RefactoringAnalyzer Project Report

Niko Hokkanen
University of Oulu
Oulu, Finland

Carlos Pantin
University of Oulu
Oulu, Finland

Paavo Parviainen
University of Oulu
Oulu, Finland

Niko Siltala
University of Oulu
Oulu, Finland

Jack Sundholm
University of Oulu
Oulu, Finland

ABSTRACT

This report, part of the Software Development, Maintenance, and Operations course in Fall 2024, explores refactoring activities across software projects, focusing on developer effort and bug-fixing commits. Utilizing tools like RefactoringMiner and custom Python scripts, the project analyzes code modifications and tracks developer effort (measured through Touched Lines of Code, TLOC) and bug-related issues through GitHub's API. The methodology includes repository cloning, refactoring identification, commit difference analysis, and developer effort quantification. Results are compiled into structured datasets (JSON, CSV), offering insights into refactoring practices and their impact on software maintenance and bug resolution.

KEYWORDS

Data Mining, RefactoringMiner, Bug-Fixing Commit, Developer Effort, Touched Lines of Code (TLOC), Github API, Software Maintenance, Commit Difference Analysis

ACM Reference Format:
Niko Hokkanen, Carlos Pantin, Paavo Parviainen, Niko Siltala, and Jack Sundholm. 2024. RefactoringAnalyzer Project Report. In . ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This report is part of the coursework for the **Software Development, Maintenance, and Operations** course (811372A) in Fall 2024. The project involves mining refactoring activities from a set of software projects and analyzing the developer effort and bug-fixing commits associated with those refactorings.

The analysis uses several tools, including RefactoringMiner for identifying refactorings, and additional Python scripts to collect data on developer effort and issue tracking. This document provides an overview of the project, the data collected, and the methodology used.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 REPOSITORY STRUCTURE

The project repository includes Python scripts and tools designed to perform the following tasks:

- `DeveloperEffort.py`: Tracks the effort exerted by developers during the refactoring process by counting touched lines of code (TLOC).
- `DividedProjects.py`: Handles project division tasks, potentially for parallel analysis.
- `GetBugIssueData.py`: Collects issue tracking and bug-fixing commit information from GitHub's API.
- `RefactoringRunner.py`: Runs RefactoringMiner on each cloned repository to collect refactoring data.
- `ProduceUniqueRepos.py`: Gathers unique repositories for analysis from the dataset.

These files work together to produce the necessary output in the form of JSON, CSV, and other data formats as required by the coursework instructions.

3 METHODOLOGY

The methodology follows the project steps outlined in the coursework, including:

3.1 Refactoring Mining

The refactoring mining process utilizes the `RefactoringRunner.py` script to extract refactoring activities from software repositories on GitHub, more specifically, Apache Foundation projects. This section outlines the methodology used to clone the repositories and analyze their refactoring history.

The dataset provided in the coursework is used to identify and clone the relevant GitHub repositories. Once cloned, the process is as follows:

(1) **Setup and Tools:** The `RefactoringRunner.py` script checks for the availability of essential tools such as `git` and `java`, as RefactoringMiner requires Java to operate. The path to the RefactoringMiner executable is determined dynamically within the script.
(2) **Cloning the Repository:** The script creates a temporary directory and clones the specified GitHub repository into this location. The cloning process is performed using the `subprocess` library.
(3) **Running RefactoringMiner:** After the repository is cloned, the script executes the RefactoringMiner tool to analyze the refactoring history of the project. It specifies the cloned repository and the desired output file in JSON format.
(4) **Error Handling:** The script is designed to handle subprocess errors and unexpected exceptions gracefully. If any issues occur during the cloning of the

repository or the execution of RefactoringMiner, they are logged with the corresponding error messages.

(5) **Data Structure of the JSON Output:** Upon successful completion, RefactoringMiner generates a JSON file containing structured information about the refactoring activities identified within the repository. The JSON file includes the following key attributes:
- `refactorings`: An array of objects, where each object represents a specific refactoring action detected. Each refactoring object contains:
  - `type`: The type of refactoring performed (e.g., method extraction, class renaming).
  - `commit_hash`: The hash of the commit where the refactoring occurred.
  - `timestamp`: The date and time when the commit was made.
  - `changed_files`: A list of files that were modified as part of the refactoring.
  - `diff`: The changes made to the code, including added and deleted lines.
- `repository`: Information about the repository from which the refactorings were mined, including:
  - `name`: The name of the repository.
  - `url`: The URL of the repository.

3.2 Commit Differences

The analysis of commit differences is essential for understanding the changes made in each refactoring commit (RC). This section describes the methodology used to calculate the differences between the current commit and its previous commit.

