1 Extracting Software Product Lines

Software product lines (SPL) are a development approach to systematically manage and reuse software artefacts. In particular, different variants of the same product can be configured and, thus, are customized to specific needs. While SPLs promise several benefits over other reuse approaches, they are still not common in practice. There, software is often copied and only afterwards adapted to new requirements, the clone-and-own approach. This way, several independent systems, which share a common code base, emerge. Maintaining and managing those systems is an expensive task and SPLs are normally better suited for this. For this reason, companies may decide to change their development approach and extract a SPL from its legacy systems. The following projects focus on topics of this migration process.

1.1 Implementing Feature Location in FeatureIDE

Type of Project: Software-Project, Bachelor’s Thesis
Status: Open
Language: English, German

FeatureIDE supports developers in implementing feature-oriented SPLs. It is an Eclipse plug-in that provides several implementation techniques, for example, feature- and aspect-oriented programming. FeatureIDE is permanently extended with new functionality. However, the analysis of existing products is rarely supported. Feature location describes the process of identifying and mapping features in source code. The result of those techniques are possible feature candidates. Afterwards, developers have to manually select and scope those. Appropriate tool support is a helpful means for extracting a SPL from existing variants. Still, FeatureIDE does not provide such a tool.

The goal of this project is to develop a plug-in for FeatureIDE that extends its functionality with feature location. Therefore, the following tasks are to be done.

- Analyze literature on existing feature location techniques (for multiple variants).
- Propose and implement a feature location approach in FeatureIDE.
- Evaluate the plug-in on an open-source project.

1.2 Extract-Method Refactoring

Type of Project: Software-Project
Status: Open
Language: English, German
There are several refactorings to improve the quality of source code. Extract-method is used to separate parts of large methods. Thus, it can solve design flaws, for example, duplicated code, feature envy, or long method. In addition, breaking long methods down to smaller parts can improve understanding and maintaining of the code. However, in the context of SPLs the extraction can be more complex. Feature code and interactions must be handled by the developer. Tool support for this task may help to extract the code but also to highlight possible problems.

Goal of this project is to implement support for an extract-method refactoring for FeatureIDE. Thus, the following steps must be done:

- Assess possible conflicts that can occur during the extract-method refactoring.
- Implement semi-automatic tool support.
- Evaluate your implementation.

1.3 Mapping Features to Code in Multiple Variants

Type of Project: Scientific Project, Master’s Thesis
Status: Open
Language: English, German

Locating and mapping features to their code representatives (a.k.a. variability mining) is an important task in analysing legacy systems and SPL development. For this project we aim to analyse multiple cloned variants. At first, features are located in a single variant. The result of this step is a feature model that is linked with the according code fragments. Afterwards, this models shall be mapped and extended to the remaining variants. Finally, it includes the all features of the program family. This can be done by analysing code clones and, thus, mapping found features to the code of additional variants. Still, we do not only want to identify the code we know about but simultaneously mine for additional seeds for existing and also new features.

The goal of this project is to develop a concept and support Following tasks are to be done:

- Analyse literature on feature location, variability mining, and code clones.
- Develop an semi-automatic approach to support the described process.
- Implement necessary tool support. You may use FeatureIDE, which already supports some of the above tasks (e.g. code-clone detection).
- Evaluate your approach on a case study.

2 Literature Analysis

Literature analysis, reviews, and surveys are a first step to get familiar with a new scientific topic. Several management tools for literature exist and support the search process. They
can access several publisher or search engines (e.g. Google Scholar, ACM digital library, Scopus, ...). Still, a literature review remains a challenging task and requires a lot of manual work. The following projects aim to support this process with tools.

2.1 Step-wise Refinement and Mining of Search Results

Type of Project: Scientific Project, Master’s Thesis
Status: Open
Language: English, German

After each scientific search, a researcher owns a list of possibly relevant literature. However, he still has to assess content and quality of all articles. The idea of this project is to develop a tool to support this evaluation. Thus, a researcher can faster identify relevant and important work on the topic and simplify reviews. We propose a process as follows. A researcher provides a list of possibly relevant literature. He can define quality criteria (e.g. publication type, publisher, ...) and also rate the importance of articles. An algorithm shall provide a learner/recommender that assesses the remaining articles (e.g. based on quality criteria, keywords, title, ...).

Finally, the goal is to develop and implement an approach that supports the evaluation of literature quality and identification of the most relevant articles. Therefore, the following tasks are part of this project:

- Analyse literature on literature surveys and data mining/recommender systems.
- Propose quality criteria, assessments, and a suited learner.
- Develop and implement the required tool.
- Evaluate your approach on a suited data basis.

3 Bring your own topic

Besides the presented topics above, I encourage students to propose their own topics, just contact me:

jkrueger@ovgu.de

Jacob Krüger