IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154307

Survey-Based Research in Software Engineering

Kavya K[1], Layashree L[2], Raziya M[3], S Vaishnavi[4], Dr. Muhibur Rahman T.R[5]

6th Sem B.E.(CS&E), Ballari Institute of Technology and Management (BITM), Ballari, Karnataka – 583104[1-4]

Associate Professor, Department of Computer Science and Engineering,

Ballari Institute of Technology and Management (BITM), Ballari, Karnataka – 583104[5]

Abstract: Survey-based research in software engineering is an empirical method used to collect data from developers and organizations to understand current trends, challenges, and practices. This paper discusses survey design, sampling techniques, and data collection methods to ensure reliable and valid results. It highlights the role of statistical analysis in interpreting data and deriving meaningful insights. The study concludes that survey research supports improved software development processes and evidence-based decision-making.

Keywords*:* Include at least 4 keywords or phrases.

I. INTRODUCTION

This file is meant to be used as a guide when preparing your paper. You can find and download it from the conference's official website. If anything in the guidelines is unclear, you can reach out to the publications team through the contact details provided there. Details about submitting the final version of your paper are also available on the same website.

II. LITERATURE REVIEW

[1] Barbara A. Kitchenham and Stuart Charters (2007) presented comprehensive guidelines for conducting systematic literature reviews in software engineering. Their work explains each phase of the review process, including planning the review, defining research questions, identifying relevant studies, applying inclusion and exclusion criteria, and synthesizing findings. They emphasize the importance of transparency and repeatability so that other researchers can validate results. This study is highly important for survey-based research because it provides a structured approach to collecting and analyzing existing knowledge.

[2] Tore Dyba, Kitchenham, and Magne Jørgensen (2005) focused on evidence-based software engineering, which promotes the use of empirical data to guide decision-making in software development. The authors highlight how combining practitioner experience with research evidence leads to better outcomes. They also discuss challenges such as limited data availability and variability in study quality. This work supports survey-based research by encouraging the collection of real-world data from developers to improve software practices.

[3] Robert Feldt and Andreas Magazinius (2010) analyzed common threats to validity in empirical software engineering research. They categorized these threats into internal, external, construct, and conclusion validity, explaining how each can affect study results. For survey research, they pointed out issues such as biased questions, small sample sizes, and misinterpretation of responses. Their work provides practical recommendations for reducing these risks and improving the accuracy of survey findings.

[4] Martin Höst, Björn Regnell, and Claes Wohlin (2000) investigated whether students can be used as valid participants in software engineering studies instead of professionals. They compared results obtained from both groups and found differences in experience, decision-making, and outcomes. This study is important for survey-based research because it helps researchers decide on appropriate participant groups and understand how participant characteristics can influence results.

[5] Claes Wohlin et al. (2012) authored a widely recognized book on experimentation in software engineering. The book covers research planning, hypothesis formulation, variable selection, data collection methods, and statistical analysis. It also discusses ethical considerations and reporting standards. Although focused on experiments, many of its principles are directly applicable to survey-based research, especially in designing studies and analyzing collected data.

[6] Dag I. K. Sjøberg et al. (2005) conducted a comprehensive survey of controlled experiments in software engineering. Their study reviewed numerous experiments to identify trends, methodologies, and common practices. They highlighted

DOI: 10.17148/IJARCCE.2026.154307

issues such as limited sample sizes and lack of replication. This research provides valuable insights into how empirical studies, including surveys, are conducted and how their quality can be improved.

[7] Forrest Shull, Janice Singer, and Sjøberg (2008) contributed to the field through their work on advanced empirical software engineering methods. They discussed different research approaches such as surveys, case studies, and experiments, explaining when and how to use each method effectively. Their work emphasizes combining multiple research methods to obtain more reliable and comprehensive results.

[8] Jonas Linåker et al. (2016) provided detailed guidelines specifically for conducting surveys in software engineering. They covered aspects such as designing clear and unbiased questionnaires, selecting representative samples, and ensuring high response rates. The authors also discussed data analysis techniques and reporting standards. This paper is particularly important because it directly addresses best practices for survey-based research.

[9] Per Runeson and Höst (2009) presented guidelines for conducting case study research in software engineering. They described how to design case studies, collect qualitative and quantitative data, and analyze findings. Although their focus is on case studies, the principles they provide—such as data validation and triangulation—are highly relevant to survey research as well.

[10] Natalia Juristo and Ana M. Moreno (2013) explained the fundamentals of software engineering experimentation. Their work includes detailed discussions on experimental design, data collection, statistical testing, and result interpretation. These concepts are essential for survey-based research, especially when analyzing survey data and drawing valid conclusions.

[11] Tim Menzies and Martin Shepperd (2012) emphasized the importance of repeatability and reproducibility in empirical research. They argued that research findings should be consistent when studies are repeated under similar conditions. In survey-based research, this means designing clear methodologies and documenting processes so that others can replicate the study and verify results.

[12] Barbara A. Kitchenham et al. (2002) proposed preliminary guidelines for conducting empirical research in software engineering. Their work outlines best practices for planning, executing, and reporting studies. They stress the importance of clear research questions, proper data collection methods, and accurate reporting. These guidelines form a strong foundation for conducting high-quality survey-based research.

III. CONCLUSION

Survey-based research is very important in software engineering because it helps us understand how things actually work in the industry. It gives insights into the challenges developers face, the practices they follow, and the new trends that are emerging. To get useful results, it is essential to design surveys properly, choose the right data collection methods, and apply suitable statistical analysis techniques.

The findings show that surveys are commonly used to collect real-world data and support improvements in software development processes. In the future, this type of research can be improved by making surveys more reliable, involving a more diverse group of participants, and using automated tools to make data analysis easier and more accurate.

REFERENCES

[1]. B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE Technical Report, Keele University, 2007.
[2]. T. Dybå, B. A. Kitchenham, and M. Jørgensen, "Evidence-based software engineering for practitioners," IEEE Software, vol. 22, no. 1, pp. 58–65, 2005.
[3]. R. Feldt and A. Magazinius, "Validity threats in empirical software engineering research—An initial survey," in Proc. ESEM, 2010, pp. 374–379.
[4]. M. Höst, B. Regnell, and C. Wohlin, "Using students as subjects—A comparative study of students and professionals in lead-time impact assessment," Empirical Software Engineering, vol. 5, no. 3, pp. 201–214, 2000.
[5]. C. Wohlin et al., *Experimentation in Software Engineering*, Springer, 2012.
[6]. D. Sjøberg et al., "A survey of controlled experiments in software engineering," IEEE Transactions on Software Engineering, vol. 31, no. 9, pp. 733–753, 2005.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154307

[7]. F. Shull, J. Singer, and D. I. K. Sjøberg, *Guide to Advanced Empirical Software Engineering*, Springer, 2008.
[8]. J. Linåker, S. Sulaman, R. M. de Mello, and M. Höst, "Guidelines for conducting surveys in software engineering," Empirical Software Engineering, vol. 21, no. 5, pp. 1421–1453, 2016.
[9]. P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," Empirical Software Engineering, vol. 14, no. 2, pp. 131–164, 2009.
[10]. N. Juristo and A. M. Moreno, *Basics of Software Engineering Experimentation*, Springer, 2013.
[11]. T. Menzies and M. Shepperd, "Special issue on repeatability in empirical software engineering," Empirical Software Engineering, 2012.
[12]. B. Kitchenham et al., "Preliminary guidelines for empirical research in software engineering," IEEE Transactions on Software Engineering, vol. 28, no. 8, pp. 721–734, 2002.

