Automatically Fixing Static Analysis Tools Violations

Extended Abstract

Diego Marcilio[1,2], **Rodrigo Bonifácio**[2]

[1]USI Università della Svizzera italiana, Lugano, Switzerland

[2]Computer Science Department, University of Brasília (UnB), Brasília, Brazil

Abstract. *Static analysis tools analyze source code to find deviations, or violations, from recommended programming practices defined as rules. A warning is raised when a piece of code violates any rule. Even though these tools can help to identify defects, developers still face several barriers when using them. Among the challenges are the significant number of reported warnings, often caused by false-positives, and the need to devise fixes, a repetitive and error-prone process. In this work, we addressed these two difficulties in two stages: 1) we identified which kind of rules are mostly fixed by Java developers when using SonarQube (a widely used static analysis tools); 2) we implemented a tool that provides automatic fixes for a subset of the previously commonly fixed found rules. The results obtained indicate that providing automatic fixes for commonly fixed warnings is feasible and welcomed by developers.*

Resumo. *Ferramentas de análise estática analisam código-fonte definem regras para encontrar desvios de práticas recomendadas de programação. Um alerta é lançado quando um trecho de código viola uma ou mais regras. Esta análise pode encontrar defeitos de* software *de forma antecipada, No entanto, programadores encontram diversas dificuldades na adoção dessas ferramentas. Dentre os principais desafios estão o alto número de alertas, comumente causado por falso-positivos, e a necessidade de se idealizar e implementar uma correção, o que pode ser um processo repetitivo e passível de erro. Nesse trabalho, nós abordamos essas duas dificuldades em duas etapas: 1) nós identificamos quais regras são mais corrigidas ao analisar projetos Java que usam a ferramenta SonarQube (dentre as mais utilizadas no contexto de projetos Java); 2) nós implementamos uma ferramenta que fornece correções automáticas para um subconjunto dessas regras comumente corrigidas. Nossos resultados indicam que prover correções automáticas não é só viável, como é bem recebido por desenvolvedores.*

Static code analysis tools (SATs) are well known for detecting possible sources of defects earlier in the software development process. They provide ways to detect problems quickly – bugs, bad practices, and code uncompliant to style choices – in particular because SATs do not need to run programs: they work *statically* on the source or byte code. Although having the potential to detect a vast and diverse set of problems swiftly, SATs are infamously known for being *imprecise*. It is not uncommon to observe thousands of warnings being reported when these tools execute in a project for the first time. Moreover, understanding the reported problems and coming up with a proper fix is often a nontrivial task.

The research that comprehends the first author's master thesis [Marcilio et al. 2019a, Marcilio et al. 2019b, Marcilio et al. 2020] aimed to improve the practical usability of SATs by automatically providing *fix suggestions*: source code modifications that make it compliant with the rules checked by the analysis tools.

First [Marcilio et al. 2019a], we investigated SonarQube's usage on hundreds of systems from two large open-source foundations (Apache Software Foundation and Eclipse Foundation) and two Brazilian Federal Government institutions. We also surveyed 18 developers of the analyzed projects by asking 6 closed questions. Our analyses showed that only 8.76% of the violations were fixed, even though practitioners find SATs' reports relevant—rejecting pull-requests or even postponing releases based on its outcomes. We found that 20% of the rules – from the almost 422 thousand violations we mined – correspond to 80% of the fixes.

We then developed SpongeBugs [Marcilio et al. 2019b, Marcilio et al. 2020] on the assumption that frequently fixed violations are more likely to correspond to real issues of practical relevance. We selected 11 rules among the most widely used from a combined dataset of 1,033 projects with 18 million violations. SpongeBugs can fix violations in a completely automatic, precise, and scalable manner. Even though SpongeBugs leverages lightweight transformations techniques to achieve its precision and effectiveness, some of the rules involve intricate modifications, such as altering multiple lines and introducing fields and variables. We submitted 946 fixes spanning 38 pull-requests to various well-established open-source projects, including the Eclipse IDE, SonarQube, and SpotBugs. Project maintainers accepted 825 (87%) of those fixes—most of them (97%) without modifications. SpongeBugs removed 85% of all violations ($\sim$7.5K) of the considered rules in the analyzed projects. All of its fixes compiled successfully, and only 0.6% of them were identified as false positives. SpongeBugs is also scalable to large projects, as it can run in under 10 minutes on projects as large as half a million lines of code.

Providing developers with automatic fixes for SATs' violations can be feasible and well-received. By automatically fixing violations that frequently appear, we can free developers to focus on more urgent and complex tasks, such as developing new features and fixing bugs.

References

Marcilio, D., Bonifácio, R., Monteiro, E., Canedo, E., Luz, W., and Pinto, G. (2019a). Are static analysis violations really fixed?: A closer look at realistic usage of SonarQube. In *Proceedings of the 27th International Conference on Program Comprehension*, ICPC '19, pages 209–219, Piscataway, NJ, USA. IEEE Press.

Marcilio, D., Furia, C. A., Bonifácio, R., and Pinto, G. (2019b). Automatically generating fix suggestions in response to static code analysis warnings. In *19th International Working Conference on Source Code Analysis and Manipulation, SCAM 2019, Cleveland, OH, USA, September 30 - October 1, 2019*, pages 34–44. IEEE.

Marcilio, D., Furia, C. A., Bonifácio, R., and Pinto, G. (2020). Spongebugs: Automatically generating fix suggestions in response to static code analysis warnings. *Journal of Systems and Software*.