Title:

A systematic review on search-based refactoring

Authors:

Thainá Mariani, Silvia Regina Vergilio from Computer Science Department, Federal University of Paraná (UFPR)

Address:

https://www.sciencedirect.com/science/article/abs/pii/S0950584916303779

Date:

March 2017

Summary:

This article presents a systematic review to explore Search-Based Refactoring (SBR), analysing 71 primary studies over sixteen years to classify them by following common characteristics and trends based on the main SBR elements. Its result was that the use of code as the most addressed artifact and evolutionary algorithms as the dominant search technique. Furthermore, it was established that most of the solutions are included in the Flower's Catalog refactors. The authors of the article made research in three phases. First, they create the research protocol to be followed. Secondly, the data from the different documents are synthesized and analysed. Finally, they report the review and specify the mechanisms and format of the main report.
By identifying common characteristics and trends in SBR approaches, the article highlights the importance of refactoring processes in addressing code complexity and improving design.

Title:

Code smells and refactoring: A tertiary systematic review of challenges and observations

Authors:

Guilherme Lacerda, Marcelo Pimenta from Federal University of Rio Grande do Sul, Institute of Informatics Porto Alegre, RS, Brazil
Fabio Petrillo, Yann Gaël Guéhéneuc, from Concordia University, Department of Computer Science and Software Engineering Montreal, Quebec, Canada

Address:

https://www.sciencedirect.com/science/article/pii/S0164121220300881

Date:

September 2020

Summary:

This article explains the investigation the authors underwent to determine the relationship between code smells and refactoring by analysing 40 studies. They explore various aspects such as detection approaches, tools, refactoring techniques, and their impact on software quality. The review emphasizes on their importance regarding software maintenance and evolution. It also highlights different open issues for future research and discusses implications for practitioners, researchers, and instructors, aiming to bridge the gap between theoretical understanding and practical application in software engineering, which is why this article is significant regarding refactoring, and thus, Software Engineering.