

Homework 4 HT2023

Casper Kristiansson

October 12, 2023

Reading an article

For this assignment where the goal is to pick an article, I chose the paper "A Longitudinal Analysis of Bloated Java Dependencies" by César Soto-Valero, Thomas Durieux, and Benoit Baudry [7]. The paper can be found on Google Scholar where it has been posted by ACM dl Digital Library [1].

c) Medium of Publication

The publication medium of the paper was a conference proceeding which was called "Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering" [7]. The publication medium is in the field of Computer Science which specifically focuses on software engineering. The publication medium has good recognition and hosts a lot of events yearly [4]. As stated on their website "is an internationally renowned forum for researchers, practitioners, and educators to present and discuss the most recent innovations, trends, experiences, and challenges in the field of software engineering". The publication medium is a peer-reviewing method which is called **the double-blind review process** [3].

d) Purugganan and Hewitt Technique

1. Complete citation

Soto-Valero, C., Durieux, T., & Baudry, B. (2021). A Longitudinal Analysis of Bloated Java Dependencies. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 1021–1031). Association for Computing Machinery. <https://doi.org/10.1145/3468264.3468589>.

2. Key Words

dependencies, java, software bloat

3. General subject

Software Engineering

4. Specific subject

Bloated dependencies in a single software ecosystem: Java/Maven

5. Hypothesis

I couldn't find any specific hypothesis but the study does mention four different research questions that were used to guide the research of Bloated dependencies.

6. Methodology

The study uses different methodologies to tackle the different research questions. For example, question 1 used a global analysis of dependencies bloating and evaluated bloating trends in projects. Question 2 analyzes the evolution of dependencies which uses the same process by collecting the status of dependencies rather than representing it (in this case using a history of specific projects).

7. Result(s)

For research question 2 the objective was to investigate if software is identified as bloated at one point in time if it's going to remain so in the future. They found that the status of dependencies that indicate to be bloated will likely remain so. The findings they provided suggest areas of improvement in dependency management.

8. Summary of key points

- (a) Analyzed 31,515 dependencies among 435 Java projects.
- (b) Found that 89.2% of dependencies that were bloated remained so.
- (c) Bloated dependencies often came from new dependencies that weren't used
- (d) Suggests the usage of Dependancy to avoid unnecessary maintenance.

9. Context (how this article relates to other work in the field; how it ties in with key issues and findings by others, including yourself)

I think that this article has a lot of relevant work, especially in the software development industry. Managing dependencies on projects is always a hard task especially when a project grows over time.

10. Significance (to the field; in relation to your own work)

- (a) Highlights the issue of bloating
- (b) Maintenance efforts of dependencies
- (c) Dependency management

11. Important Figures and/or Tables (brief description; page number)

Figure 4 highlights "Overview of our data collection pipeline. From a set of 147,991 Java projects on GitHub, we analyze the usage status of the

dependencies in 435 Maven projects over time, to produce a dataset of 31,515 dependency trees”. which can be seen on page 1023. This shows the collection of data among the different Java projects.

e) Reference Article and Recent Citing Article

The article I chose to follow is ”In-depth investigation into debloating modern Java applications” [2]. The reason for selecting it is due to its relevance and focus on the same subject. A paper that cites my selected article recently is ”Studying and understanding the tradeoffs between generality and reduction in software debloating” [8]. This paper talks about the balance between maintaining software and achieving effective debloating.

f) IMRaD

The IMRaD format is based on a paper having the division and sections as an introduction, methodology, results, and discussion. My selected paper does follow IMRaD format where the paper has all of the required sections. But the paper does introduce a couple of new headers like related work etc. The second article I chose is ”Hierarchical text-conditional image generation with clip latent” [5]. This article to an extent uses the IMRaD format. The content uses that structure but it is much harder to get an understanding of it due to using other headers. It has an introduction and methodology sections but the rest of the paper discusses the actual content of it.

The IMRaD format could be more appropriate for specific sub-fields of computer science where specific areas it might not be as appropriate. For example, I believe IMRaD is used more where experimental studies are used.

g) Citation Count

Via Google Scholar my selected article has been referenced a total of 20 times [6].

h) Essence of the Chosen Article’s Contribution

The article conducted a study of software bloat in Java dependencies which revealed that out of 48,469 analyzed dependencies, 89.2% of dependencies that become bloated remain over time. The findings of the paper highlight the importance of the usage of debloating tools and give insight into the usage of automated dependency bots.

References

- [1] ACM. Acm digital library, 2023. Available: <https://dl.acm.org/> [Accessed: 12 October 2023].
- [2] BRUCE, B. R., ZHANG, T., ARORA, J., XU, G. H., AND KIM, M. Jshrink: In-depth investigation into debloating modern java applications. In *Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (New York, NY, USA, 2020), ESEC/FSE 2020, Association for Computing Machinery, pp. 135–146.
- [3] ESEC/FSE. Research papers track - esec/fse 2022, 2022. Available: <https://2022.esec-fse.org/track/fse-2022-research-papers> [Accessed: 12 October 2023].
- [4] ESEC/FSE. Acm joint european software engineering conference and symposium on the foundations of software engineering, 2023. Available: <https://www.esec-fse.org/> [Accessed: 12 October 2023].
- [5] RAMESH, A., DHARIWAL, P., NICHOL, A., CHU, C., AND CHEN, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.
- [6] SCHOLAR, G. Articles that cite the following paper, 2023. Available: https://scholar.google.com/scholar?cites=2867183705750282335&as_sdt=2005&sciodt=0,5&hl=en [Accessed: 12 October 2023].
- [7] 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 Foundations of Software Engineering* (New York, NY, USA, 2021), ESEC/FSE 2021, Association for Computing Machinery, p. 1021–1031.
- [8] XIN, Q., ZHANG, Q., AND ORSO, A. Studying and understanding the trade-offs between generality and reduction in software debloating. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering* (2022), pp. 1–13.