Refactoring Articles

Refactoring with domain-driven design in an industrial context: An action research report

Authors: Ozan Özkan; Önder Babur; Mark G.J. van den Brand

Journal: Empirical Software Engineering, 28(4), Article 94

Published: July 2023

URL: https://link.springer.com/article/10.1007/s10664-023-10310-1

Summary: This research study investigates the effect of applying Domain Driven Design (DDD) principles to refactor an industrial software system. DDD in short is a set of principles aimed at aligning code better with how businesses actually work. Trying to make clear manageable code that accurately reflects the company as a whole and keeps everyone on the same page even the non coders. Working with an industrial partner in this study they refactored the system according to DDD. They evaluated its perceived effectiveness using qualitative focus groups and a modified Technology Acceptance Model questionnaire. Their findings demonstrated that DDD-based refactoring can substantially enhance software maintainability, offering standardized design improvements while highlighting challenges related to cognitive load and complexity for engineers. This work is relevant as it provides evidence on the practical application of refactoring in real-world contexts.

Visualizing software refactoring using radar charts

Authors: Abdel-Rahman Al-Ghuwairi; Dimah Al-Fraihat; Yousef Sharrab; Huda Alrashidi; Nouf Almujally; Ahmed Kittaneh; Ahmed Ali

Journal: Scientific Reports, volume 13, Article number 19530

Published: 13 March 2024

URL: https://www.nature.com/articles/s41598-023-44281-6

Summary: This study introduces the Radar Chart Refactoring Visualization (RcRV), a tool that generates radar chart visualizations to represent refactoring activities across multiple software releases. By mapping refactoring events as concentric rings on a radar chart, RcRV enables developers to analyze trends in code complexity, maintainability, and other quality metrics simultaneously. In the study they found that evaluations with developers showed the tool's usefulness and ease of use, suggesting that visualizing refactoring can enhance comprehension of code evolution and guide maintainability-focused refactoring decisions. The paper helps add context to the importance of refactoring when addressing challenges in code and understanding the cumulative impact of multiple automated refactoring operations.