Presentation abstract: A first case study about key socio-technical software indicators at Forem

Nicolas Riquet
Forem, Belgium
University of Namur
Namur, Belgium
nicolas.riquet@unamur.be

Xavier Devroey
Namur Digital Institute
University of Namur
Namur, Belgium
xavier.devroey@unamur.be

Benoît Vanderose
Namur Digital Institute
University of Namur
Namur, Belgium
benoit.vanderose@unamur.be

Identifying software maintainability risks is hard yet essential for large organizations like Forem. Such risks are, by nature, socio-technical, but the technical and social parts are usually considered separately, preventing managers from having a clear and complete view of the situation. More specifically, this presentation focuses on our recent study on the potential impact of staff turnover on code complexity. We performed a historical case study at Forem, the Public Service for Employment and Vocational Training in Wallonia (Belgium), counting 92 developers, including several external consultants hired to work fixed-term periods on specific projects.

We developed GITDELVER [1], a tool relying on Pydriller [2], to mine 101 software repositories written in various programming languages, and assess the relevance of applying a data-driven approach to help identify and manage socio-technical risks in Forem's codebases. We produced a dataset containing information about the evolution of 67,666 unique source files (280,430 modifications) and their complexity metrics. Such metrics can help identify (i) bad coding practices and design mistakes (increasing technical debt [3], [4]), and (ii) developers in need of coaching or teams rushing to meet a deadline [5] (increasing the social debt [6]). We further extended the dataset with information about developer leaving dates, allowing us to determine for each file the number of authors still working at Forem. Figure 1 shows an overview of the dataset generation pipeline. The mining process took 23 hours on a 2.50GHz Intel Core i5-7200U CPU with 8GB RAM. The manual steps involved in the data preparation and analyses took five additional days.

Our results show that 74.12% of source files never had more than one author, 20.83% are orphan files (no author remaining in the organization) and 58.43% are near-orphans (one author remaining). Pearson's correlation values reported in Table I show that there is no correlation (i.e., values are close to 0) between authorship and code complexity in the codebases, potentially denoting good development practices currently in place at Forem. We considered only source files authored by more than one developer (83,906 file modifications) to avoid noises introduced by single-author files.

Finally, we conducted interviews with seven developers from six teams to compare our findings to their perceptions.

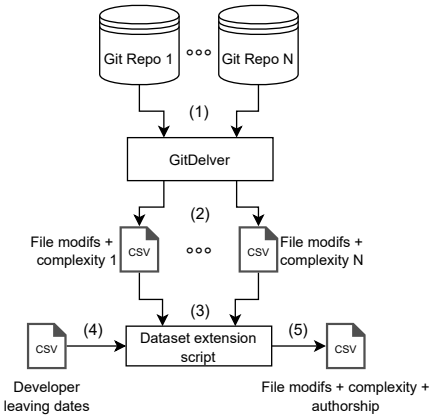

Fig. 1. Dataset generation pipeline

TABLE I
PEARSON'S CORRELATION COEFFICIENTS

	Total authors	Remaining authors
Methods count	-0.001306	0.006870
NLOC	0.000443	0.009722
Cyclomatic complexity	-0.000935	0.002575
NLOC / Methods count	0.000524	0.009334
Complexity / Methods count	0.010188	-0.110847

Our results show that it is tough for developers to view the authorship situation in their codebases accurately. They also rightly think that authorship does not affect code complexity over time at Forem.

REFERENCES

[1] N. Riquet. GitDelver. https://github.com/nicolasriquet/GitDelver.
[2] D. Spadini, M. Aniche, and A. Bacchelli, "Pydriller: Python framework for mining software repositories," in *ESEC/FSE '18*, 2018, pp. 908–911.
[3] N. Ernst, R. Kazman, and J. Delange, *Technical Debt in Practice*. The MIT Press, 2021.
[4] P. Kruchten, R. L. Nord, and I. Ozkaya, "Technical debt: From metaphor to theory and practice," *IEEE Software*, vol. 29, no. 6, pp. 18–21, 2012.
[5] B. Bafandeh Mayvan, A. Rasoolzadegan, and A. Javan Jafari, "Bad smell detection using quality metrics and refactoring opportunities," *Journal of Software: Evolution and Process*, vol. 32, no. 8, p. e2255, 2020.
[6] D. A. Tamburri, P. Kruchten, P. Lago, and H. van Vliet, "Social debt in software engineering: insights from industry," *Journal of Internet Services and Applications*, vol. 6, no. 1, p. 10, dec 2015.