The process for obtaining commit differences involves the following steps:

(1) **Setup and Tools:** The analysis utilizes the `pydriller` library, a Python tool designed for mining software repositories and analyzing commit data. It provides functionalities to traverse commits, retrieve modified files, and extract diff information.

(2) **Iterating Through Commits:** The script iterates through the commits in the specified repository using the `Repository` class from the `pydriller` library. For each commit, it checks if the commit has a parent (indicating that it is not the initial commit).

(3) **Collecting Diff Information:** For each refactoring commit, the script collects the following information:
- `previous_commit_hash`: The hash of the previous commit.
- `diff_stats`: A dictionary containing:
  - `insertions`: The total number of lines added in the commit.
  - `deletions`: The total number of lines removed in the commit.
  - `files_modified`: The count of files that were modified in the commit.
- `diff_content`: A list of modified files, where each entry contains:
  - `filename`: The name of the modified file.
  - `added_lines`: A list of added lines, including their line numbers and content.
  - `deleted_lines`: A list of deleted lines, including their line numbers and content.

(4) **Output Management:** Once the commit differences are collected, the script ensures that the output directory exists. It then writes the gathered commit difference data into a structured JSON file. This file includes all relevant information about the commit diffs required for the project.

This structured approach to calculating commit differences enhances our understanding of the modifications made during refactoring activities and provides insights into the evolution of the codebase.

3.3 Developer Effort

To assess the developer effort involved in each refactoring, we measured the TLOC (Touched Lines of Code) between a refactoring commit and its immediate predecessor. The `DeveloperEffort.py` script implements this process, utilizing the `scc` tool to count the total lines of code affected by each refactoring.

3.3.1 Methodology.

(1) **Setup and Tools:** The `scc` tool was selected to provide language-agnostic line-counting capabilities across multiple programming languages. This tool counts the number of lines affected in a commit, covering a wide range of languages commonly used in refactoring activities, such as C, Python, Java, and JavaScript. We defined a set of file extensions for the tool to include, allowing `scc` to focus only on source code files relevant to the project. The script uses the `subprocess` library to run system commands, including `git checkout`, to switch between commits and gather accurate line count metrics from both the current and previous refactoring commits.

(2) **Process:** For each refactoring commit, `DeveloperEffort.py` first calculates the total lines of code (TLOC) by running the `scc` command on the current commit. The script then uses the `subprocess` library to execute the `git checkout` command, allowing it to check out the previous commit in the repository history and rerun `scc` to calculate the TLOC for this commit as well. The TLOC absolute difference between these two commits is then calculated to quantify the extent code changes made during the refactoring and know the Touched Lines of Code.

(3) **Data Storage:** Each refactoring commit's developer effort data is recorded with the following attributes: `Developer`, `Refactoring Hash`, `Previous Hash`, `TLOC`, `Current LOC`, and `Previous LOC`. This information is output to a CSV file, where each row corresponds to a refactoring commit.

3.4 Bug-fixing Commits Github

For projects that utilize GitHub as an Issue Tracking System (ITS), the process of mining bug-fixing commits involves using the GitHub REST API to gather relevant issue data. This section outlines the methodology implemented in the

`GetBugIssueData.py` script for collecting bug-related information.

The script operates as follows:

(1) **Setup and Tools:** The analysis is performed using the `requests` library for making HTTP requests to the GitHub API. The script is designed to fetch issues from a specified repository, filtering for bug-related issues if necessary.
(2) **Fetching Repository Information:** The script begins by extracting the owner and repository name from the provided GitHub repository URL. It ensures the repository uses GitHub's issue tracking feature by making a request to the appropriate endpoint.
(3) **Handling Rate Limits:** To comply with GitHub API rate limits, the script monitors the number of API calls made and sleeps for a calculated duration if approaching the limit. This prevents interruptions due to exceeding the allowed number of requests.
(4) **Collecting Issue Data:** The script retrieves all issues associated with the repository using pagination to ensure it collects all relevant data. It subsequently serializes the issue information into a JSON file, storing details about each issue, including its state and labels.
(5) **Output Management:** Upon completion, the script generates a JSON file containing all the retrieved issues. Additionally, it updates a results index file to log the outcome of the operation, including the total number of issues found and any errors encountered during the process.

3.5 Bug-fixing Commits Jira

For projects that utilize Jira as an Issue Tracking System (ITS), the process of mining bug-fixing commits involves using the JIRA REST API to gather relevant issue data. This section outlines the methodology implemented in the `GetBugIssueDataJira.py` script for collecting bug-related information.

