Research Motivation

Over the past few years, emerging technologies have had a significant impact on the processes of software engineering (SE), shifting the focus to a data-driven decision-making approach. This approach relies on data analysis to make accurate and rapid decisions, rather than relying solely on human experience-based intuition. However, although data-driven approaches utilize advanced techniques such as Machine Learning (ML) and Natural Language Processing (NLP) to derive insights from data, the utility of the models used in these methods is heavily reliant on the quality of the data used in building them, which makes Data Quality (DQ) a critical success factor [1-2]. In other words, even though automation is transforming software engineering processes, human interpretation is still necessary to assess and verify the quality of the data – what we can call "fitness for use" [2-3].

Poor Data Quality (DQ) can have a substantial impact both socially and economically. From a software engineering point of view, bad DQ can lead to huge amounts of reworking [2, 4-5]. This may lead to poor results and substantial technical debt that needs to be repaid later. Unfortunately, based on the literature, the software engineering view of DQ is still limited and it is often neglected – despite its importance [2, 6-8]. Studies point out that this attitude towards DQ can be linked to lack of motivation and the inadequacy of skills in identifying DQ issues that may affect the outcomes.

In this research, we address this problem using the concept of digital nudging for motivation and behaviour change. Nudge theory has shown potential in modifying human behaviour, generally, and in encouraging and promoting software engineering best practices, specifically. Our aim in this research is to evaluate the effectiveness of using digital nudges to encourage software engineers to review and address data quality issues, in order to make better decisions. We designed a nudge-based framework for DQ to promote software engineers' motivation and behaviour towards DQ. The framework was developed based on a narrative literature review and previous empirical insights. It provides guidelines on how to design effective digital nudges to change software engineers' behaviour towards DQ.

In order to evaluate the effectiveness of the framework, we are using two evaluation methods. The first one is conducted through this series of expert feedback which has been identified as an effective method to gather early feedback [9]. The second approach focuses on demonstrating the utility of our framework. This involves implementing the designed artifact in an actual context, allowing for real-world testing and assessment which is through controlled experiments.

Your Contribution

As an expert, your feedback plays a crucial role in strengthening the framework. By leveraging your expertise, we can refine its components, address potential gaps, and ensure it remains both rigorous and practically relevant. Your insights directly contribute to shaping a more robust and applicable tool to advance research and practice in data quality and software engineering.

The feedback process will take place in two stages: first through an evaluation survey, followed by a short interview. All relevant materials, including the framework description and survey, are attached.

Information from the session will be kept confidential. Your data will have a number associated with it. After the research is completed, we may save the notes for future use by ourselves or others, but your name will not be included.

References

[1] Bosu, M.F., MacDonell, S.G. (2013) Data quality in empirical software engineering: a targeted review, in Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering (EASE2013). Porto de Galinhas, Brazil, ACM Press, pp.171-176.

[2] Franch, X. (2021). Data-Driven Requirements Engineering: A Guided Tour. Communications in Computer and Information Science, pp.83–105. doi:10.1007/978-3-030-70006-5_4.

[3] Dobyns, L. and Crawford-Mason, C. (1991). Quality or Else: The Revolution in World Business. Organizational Dynamics, vol. 21, no. 2, pp. 72-73.

[4] Bobrowski, M., Marr., M., Yankelevich, D. (2000). A Software Engineering View of Data Quality.

[5] Loshin, D. (2001). Enterprise knowledge management : the data quality approach. San Diego, Calif.: Morgan Kaufmann.

[6] Shepperd, M. (2011). Data quality. Proceeding of the 2nd international workshop on Emerging trends in software metrics - WETSoM ’11. doi:10.1145/1985374.1985376.

[7] Liebchen, G.A. and Shepperd, M. (2008). Data sets and data quality in software engineering. Proceedings of the 4th international workshop on Predictor models in software engineering -PROMISE ’08. doi:10.1145/1370788.1370799.

[8] Liebchen, G. (2010). Data cleaning techniques for software engineering data sets. Doctoral thesis, Brunel University.

[9] Baskerville, R., Pries-Heje, J., Venable J. (2009). Soft design science methodology. in Proc. 4th Int. Conf. Des. Sci. Res. Inf. Syst. Technol., (DESRIST). New York, NY, USA. doi: 10.1145/1555619.1555631.