Universidad Carlos III de Madrid

SOFTWARE DEVELOPMENT

Bachelor's Degree in Computer Science & Engineering

GE3

Academic Year 2023/2024

Salvador Ayala Iglesias | 100495832 | 100495832@alumnos.uc3m.es | g89
Inés Fuai Guillén Peña | 100495752 | 100495752@alumnos.uc3m.es | g89
Kritika Pantha | 100531231 | 100531231@alumnos.uc3m.es | g89

TABLE OF CONTENTS

Article 1:

Title: Student Code Refactoring Misconceptions
Name of the conference: ITiCSE 2023: Innovation and Technology in Computer Science Education
Web address: https://dl.acm.org/doi/abs/10.1145/3587102.3588840
Authors: Eduardo Oliveira, Hieke Keuning, Johan Jeuring
Date of publication: 30 June 2023
Brief summary: This paper highlights the importance of teaching students about code quality through refactoring from the start. It identifies and categorizes common refactoring misconceptions, offering a foundational list of 25 such misconceptions along with a detailed website for further reference. Understanding and addressing refactoring misconceptions is vital for software development, affecting code quality and productivity. Fixing things early makes the code stronger and easier to change, which helps avoid problems and makes it easier for teams to work together.

Article 2:

Title: Behind the Intent of Extract Method Refactoring: A Systematic Literature Review
Name of the journal: IEEE Transactions on Software Engineering
Web address: https://ieeexplore.ieee.org/abstract/document/10381511
Authors: Eman Abdullah AlOmar, Mohamed Wiem Mkaouer, Ali Ouni
Date of publication: 04 January 2024
Brief summary: This paper reviews *Extract Method* refactoring, a vital aspect of software engineering for enhancing code quality. Through a systematic literature review, the study identifies 83 primary studies and highlights its significance in improving code quality, reducing duplication, and promoting modularity. By decomposing complex code sections into smaller, more manageable methods, it enhances readability and developer efficiency while supporting code analysis and optimization efforts. Overall, *Extract Method* refactoring is essential for ensuring software systems' maintainability, scalability, and agility.

Article 3:

Title: Semantic Code Refactoring for Abstract Data Types
Name of the journal: Proceedings of the ACM on Programming Languages
Web address: https://dl.acm.org/doi/abs/10.1145/3632870
Authors: Shankara Pailoor, Yuepeng Wang, Işıl Dillig
Date of publication: 05 January 2024
Brief summary: This paper presents an automated method for semantic code refactoring, specifically targeting modifications to the data representation of abstract data types (ADTs). Using counterexample-guided inductive synthesis (CEGIS), the method reduces the relational synthesis problem to programming-by-example problems and uses symbolic reasoning techniques based on logical abduction. Tested on 30 Java classes sourced from Github, the *Revamp* tool demonstrates its effectiveness in refactoring ADTs and re-implementing modified methods. Automating semantic code refactoring, especially for abstract data types (ADTs), is crucial in software development for efficiency, code quality, and agility. By automating repetitive tasks, developers save time and ensure consistency, improving maintainability and scalability.

Article 4:

Title: Fixing Your Own Smells: Adding a Mistake-Based Familiarisation Step When Teaching Code Refactoring
Name of the journal: Proc. SIGCSE'24, pages 1307-1313. ACM, 2024
Web address: https://arxiv.org/abs/2401.01011
Authors: Ivan Tan, Christopher M. Poskitt
Date of publication: 02 January 2024
Brief summary: The paper proposes a new method to teach refactoring to undergraduate computing students by incorporating exercises intentionally designed to create 'code smells', aiming to familiarize students with the code they create. A study with 35 novice undergraduates showed that this 'mistake-based' approach was significantly more effective and enhanced students' confidence compared to traditional teaching methods. This approach is crucial for software development as it familiarizes students with real-world scenarios and enhances problem-solving skills, contributing to cleaner, more maintainable codebases and improving overall development efficiency.

Article 5:

Title: The untold story of code refactoring customizations in practice
Name of the conference: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)
Web address: https://ieeexplore.ieee.org
Authors: Daniel Oliveira, Wesley K. G. Assunção, Alessandro Garcia, Ana Carla Bibiano, Márcio Ribeiro, Rohit Gheyi, Baldoino Fonseca
Date of publication: 20 May 2023
Brief summary: This paper studies refactoring in practice by developers, specifically looking at the custom modifications that they make on top of the standard refactoring tools provided by IDEs. It is important for developers to be able to skillfully use the tools provided by IDEs, such as PyCharm and VSCode, however, it is sometimes necessary to do manual refactoring. The research conducted an analysis of over 1,000 refactorings, which provide valuable insights to refactoring patterns that all developers (including those who work on IDEs) can learn from.

Article 6:

Title: Refactoring in Computational Notebooks
Name of the conference: ACM Transactions on Software Engineering and Methodology
Web address: https://doi.org/10.1145/3576036
Authors: Eric S. Liu, Dylan A. Lukes, William G. Griswold
Date of publication: 26 April 2023
Brief summary: This research looks at refactoring in the environment of computational notebooks, such as JupyterLab. From the analysis of 15,000 Jupyter notebooks, it is clear that refactoring was crucial to the authors of the notebook. The paper not only highlights the importance of refactoring to reduce technical debt, which is true regardless of whether a developer is using a computational notebook or an IDE, but also reveals the shortcomings of notebooks in providing adequate tools for an important aspect of computation for its user base. It is important for software developers to account for refactoring within their process, as well as when determining the developmental environment for their projects.