

# Exploring ChatGPT's code refactoring capabilities: An empirical study

Kayla DePalma , Izabel Miminoshvili , Chiara Henselder , Kate Moss ,  
Eman Abdullah AlOmar

Efstathios Iosifidis, DVM

## Objectives

**Main Goal:** Assess ChatGPT's ability to refactor Java code segments.

**Research Questions:**

**RQ1:** Can ChatGPT effectively perform code refactoring?

**RQ2:** Does the refactored code maintain the functionality of the original?

**RQ3:** Can ChatGPT generate accurate documentation for the refactored code?

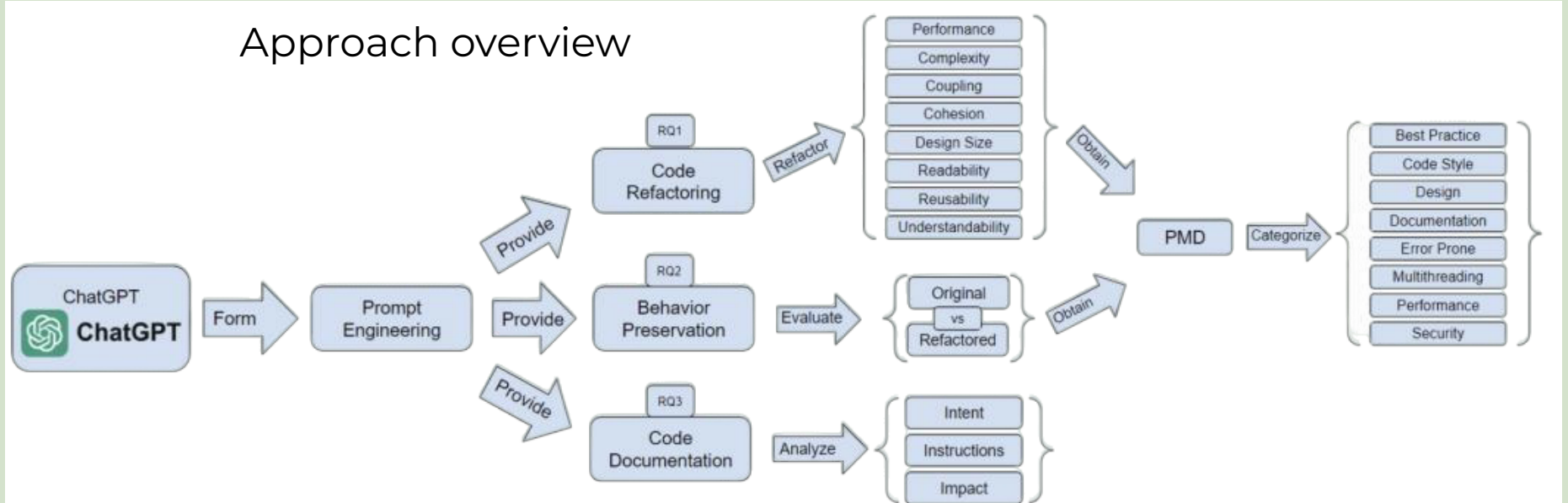
## Methodology

**Dataset:** Utilized 40 Java files, each containing one class with 20-50 lines of code.

**Prompt Engineering:** Developed specific prompts for ChatGPT to refactor code based on eight quality attributes.

**Analysis Tools:** Employed PMD (a source code analyzer) to identify coding flaws in the refactored segments.

### Approach overview



## Findings

**Effectiveness:** ChatGPT successfully refactored code segments in 39 out of 40 instances.

**Behavior Preservation:** The original functionality was preserved in 311 out of 320 trials.

**Documentation Accuracy:** Generated documentation was accurate in 310 out of 320 instances.

## Conclusions

**Strengths:** ChatGPT provides valuable suggestions for improving code quality, particularly with minor changes that enhance readability and efficiency.

**Limitations:** Variability in responses indicates unpredictability; human oversight is essential for complex modifications.

**Recommendations:** Suggest using ChatGPT as an aid for initial refactoring steps while relying on human programmers for final decisions.