ACM SIGSOFT SEN Empirical Software Engineering: Introducing Our New Regular Column

Justus Bogner
Vrije Universiteit Amsterdam
The Netherlands
j.bogner@vu.nl

Roberto Verdecchia
University of Florence
Italy
roberto.verdecchia@unifi.it

ABSTRACT

From its early foundations in the 1970s, empirical software engineering (ESE) has evolved into a mature research discipline that embraces a plethora of different topics, methodologies, and industrial practices. Despite its remarkable progress, the ESE research field still needs to keep evolving, as new impediments, shortcoming, and technologies emerge. Research reproducibility, limited external validity, subjectivity of reviews, and porting research results to industrial practices are just some examples of the drivers for improvements to ESE research. Additionally, several facets of ESE research are not documented very explicitly, which makes it difficult for newcomers to pick them up. With this new regular ACM SIGSOFT SEN column (SEN-ESE), we introduce a venue for discussing meta-aspects of ESE research, ranging from general topics such as the nature and best practices for replication packages, to more nuanced themes such as statistical methods, interview transcription tools, and publishing interdisciplinary research. Our aim for the column is to be a place where we can regularly spark conversations on ESE topics that might not often be touched upon or are left implicit. Contributions to this column will be grounded in expert interviews, focus groups, surveys, and position pieces, with the goal of encouraging reflection and improvement in how we conduct, communicate, teach, and ultimately improve ESE research. Finally, we invite feedback from the ESE community on challenging, controversial, or underexplored topics, as well as suggestions for voices you would like to hear from. While we cannot promise to act on every idea, we aim to shape this column around the community interests and are grateful for all contributions.

1. INTRODUCTION

Empirical software engineering (ESE) emerged in response to the so-called "software crisis" of the 1960s, when growing system complexity revealed that theory alone was insufficient for effective software development [7]. While the term "software engineering" was first popularized during the 1968 and 1969 NATO conferences [13], it was only in the 1970s that researchers like Victor R. Basili at the University of Maryland began applying rigorous scientific methods to software engineering [3]. Basili's pioneering work, which includes well-known frameworks such as Goal-Question-Metric [2], laid the foundations for empirical practices in the field, and guided software engineering on the path of becoming a more rigorous, evidence-based research field. Over the years, many others have provided substantial methodological contribution to ESE research, e.g., Barbara Kitchenham for systematic literature reviews [9], Carolin Seaman for qualitative studies [15], or Natalia Juristo [8] and Claes Wohlin [22] for controlled experiments, to only name a few (there are, of course, many others).

From the 80s and 90s into the 21st century, methodologies expanded from small-scale lab and student experiments to larger industrial case studies and controlled experimentation, tied together by efforts to establish reproducibility, replication, and stronger methodological robustness in software engineering research [6]. This was by no means an easy transformation, as many early SE empiricists received considerable push-back by SE researchers coming from a rationalist background [17]. But over the past two decades, the ESE discipline has matured considerably: it now includes both quantitative and qualitative approaches, leverages data mining of large software repositories, employs systematic secondary studies and mixed-method designs, places strong emphasis on open science [11], and is supported by dedicated journals and conferences such as the Empirical Software Engineering journal (EMSE)[1], the International Symposium on Empirical Software Engineering and Measurement (ESEM)[2], and the International Conference on Evaluation and Assessment in Software Engineering (EASE)[3]. Lastly, we also have high-quality, modern ESE books at our disposal, such as Wohlin et al. [22], Felderer and Travassos [5], or Mendez et al. [12], an evolving collection of community-driven empirical standards [14], and have arguably become much better at the methodological education of the next generation of ESE researchers.

