricalStandards, visited 2021-01-02
9. de Almeida Biolchini, J.C., Mian, P.G., Natali, A.C.C., Conte, T.U., Travassos, G.H.: Scientific research ontology to support systematic review in software engineering. Advanced Engineering Informatics **21**(2), 133–151 (2007)
10. Besançon, L., Rönnberg, N., Löwgren, J., Tennant, J.P., Cooper, M.: Open up: a survey on open and non-anonymized peer reviewing. Research integrity and peer review **5**(1), 1–11 (2020)
11. Biolchini, J., Mian, P.G., Natali, A.C.C., Travassos, G.H.: Systematic review in software engineering. System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES **679**(05), 45 (2005)
12. Charmaz, K.: Constructing grounded theory. sage (2014)
13. Corbin, J., Strauss, A.: Basics of qualitative research: Techniques and procedures for developing grounded theory. Sage publications (2014)
14. Glaser, B.G.: The future of grounded theory. Qualitative health research **9**(6), 836–845 (1999)
15. Grant, M.J., Booth, A.: A typology of reviews: an analysis of 14 review types and associated methodologies. Health Information & Libraries Journal **26**(2), 91–108 (2009)
16. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)
17. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering–a systematic literature review. Information and software technology **51**(1), 7–15 (2009)
18. Kitchenham, B., Brereton, P.: A systematic review of systematic review process research in software engineering. Information and software technology **55**(12), 2049–2075 (2013)
19. Kitchenham, B.A.: Systematic review in software engineering: where we are and where we should be going. In: Proceedings of the 2nd international workshop on Evidential assessment of software technologies. pp. 1–2 (2012)
20. Kitchenham, B.A., Dyba, T., Jorgensen, M.: Evidence-based software engineering. In: Proceedings. 26th International Conference on Software Engineering. pp. 273–281. IEEE (2004)
21. Köhler, T., González-Morales, M.G., Banks, G.C., O'Boyle, E.H., Allen, J.A., Sinha, R., Woo, S.E., Gulick, L.M.: Supporting robust, rigorous, and reliable reviewing as the cornerstone of our profession: Introducing a competency framework for peer review. Industrial and Organizational Psychology **13**(1), 1–27 (2020)
22. Labaree, R.V.: Research Guides: Organizing Your Social Sciences Research Paper: Qualitative Methods, https://libguides.usc.edu/writingguide/qualitative
23. Labaree, R.V.: Research Guides: Organizing Your Social Sciences Research Paper: Quantitative Methods, https://libguides.usc.edu/writingguide/quantitative
24. Mian, P., Conte, T., Natali, A., Biolchini, J., Travassos, G.: A systematic review process for software engineering. ESELAW **5**, 2nd (2005)
25. Ralph, P., Baltes, S., Bianculli, D., Dittrich, Y., Felderer, M., Feldt, R., Filieri, A., Furia, C.A., Graziotin, D., He, P., et al.: Acm sigsoft empirical standards. arXiv preprint arXiv:2010.03525 (2020)
26. Yin, R.K.: Case study research and applications: Design and methods. Sage publications (2017)