Homework 6 HT2023

Casper Kristiansson

November 7, 2023

1 A technical summary

The article ”A Longitudinal Analysis of Bloated Java Dependencies” [1] investigates software bloating specifically bloated dependencies in Java projects [2]. It specifically examines and studies the growth and persistence of bloated dependencies in Maven-managed projects via Github. The reason for doing this research is to highlight the importance of managing bloated dependencies with the help of software tools.

The authors were able to collect data on the history of 31,515 Maven dependency trees from 435 open-source projects. The analysis of this data showed that the majority of bloated dependencies stayed bloated over time (86%). This shows that even if a dependent one was bloated it will most likely remain bloated over time. Therefore the authors highlight the importance of using software tools like DepClean that perform debloating of depencies.

The study was able to highlight the issue with software bloating and how software can solve bloating dependencies issues and better maintain software. Previous studies around this subject often just considered the issue of software bloating and not how software bloating is affected long-term. Analyzing the software bloating over a long-term period and understanding its growth in Java projects highlights the importance of the issue.

The findings in the study highlight the importance of managing dependencies in projects. It shows that dependencies that are bloated will often remain bloated over time. The study mentions tools like DepClean to assist developers in identifying unnecessary dependencies to help with the maintenance of software. Using this strategy will help with dependency management which will lead to an immediate reduction footprint of the software which helps in long-term maintenance challenges and security risks associated with bloated dependencies.

References

[1] Soto-Valero, C., Durieux, T., and Baudry, B. A longitudinal analysis of bloated java dependencies. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foun-*

dations of Software Engineering (New York, NY, USA, 2021), ESEC/FSE 2021, Association for Computing Machinery, p. 1021–1031.

[2] Soto-Valero, C., Harrand, N., Monperrus, M., and Baudry, B. A comprehensive study of bloated dependencies in the maven ecosystem. *Empirical Software Engineering 26*, 3 (2021), 45.

2 ChatGPT Improvements

2.1 ChatGPT Improved Text

The study ”A Longitudinal Analysis of Bloated Java Dependencies” delves into the prevalence and persistence of bloated dependencies within Java/Maven projects. The research is motivated by the need to understand and mitigate the impacts of bloated dependencies—superfluous third-party libraries included in applications that are not necessary for their execution. The primary goal is to underscore the benefits of debloating tools, such as DepClean, in managing these dependencies effectively.

Through a rigorous methodology, the authors scrutinized the historical data of 435 Java projects, encompassing 31,515 versions of Maven dependency trees. The findings were stark, revealing a consistent trend: 89.2% of direct dependencies identified as bloated remained so in subsequent versions. This pattern underscores a significant insight—that once a dependency becomes bloated, it is likely to stay that way, reinforcing the utility of proactive debloating.

Moreover, the study brings to light the unnecessary maintenance efforts expended on these bloated dependencies, including updates suggested by Dependabot—a popular dependency management bot. This observation spotlights the inefficiencies in current maintenance practices and the potential for optimization through debloating tools.

The authors’ analysis contributes a novel perspective by situating the problem of software bloat in the context of software evolution. Prior research typically addressed the immediate issues of bloat, whereas this study extends the discourse by examining its longitudinal effects. It suggests that bloat is not a transient issue but a persistent one that accumulates over time, affecting long-term maintenance and security.

In summary, the study by Soto-Valero et al. provides compelling evidence for the sustained nature of software bloat in Java projects and the consequent need for effective dependency management strategies. By highlighting tools like DepClean, the research advocates for a more streamlined approach to software maintenance, which could result in a reduced software footprint, alleviating long-term maintenance and security concerns.

2.2 Comparison

First off ChatGPT decided to remove all citations from the text even if they were inline (latex). Overall ChatGPT was able to drastically improve the language.

In some situations, the technical summary might have made a statement that didn't make a lot of sense like "even if a dependent one was bloated it will most likely remain bloated over time" which in this case was a typo of dependent which should have been dependency.

ChatGPT also chose to mention and discuss Dependabot which was not originally included in the technical summary. This is a good example where ChatGPT chose to add more information because it deemed it to be of interest.