However, while we have made substantial progress as a field, there are still many things to improve. Over the years, countless members of the community have pointed out various deficiencies in ESE research. Some examples include:

- the suboptimal reproducibility of ESE studies [10], which has been pointed out several times in the last decades[4],
- the difficulty of accumulating evidence on a phenomenon via replications due to the tendency of reviewers to expect substantial novelty and "surprising" findings [16, 4],
- the bad practice of drawing broad conclusions from low-quality or non-generalizable evidence [19],
- the inconsistent ESE education at universities that insufficiently prepares students for typical modern SE careers [20],
- how the peer review processes of our flagship conferences tend to discourage innovative contributions on hard problems because they are too easy to reject[5],

[1] https://link.springer.com/journal/10664
[2] https://www.esem-conferences.org
[3] https://conf.researchr.org/series/ease
[4] https://cacm.acm.org/blogcacm/the-rise-of-empirical-software-engineering-ii-what-we-are-still-missing/
[5] https://sigsoft.medium.com/conferences-in-software-engineering-reflections-after-30-years-ca525d98c584

- how our unstructured peer review processes lead to subjective and unreliable paper evaluations [1],
- and that the state of SE academia-industry bridges and SE science communication is worthy of improvement [21, 23].

Despite our advances, empirical research in software engineering has by no means become easy, and demands much in terms of discipline, rigor, and expertise. While we have more resources available than ever to support us, it is unavoidable that many things also remain unclear or are not sufficiently documented in an actionable way for junior researchers to pick up. We recently noticed that there was little documented guidance for one of those things, namely how to write effective ESE papers.

2. NOTES ON WRITING ESE PAPERS

After reading and writing the n-th ESE paper, its structure, content, and drafting process become almost second nature to many seasoned ESE researchers. However, explaining all the nuances that ensure that an ESE study is well presented, especially to less-experienced researchers such as Master and PhD students, remains difficult. While the ESE community seems to have developed a reasonable shared understanding of how our ESE papers should roughly look like, we realized that this process is rarely documented and therefore hard to teach to newcomers. To improve the situation, we started drafting a document detailing the common structure of ESE papers, the content of their different sections, and general advice on how to write them effectively. Our goal with releasing the final document, also published in ACM SIGSOFT SEN not long ago [18], was not to set a new standard or lecture others on how ESE papers should be written. Instead, it was to share how we are used to addressing a somewhat difficult and not very well-documented ESE topic, i.e., how to write effective ESE papers, in the hope that it would support early-career researchers and hopefully open a discourse on how the ESE community writes their papers. We believe that there are many other ESE topics, e.g., some of the challenges mentioned in the introduction, that would benefit from a similar treatment.

3. A REGULAR ESE COLUMN: SEN-ESE

To start making progress towards this goal, we have been invited to start a regular ACM SIGSOFT SEN column about empirical software engineering (SEN-ESE). In the same spirit in which we released the *Notes on Writing Effective ESE Papers*, we envision this column to be a place where we can regularly surface and spark conversations on ESE meta-topics that might not often be touched upon or are left implicit. In the SEN-ESE column, we plan to cover aspects regarding all facets of ESE research, including but not limited to:

- general ESE research, e.g., the nature of replication packages and best practices for creating them or publishing negative results within ESE venues,
- quantitative research, e.g., the role of hardware components in runtime metrics measurements or which usage of statistical methods is generally accepted in most ESE venues,
- qualitative research, e.g., transcription tools and guidelines for their usage or best practices for sampling interview participants,
- topics related to writing and presenting ESE research, e.g., how to write a great discussion section or how to make engaging presentations about ESE research,
- and potentially even community aspects, such as typical reviewing or editorial processes of ESE venues or conducting and publishing interdisciplinary ESE research.

Since we are far from being seasoned experts in most of these topics ourselves, we plan to ground our upcoming column articles in one of the following formats:

- interview(s) or a focus group with experts on a topic,
- a questionnaire survey with the ESE community about a topic,
- or a position or opinion piece by us about a topic, with references to other ESE papers, and potentially with invited co-authors.

Overall, our goal with this regular column is not only to stimulate the ESE community to reason about how to effectively and systematically apply ESE research, but also how we can improve our community processes and research methods, to adapt them, and eventually to evolve them to improve the ESE research field as a whole. We hope that this column can ignite conversations that will ultimately lead to improving how ESE research is designed, executed, and reported.

