Comparing the intensity of variability changes in software product line evolution (Summary)

Christian Kröher[1] Lea Gerling[2] Klaus Schmid[3]

Abstract: *This paper summarizes [KGS23], published in the Journal of Systems & Software.*

Software Product Line (SPL) evolution affects multiple kinds of artifacts like code, build, and variability model. Only part of these artifacts are relevant to variability. We studied the intensity (frequency and amount) in which developers change variability information in our earlier work. Applying this to the Linux kernel revealed that changes to variability information occur infrequently and only affect small parts of the analyzed artifacts. Here, we extend our previous work to a comparative evolution analysis of four SPLs. We provide a detailed analysis of the individual intensities of variability changes by applying our updated approach to each SPL. A comparison of these results confirms our findings of infrequent and small changes to variability information in our previous study.

Keywords: Software product line evolution; Evolution analysis; Variability changes; Change Intensity

1 Summary

The evolution of a Software Product Line (SPL) typically affects a variety of artifact types, like source code, build system, and variability model artifacts. For each type, changes can be differentiated into explicit changes to *variability information* and changes to *artifact-specific information*, i.e., which do not affect variability.

In our previous work, we already studied product line evolution from this perspective, using the Linux-kernel as an example [KGS18]. There, we introduced a fine-grained approach to identify the *intensity* (frequency and amount) of changes to code, build, and variability model artifacts. The application of these processes to the Linux kernel revealed that changes to variability information occur infrequently and only affect small parts of the analyzed artifacts. These results built the foundation for our approach to incremental SPL verification [Kr22], which allowed to significantly accelerate dead code analysis.

While these results were very promising, it was unclear whether our finding that the rate of variability change is rather low would be specific to Linux or whether this would generalize

[1] University of Hildesheim, Institute of Computer Science, Software Systems Engineering, Universitätsplatz 1, 31141 Hildesheim, Germany, kroeher@sse.uni-hildesheim.de
[2] University of Hildesheim, Institute of Computer Science, Software Systems Engineering, Universitätsplatz 1, 31141 Hildesheim, Germany, gerling@sse.uni-hildesheim.de
[3] University of Hildesheim, Institute of Computer Science, Software Systems Engineering, Universitätsplatz 1, 31141 Hildesheim, Germany, schmid@sse.uni-hildesheim.de

doi:10.18420/sw2024_24

to other product lines. Hence, in [KGS23] we extended this analysis to three additional SPLs: the coreboot firmware, the BusyBox UNIX utilities, and axTLS embedded SSL besides the Linux kernel. In this paper [KGS23]

- we describe an updated version of our approach for the automated extraction and analysis of commits;
- we reveal and analyze the evolution of artifact-specific and variability information in code, build, and variability model artifacts in four SPLs;
- we compare and discuss this evolution data based to provide a broader understanding on SPL evolution.

The extension of our analysis reveals that differences among the analyzed SPLs exist: The frequency of variability changes varies. However, our analysis also confirms that in general, changes to variability information occur infrequently and only affect small parts of the analyzed artifact types. Consequently, our earlier results in the context of the Linux-kernel can be reproduced and extend to other cases.

2 Data Availability

The journal paper is available at DOI https://doi.org/10.1016/j.jss.2023.111737. The implementation is available on GitHub: `https://github.com/CommitAnalysisInfrastructure/`, while the data is available at `https://doi.org/10.5281/zenodo.7273340`.

Acknowledgments

The original work was partially supported by the Evoline project, funded by the DFG (German Research Foundation) under Priority Programme SPP 1593 and by the ITEA3 project REVaMP^2 funded by the BMBF (German Ministry of Research and Education) under grant 01IS16042H. Any opinions expressed herein are solely by the authors and not of the DFG or BMBF.

References

[KGS18] Kröher, C.; Gerling, L.; Schmid, K.: Identifying the Intensity of Variability Changes in Software Product Line Evolution. In: 22nd International Systems and Software Product Line Conference (SPLC'18). Pp. 54–64, 2018, DOI: 10.1145/3233027.3233032.

[KGS23] Kröher, C.; Gerling, L.; Schmid, K.: Comparing the Intensity of Variability Changes in Software Product Line Evolution. Journal of Systems and Software 203 (C), 2023, DOI: 10.1016/j.jss.2023.111737.

[Kr22] Kröher, C.; Flöter, M.; Gerling, L.; Schmid, K.: Incremental Software Product Line Verification - A Performance Analysis with Dead Variable Code. Empirical Software Engineering 27 (68), pp. 1–41, 2022, DOI: 10.1007/s10664-021-10090-6.