## 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 on Search-Based Refactoring (SBR), a type of refactoring that uses search techniques and algorithms to treat the refactoring process as an optimization algorithm and which has been an increasingly popular topic of research in the field of computer science and engineering over the past years. The article analyses 71 primary studies over sixteen years to classify them by following common characteristics and trends based on the main SBR elements. The article concludes from this analysis that code is the most addressed artefact in the field of SBR and that evolutionary algorithms (a type of search algorithm which use markov chains and probability calculus to mimic mechanisms found in living beings) are the dominant search technique. Furthermore, it was established that most of the solutions found by SBR techniques are found in the 1999 book ‘Refactoring’, written by software engineer Martin Flower. The authors of the article made research in three phases. First, they created the research protocol to be followed. Secondly, the data from the different documents were synthesised and analysed. Finally, they reported the review and specified 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. We believe this article is of high interest to software engineers, as it illustrates the emerging field of search based refactoring.

## 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 that code smells and refactoring have with overall software quality and its ability to be improved and maintained. To do this the authors analysed 40 studies and explored various aspects about them such as the most common bad smells in code, detection approaches for said bad smells and available tools that can be useful for dealing with them. The article also goes over refactoring techniques and their usefulness when it comes to improving software quality. The review emphasises the importance of these refactoring techniques regarding software understandability, maintainability, testability, complexity, functionality, and reusability. It poses the current problems developers encounter in the refactoring process, such as the inability to detect bad smells present in code, the decision of which refactoring technique to apply and wether the time and resources needed to apply this techniques is proportional to the benefit obtained by the removal of these smells. Another crucial point it highlights is the presence of different open issues for future research on this field and discusses the implications this has for practitioners, researchers and instructors, aiming to bridge the gap between theoretical understanding and practical application in software engineering. We believe this article presents a clear image on the problem of bad smells in software development and the techniques and tools we can use to detect and eliminate them. This is why this article is significant regarding refactoring, and thus, Software Engineering.