Reverse Engineering Feature-Aware Commits from Software Product-Line Repositories

Lukas Bormann, 12.09.2022

Advisor: M.Sc. Paul Maximilian Bittner

Reviewer: Prof. Dr. Thomas Thüm

1 Introduction

Version control systems (VCS) such as Git and Subversion are widely used both in open-source projects and in the industry. VCSs are designed to track changes made to the source code and group these changes chronologically in commits [MDR18].

Even tough software often has to allow for variability to be applicable in different environments, commonly used VCSs do not provide any functionality to handle variability in a commit. If a developer wants to revert a change to a feature, which can be achieved by reverting one or multiple commits, the developer has to revert all changes that were committed and handle the unwanted reverts manually, even though some changes might not affect the feature at all and are just part of the commits.

Existing post-commit and commit-visualizing assistants, for example split commits into different developer activities [She+21] or identify uncommitted features and regroup them into separate branches [Zho+18], do not address variability in commits.

Thus we propose an operator which addresses variability in commits. In this context our operator refers to a process, that converts a commit into a feature-aware commit. A feature-aware commit is a commit, that is reduced to a commit that only affects a single feature, it can be used to restructure and recommit changes to sort commits by features. Our operator relies on a transformation of the committed source code into a graph-based view [Vie21] and turning this graph-based view into feature-aware commits. This leads to increased readability of variability in a repository.

In the long term we aim to construct a visual interface that shows specific views of features, to give a developer insight into the progress of a certain feature in the project. Additional research we want to investigate it the possibility of chronological ordering of feature-aware commits, in addition to ordering by feature.

2 Problem Statement

VCSs can be used as intended, with small and fine-grained commits, about a certain feature. Yet in practice commits are rather large with many loosely related changes across features. Thus we propose the decomposition into feature-aware commits

3 Contribution

The goal of this thesis is to build an operator that allows for traceability of feature evolution in the commit history. To create this operator we will evaluate existing methods based on their capabilities to trace the evolution of features. Based on these findings, we will implement feature traceability with the VariantSync tree. The resulting operator allows commits to be broken down into feature-aware commits.

3.1 Work Packages

1. **Literature Research**: We build a sufficient knowledge base in Variation Trees, other possible representations of commits, and procedures that provide a better understanding of commits.
2. **Concept**: We develop an operator on how to restructure large commits to be feature-aware.
3. **Initial Prototype**: We implement our operator to make commits feature-aware.
4. **Empirical Evaluation**: We perform a qualitative comparison with similar programs and our operator. We collect statistics about the capability of our operator to break up arbitrary-sized commits into feature-aware commits and the achieved fine granularity of the feature-aware commits.
5. **Writing the Thesis**: We write the thesis

3.2 Schedule

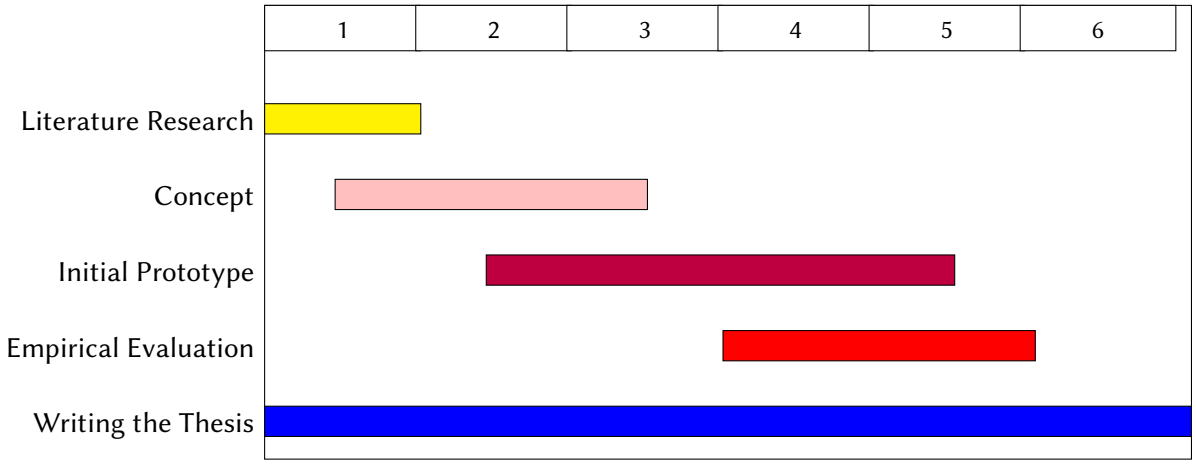

Figure 1: Thesis time schedule in months

References

[MDR18] Ward Muylaert and Coen De Roover. "[Research Paper] Untangling Composite Commits Using Program Slicing". In: *2018 IEEE 18th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. 2018, pp. 193–202. DOI: 10.1109/SCAM.2018.00030 (see Page 1).

[Zho+18] Shurui Zhou et al. "Identifying features in forks". In: *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. IEEE. 2018, pp. 105–116 (see Page 1).

[She+21] Bo Shen et al. "SmartCommit: a graph-based interactive assistant for activity-oriented commits". In: *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2021, pp. 379–390 (see Page 1).

[Vie21] Sören Viegener. "Empirical Evaluation of Feature Trace Recording on the Edit History of Marlin". Bachelor's Thesis. Germany: University of Ulm, Apr. 2021. DOI: 10.18725/OPARU-38603. URL: https://oparu.uni-ulm.de/xmlui/bitstream/handle/123456789/38679/BA_Viegener.pdf (see Page 1).