Multi-Language Software Metrics

Gil Dinis Magalhães Teixeira { gil.teixeira@fe.up.pt }

Abstract. Software metrics are of utmost significance in software engineering. They attempt to capture the properties of programs, which developers can use to gain objective, reproducible and quantifiable measurements. These measurements can be applied in quality assurance, code debugging, and performance optimization. The relevance of their use is continuously increasing with software systems growing in complexity, size, and importance. Implementing metrics is a tiring and challenging task that must be repeated for every language used. However, suppose we can decompose the implementation of software metrics into a sequence of basic queries to the source code that can be implemented in a language-independent way. In that case, these metrics can be applied to any language. Many programming languages share similar concepts (e.g., functions, loops, branches), which has led certain approaches to raise the abstraction level of source-code analysis and support multiple languages. LARA is a framework developed in FEUP (Faculty of Engineering of the University of Porto) for code analysis and transformation that is agnostic to the language being analyzed. This means that analysis strategies written in LARA have the potential of being applied to the source code of multiple languages. A study is conducted to identify the benefits and drawbacks of software metrics as well as to select and analyze relevant and commonly used software metrics. This will provide the foundations for the implementation of software metrics in a language-independent way by using the LARA framework. In order to be able to perform a multi-language analysis, we created a model on top of each compiler's AST, where every node is mapped to a common join point. This results in a virtual AST with the same join points for each language. Using this model, we also created a library of metrics that can be applied to multiple languages. To validate the developed framework, we performed an internal and external validation. The internal validation, by comparing a projected implemented in different languages, ensures that the metrics results are consistent. In contrast, the external validation compares several tools' results and performance across multiple open source projects. Even though our approach underperforms in terms of execution time, it presents a flexible and straightforward approach to analyze four different languages (C, C++, Java, JavaScript) and can calculate object-oriented, complexity, and size metrics.

Supervisor: Prof. João Carlos Viegas Martins Bispo, PhD; **Second Supervisor:** Prof. Filipe Alexandre Pais de Figueiredo Correia, PhD.

Keywords: Metrics. Compilers. Software Engineering. Software Analysis.

ACM Computing Classification System:

- Software and its engineering → Software notations and tools → General programming languages → Language features
- General and reference → Cross-computing tools and techniques → Metrics