

From *Ad-hoc* to Rapid Reviews: a systematic (and not so rapid) evolution – Keynote Speech

Guilherme Horta Travassos
Universidade Federal do Rio de
Janeiro, Coppe, PESC
Rio de Janeiro, Brazil
ght@cos.ufrj.br

ABSTRACT

Literature reviews used to be the seed of most investigations and research activities. They are usually simple to execute and mostly are performed *ad-hoc*. However, *ad-hoc* literature reviews, despite their usefulness to start studies on determining topics of interest, typically present low scientific strength due to a lack of replicability, coverage, and dependability of the researcher's experience. Therefore, researchers have invested in supporting more systematic literature reviews to minimize bias and provide more reliable findings to support their conclusions. It includes different strategies such as multivocal reviews, mapping studies, (quasi) Systematic Literature Reviews, and currently, Rapid Reviews. This talk intends to discuss and exemplify these different review strategies and their features in the context of Evidence-Based Software Engineering.

Author Keywords

Literature Studies, Systematic Literature Reviews; Mapping Studies, Multivocal Reviews, Rapid Reviews, Evidence-Based Software Engineering.

ACM Classification Keywords

•General and reference~Document types~Surveys and overviews •General and reference~Cross-computing tools and techniques~Empirical studies

DOI: 10.37789/rochi.2022.1.1.1

INTRODUCTION

Empirical methods evolved in the last decades into the context of software engineering and their related areas. Different strategies for primary studies have been investigated and used to support empirical and experimental studies in the field [4]. It resulted in many primary studies (surveys, experiments, case studies, among others) available and spread in the technical literature. Their results can be confirmatory, conclusive, non-conclusive, and contradictory. The aggregation of such studies can strengthen the beliefs on the observed phenomena and promote the evolution of knowledge. However, finding all primary studies of interest is challenging without a well-defined and organized research protocol.

Literature reviews used to be the seed of most investigations and research activities involving the search for technical

literature. They are usually simple to execute and are mostly performed unsystematically (*ad-hoc*). *Ad-hoc* reviews are narrative reviews prepared by experts to provide qualitative summaries of evidence regarding a particular topic of interest. Their simplicity influences the use of informal or subjective methods to acquire the sources of information and interpret them. Previous knowledge or preconceived perceptions can influence the citations of literature reinforcing such perceptions [9]. Therefore, despite their usefulness in starting studies on determining topics of interest, *ad-hoc* literature reviews typically present low scientific strength due to a lack of replicability, coverage, and dependability of the researcher's experience.

SECONDARY STUDIES

The need for a better organization of the research protocols and the growing number of primary studies contributed, among other factors, to investigating further strategies to support secondary studies, paving the road to systematically undertaking literature reviews in software engineering. The aims included increasing replicability, minimizing bias, and providing more reliable findings to support the conclusions regarding the primary studies and strengthen the evidence in software engineering and related areas.

The first movements promoted the Systematic Literature Reviews (SLRs) [2,8]. Inspired by the secondary studies undertaken mainly in medicine, SLRs revealed that much should be done regarding how the primary studies have been organized and reported. The first SLRs contributed to better organizing the field and identifying our findings' fragilities when expecting to aggregate quantitative studies with meta-analysis. Therefore, some adjustments took place in the research protocols to support *quasi*-Systematic Literature Reviews [11], aiming at answering research questions without the possibility of any quantitative aggregation. These literature reviews include the (i) formulation of a focused review question; (ii) comprehensive search and inclusion of primary studies; (iii) quality assessment of included studies and data extraction; (iv) synthesis of study results (meta-analysis not possible in *quasi*-SLRs), and (v) interpretation of results and report writing [9]. In addition, SLRs can benefit from automated tools [6]. Despite their threats to validity [1], SLRs are the "gold standard" in evidence-based software engineering [4].

However, most SLRs indicate the need to support a better characterization of the topics of investigation. The lack of a research agenda, a sound characterization of the topic of interest, and the inconsistency of primary studies' reports jeopardized answering the focused review questions of SLRs. Therefore, Mapping Studies [10] started to be promoted to characterize the field and offer a grounded characterization of the objects of study. So, such studies are recommended when it could be interesting to organize an initial body of knowledge characterizing an area of study, reporting the findings without expecting to produce a more robust quality assessment or synthesis.