4. CALL FOR FEEDBACK AND PARTICIPATION

While we outlined several concrete ideas above on what to cover in our column, we also want to explicitly request feedback from the ESE community, especially from more junior researchers, e.g., PhD students and postdocs. What are ESE topics that you still find challenging, even despite available documented guidance, or that you think could benefit from more nuanced, opinionated insights from an expert in the field? What are more controversial ESE topics that you would like to see discussed between experts of differing opinions? What are important topics that are rarely openly discussed or published about in the ESE community and about which you are curious? Are their well-known ESE community members whose experiences or advice on a certain topic you would be interested to hear? Let us know, and we will consider to cover your suggestions in our column! Similarly, if you feel like you have something to say about one of our mentioned topics or a different important ESE topic based on your expertise and experience, please also let us know. We will then consider whether to invite you for an interview or focus group. While we cannot promise to directly act on all received feedback, it is definitely our intention to shape this column according to broader community interests. We are therefore grateful for all feedback we receive, even if we likely cannot implement all of it. Thank you in advance, and we hope you will enjoy the SEN-ESE column! We are definitely looking forward to it!

5. ACKNOWLEDGEMENTS

We kindly thank Daniel Graziotin, University of Hohenheim, and Marvin Wyrich, Saarland University, for reviewing an earlier version of this article.

References

[1] Arham Arshad, Taher Ghaleb, and Paul Ralph. Towards a More Structured Peer Review Process with Empirical Standards. In *Evaluation and Assessment in Software Engineering*, pages 353–358, Trondheim Norway, June 2021. ACM. ISBN 978-1-4503-9053-8. doi: 10.1145/3463274.3463359.

[2] Victor R. Basili, Gianluigi Caldiera, and Dieter H. Rombach. *The Goal Question Metric Approach*, volume I. John Wiley & Sons, 1994.

[3] Barry Boehm, Hans Dieter Rombach, and Marvin V Zelkowitz. *Foundations of empirical software engineering: the legacy of Victor R. Basili*. Springer Science & Business Media, 2005.

[4] Neil A. Ernst, Jeffrey C. Carver, Daniel Mendez, and Marco Torchiano. Understanding peer review of software engineering papers. *Empirical Software Engineering*, 26(5):103, September 2021. ISSN 1382-3256, 1573-7616. doi: 10.1007/s10664-021-10005-5.

[5] Michael Felderer and Guilherme Horta Travassos, editors. *Contemporary Empirical Methods in Software Engineering*. Springer International Publishing, Cham, 2020. ISBN 978-3-030-32488-9. doi: 10.1007/978-3-030-32489-6.

[6] Michael Felderer and Guilherme Horta Travassos. The Evolution of Empirical Methods in Software Engineering. In Michael Felderer and Guilherme Horta Travassos, editors, *Contemporary Empirical Methods in Software Engineering*, pages 1–24. Springer International Publishing, Cham, 2020. ISBN 978-3-030-32488-9 978-3-030-32489-6. doi: 10.1007/978-3-030-32489-6_1.

[7] Yann-Gaël Guéhéneuc and Foutse Khomh. Empirical Software Engineering. In Sungdeok Cha, Richard N. Taylor, and Kyochul Kang, editors, *Handbook of Software Engineering*, pages 285–320. Springer International Publishing, Cham, 2019. ISBN 978-3-030-00261-9 978-3-030-00262-6. doi: 10.1007/978-3-030-00262-6_7.

[8] Natalia Juristo and Ana M. Moreno. *Basics of Software Engineering Experimentation*. Springer US, Boston, MA, 2001. ISBN 978-1-4419-5011-6 978-1-4757-3304-4. doi: 10.1007/978-1-4757-3304-4.

[9] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(TR/SE-0401):28, 2004. ISSN 13537776.

[10] Lech Madeyski and Barbara Kitchenham. Would wider adoption of reproducible research be beneficial for empirical software engineering research? *Journal of Intelligent & Fuzzy Systems*, 32(2):1509–1521, January 2017. ISSN 1064-1246, 1875-8967. doi: 10.3233/JIFS-169146.

