Ira A. Fulton Schools of Engineering
Arizona State University

CSE 566: Software Project, Process and Quality Management

Code Refactoring Project

Purpose

Code refactoring is the process of restructuring existing computer code without changing its external behavior. Code refactoring is integral to agile software development as it is intended to improve the design, structure, and/or implementation of the software. While creating improvements, it also preserves functionality. Where applicable code refactoring tools can assist this process.

The goal of this project is to investigate the capabilities of code refactoring tools to utilize refactoring patterns to address code smells.

Objectives

Learners will be able to:

- Identify code smells.
- Describe code refactoring patterns and how they address code smells.
- Analyze how code refactoring patterns improve the quality of the resulting code.
- Evaluate the capabilities of a refactoring tool.
- Determine how the refactoring tool addresses or does not address refactoring patterns.

Technology Requirements

- Word processor
- PDF option to convert the typed document

Directions

Using the *Your Name_CSE 566_Code Refactoring Project_Template*, compose a three-part paper:

Part 1: Code Smells and Refactoring Patterns (250-500 words or ½ - 1 page)

- Identify three (3) code smells.
- Identify and briefly describe a refactoring pattern that addresses each code smell.
- Discuss how each refactoring pattern improves the quality of the code.
- Cite references where used in your paper.

Part 2: Refactoring Tools and Refactoring Patterns (250-500 words or ½ - 1 page)

- Identify one refactoring tool.
- Describe the capabilities of the refactoring tool.
- Describe how the tool addresses or does not address the refactoring patterns you identified in Part 1.
- Cite references where used in your paper.

Part 3: References (3-5 research-based references)

- Using IEEE or ACM style, list the references you used in your project paper.

Formatting Specifications

The *Your Name_CSE 566_Code Refactoring Project_Template* has the established formatting with a header for your identification information (Your Name) and the session you are taking this course (e.g., Spring B 2022), headings, subheadings, line spacing, font sizes, margins, and a sans serif font. These specifications were set up to support the use of assistive technologies, such as screen readers.

In-text citations and paraphrased references within the body of your paper should be in IEEE or ACM style. Whichever style you choose, your References page must be in the same style.

Submission Directions for Project Deliverables

You are given a limited number of attempts to submit your best work. The number of attempts is given to anticipate any submission errors you may have in regards to properly submitting your best work within the deadline (e.g., accidentally submitting the wrong paper). It is not meant for you to receive multiple rounds of feedback and then one (1) final submission. Only your most recent submission will be assessed.

You must submit your Code Refactoring Project deliverable in the appropriate submission space in the course. Learners may **not** email or use other means to submit any project for review, including feedback, and grading.

The Code Refactoring Project includes **one (1)** deliverable:

1. **Written Final Project Paper:** Your Code Refactoring Project must be a single PDF with the correct naming convention: *Your Name_CSE 566_Code Refactoring Project*. You are required to use the provided *Your Name_CSE 566_Code Refactoring Project_Template*.

Evaluation

Please review the rubric for how this project will be graded. The rubric can be viewed directly in your course, through the submission space for this project. Project papers will be evaluated based on each criterion and will receive a total score.

Prior to starting any graded coursework, learners are expected to read through the rubric so they know how they will be assessed. You are encouraged to self-assess your responses and make informed revisions before submitting your final work. Engaging in this learning practice will support you in developing your best work.

Project papers missing any part of the project will be graded based on what was submitted against the rubric criteria. Missing parts submitted after the deadline will not be graded.

Review the course syllabus for details regarding late penalties.

Scoring Criteria
The learner describes **three (3)** code refactoring patterns.
The learner describes how the selected patterns improve the quality of the resulting code.
The learner identifies and describes the capabilities of a refactoring tool.
The learner discusses how the tool either addresses or does not address the refactoring patterns chosen in Part 1.
The learner includes references and cites them.