The basis for secondary studies is the technical literature. However, software engineering and its related areas used to offer explicit anecdotal evidence from practitioners in different sources, motivating the undertaken of Multivocal Reviews [7].

SLRs take time and lots of energy from a team of researchers to provide valuable and significant results. They are a *de facto* instrument to sediment high confidence evidence about a phenomenon of interest. However, practitioners sometimes need to have answers to their practice questions. Usually, they do not have time to wait for the undertaken of an SLR. In these cases, and not intended to substitute an SLR, Rapid Reviews (RR) [3] can be used. An RR is not necessarily "fast," but the research protocol is adequately adjusted to reduce the formalism of an SLR. Therefore, RRs are lighter than SLRs because they can be performed by just one person and use narrative summaries to synthesize evidence.

CONCLUSION

A secondary study is a powerful tool to observe and aggregate primary studies. Literature reviews support secondary studies. However, they must be systematic to assure replicability, reduce bias, and strengthen the evidence. Different systematic literature review strategies can support the organization of a body of knowledge regarding a topic of interest.

This talk intends to discuss and exemplify these different review strategies and their features in the context of Evidence-Based Software Engineering.

ACKNOWLEDGMENTS

The author thanks the CNPq and FAPERJ for their support. He also thanks the ESE Group at UFRJ, ISERN, and RoCHI community.

REFERENCES

1. Ampatzoglou, A., Bibi, S., Avgeriou, P., Chatzigeorgiou, A. (2020). Guidelines for Managing Threats to Validity of Secondary Studies in Software Engineering. In: Felderer, M., Travassos, G. (eds) Contemporary Empirical Methods in Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-32489-6_15
2. Biolchini, J., Mian, P.G., Natali, A.C., Travassos, G.H. (2005). Systematic Review in Software Engineering: Relevance and Utility. Technical Report ES67905, PESC - COPPE/UFRJ. Available at <http://cronos.cos.ufrj.br/publicacoes/reltec/es67905.pdf>
3. Cartaxo, B., Pinto, G., Soares, S. (2020). Rapid Reviews in Software Engineering. In: Felderer, M., Travassos, G. (eds) Contemporary Empirical Methods in Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-32489-6_13
4. Kitchenham, B. A., Budgen, D., Brereton, P. (2015) Evidence-Based Software Engineering and Systematic Reviews. Chapman & Hall/CRC. ISBN-13 978-1482228656.
5. Felderer, M., Travassos, G.H. (2020). The Evolution of Empirical Methods in Software Engineering. In: Felderer, M., Travassos, G. (eds) Contemporary Empirical Methods in Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-32489-6_1
6. Felizardo, K.R., Carver, J.C. (2020). Automating Systematic Literature Review. In: Felderer, M., Travassos, G. (eds) Contemporary Empirical Methods in Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-32489-6_12
7. Garousi, V., Felderer, M., Mäntylä, M.V., Rainer, A. (2020). Benefitting from the Grey Literature in Software Engineering Research. In: Felderer, M., Travassos, G. (eds), Contemporary Empirical Methods in Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-32489-6_14
8. Kitchenham, B., Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering (EBSE Technical Report EBSE-2007-001). Available at https://www.elsevier.com/_data/promis_misc/525444/systematicreviewsguide.pdf
9. Pai, M., McCulloch, M., Gorman, J. D., Pai, N., Enanoria, W., Kennedy, G., Tharyan, P., Colford, J. M. Jnr (2004). Systematic reviews and meta-analysis: An illustrated, step-by-step guide. The National Medical Journal of India, 17(2), pp. 86-95.
10. Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, Volume 64, pp. 1-18, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2015.03.007>.
11. Travassos, G. H., Santos, P. S. M., Mian, P., Dias Neto, A. C., Biolchini, J. (2008). An Environment to Support Large Scale Experimentation in Software Engineering. In: IEEE International Conference on Engineering of Complex Computer Systems, 2008, Belfast, pp. 193-202. <http://dx.doi.org/10.1109/ICECCS.2008.30>