[11] Daniel Mendez, Daniel Graziotin, Stefan Wagner, and Heidi Seibold. Open Science in Software Engineering. In Michael Felderer and Guilherme Horta Travassos, editors, *Contemporary Empirical Methods in Software Engineering*, pages 477–501. Springer International Publishing, Cham, 2020. ISBN 978-3-030-32488-9 978-3-030-32489-6. doi: 10.1007/978-3-030-32489-6_17.

[12] Daniel Mendez, Paris Avgeriou, Marcos Kalinowski, and Nauman Bin Ali, editors. *Handbook on Teaching Empirical Software Engineering*. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-71768-0 978-3-031-71769-7. doi: 10.1007/978-3-031-71769-7.

[13] Peter Naur and Brian Randell. *Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO*. 1969.

[14] Paul Ralph, Nauman bin Ali, Sebastian Baltes, Domenico Bianculli, Jessica Diaz, Yvonne Dittrich, Neil Ernst, Michael Felderer, Robert Feldt, Antonio Filieri, Breno Bernard Nicolau de França, Carlo Alberto Furia, Greg Gay, Nicolas Gold, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara Kitchenham, Valentina Lenarduzzi, Jorge Martínez, Jorge Melegati, Daniel Mendez, Tim Menzies, Jefferson Molleri, Dietmar Pfahl, Romain Robbes, Daniel Russo, Nyyti Saarimäki, Federica Sarro, Davide Taibi, Janet Siegmund, Diomidis Spinellis, Miroslaw Staron, Klaas Stol, Margaret-Anne Storey, Damian Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, Xiaofeng Wang, and Sira Vegas. Empirical Standards for Software Engineering Research. *arXiv*, 2020. doi: 10.48550/ARXIV.2010.03525.

[15] C.B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, July 1999. ISSN 00985589. doi: 10.1109/32.799955.

[16] Janet Siegmund, Norbert Siegmund, and Sven Apel. Views on Internal and External Validity in Empirical Software Engineering. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, pages 9–19, Florence, Italy, May 2015. IEEE. ISBN 978-1-4799-1934-5. doi: 10.1109/ICSE.2015.24.

[17] Walter Tichy. Workings of science: How software engineering research became empirical. *Ubiquity*, 2022(July):1–7, July 2022. ISSN 1530-2180. doi: 10.1145/3512339.

[18] Roberto Verdecchia and Justus Bogner. Notes on writing effective empirical software engineering papers: An opinionated primer. *ACM SIGSOFT Software Engineering Notes*, 50(3):24–36, 2025.

[19] Elaine J. Weyuker. Empirical Software Engineering Research - The Good, The Bad, The Ugly. In *2011 International Symposium on Empirical Software Engineering and Measurement*, pages 1–9, Banff, AB, Canada, September 2011. IEEE. ISBN 978-1-4577-2203-5 978-0-7695-4604-9. doi: 10.1109/ESEM.2011.66.

[20] Greg Wilson. Thoughts from a Not-So-Influential Educator. *ACM SIGSOFT Software Engineering Notes*, 45(3):21–22, July 2020. ISSN 0163-5948. doi: 10.1145/3402127.3402136.

[21] Greg Wilson, Jorge Aranda, Michael Hoye, and Brittany Johnson. Experience Report: It Will Never Work in Theory. *IEEE Software*, 41(3):80–82, May 2024. ISSN 0740-7459, 1937-4194. doi: 10.1109/MS.2024.3362649.

[22] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2nd edition, 2024. ISBN 978-3-662-69305-6 978-3-662-69306-3. doi: 10.1007/978-3-662-69306-3.

[23] Marvin Wyrich and Justus Bogner. Beyond Self-Promotion: How Software Engineering Research Is Discussed on LinkedIn. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society*, pages 85–95, Lisbon Portugal, April 2024. ACM. ISBN 9798400704994. doi: 10.1145/3639475.3640113.