The script operates as follows:

(1) **Setup and Tools:** The analysis is performed using the `requests` library for making HTTP requests to the Jira API. The script requires an API token for authentication and fetches bug-related issues from a specified Jira project that corresponds to the provided repository.
(2) **Fetching Project Information:** The script retrieves a list of available Jira projects using the `project` endpoint. It filters projects based on single-word keys to match the base name of the repository provided by the user.
(3) **Handling Rate Limits:** The script ensures compliance with Jira's API usage guidelines by handling HTTP responses and errors gracefully. It terminates or retries requests as needed to avoid disruptions due to invalid responses.
(4) **Collecting Issue Data:** Bug issues are fetched using the Jira `search` endpoint with a JQL query that filters for issues labeled as "Bug" within the specified project. The script uses pagination to collect all relevant issues, iterating until all pages are retrieved.
(5) **Data Processing and Output:** The retrieved issues are stored into a JSON file, containing details such as issue type, state, and description.
(6) **Repository Matching:** The script maps the repository URL to the base name of a corresponding Jira project, ensuring that issues are correctly associated with the intended project.

4 RESULTS

The project produces several outputs, including:

- Refactoring data in JSON format, with attributes such as refactoring type, commit hash, and inter-refactoring period.
- Commit diffs for each refactoring commit, including numerical data on the differences and the modified code content.
- Developer effort data, with attributes such as refactoring hash, previous hash, and TLOC.
- Bug-fixing commit data from GitHub's issue tracking system and Jira.
- A log for every repository analyzed stating whether errors occurred or not.

These results are compiled into a structured dataset, which forms the basis for the analysis performed in this report.

5 SCRIPT AND REQUIREMENT MAPPING

The following section maps the various Python scripts from the repository to the project requirements specified in the coursework instructions.

5.1 Step (a) - Cloning GitHub Projects

This step involves cloning the required repositories from GitHub. The `RefactoringRunner.py` script handles this task using the `subprocess` library to execute Git commands for cloning repositories from the list provided in `uniqueRepositories.txt`.

5.2 Step (b) - Mining Refactoring Activity (RefactoringMiner)

The `RefactoringRunner.py` script also runs RefactoringMiner to analyze the refactoring activity across all commits in each repository. It invokes the CLI commands for RefactoringMiner and produces the necessary output in JSON format for further analysis.

5.3 Step (c) - Calculating Diff Changes

The `GetGitDiff.py` script calculates and collects the diff changes between the detected commit and the previous commit. It uses the `pydriller` tool to gather this information.

5.4 Step (d) - Collecting Developer Effort (TLOC)

The `DeveloperEffort.py` script handles the task of collecting the total touched lines of code (TLOC) for each refactoring. It uses the `scc` tool to gather LOC information from

both the refactoring commit and the previous commit, and calculates the absolute difference.

5.5 Step (e) - Mining Bug-Fixing Commits

The `GetBugIssueData.py` and `GetBugIssueDataJira.py` scripts are responsible for collecting bug-fixing commits by interacting with GitHub's REST API and Jira's API.

5.6 Step (f) - Data Collection Logic and Submission Format

The `Runner.py` script orchestrates the entire process, calling all relevant scripts and producing the required output files in JSON and CSV formats. This script is responsible for ensuring that each part of the process runs smoothly, from cloning repositories to generating refactoring data and developer effort statistics.

6 CONCLUSION

The `RefactoringAnalyzer` project mines and analyzes refactoring activities in software projects. By combining tools like RefactoringMiner and Python scripts, the project provides valuable insights into developer effort and bug-fixing activity, supporting the broader application of refactoring in practice. Logging functionalities were also implemented to know if there was an error when running the `Runner.py` script to know if a repository failed to be analyzed completely. The outputs were successfully produced the 11th of November, 2024. It is important to note that two repositories failed to run, those being: fineract and plc4x. https://github.com/apache/fineract , https://github.com/apache/plc4x.

A REPOSITORY LIST

The following is the list of 381 Apache repositories used for the analysis in this project:

- https://github.com/apache/fineract
- https://github.com/apache/sling-org-apache-sling-feature-karaf
- https://github.com/apache/sling-org-apache-sling-committer-cli
- https://github.com/apache/hadoop-ozone
- https://github.com/apache/sling-org-apache-sling-commons-content-processing
- https://github.com/apache/sling-org-apache-sling-testing-osgi-mock
- https://github.com/apache/sling-org-apache-sling-feature-apiregions-model
- https://github.com/apache/sling-org-apache-sling-repoinit-it
- https://github.com/apache/sling-org-apache-sling-hc-support
- https://github.com/apache/sling-org-apache-sling-resourceresolver
- https://github.com/apache/sling-org-apache-sling-starter
- https://github.com/apache/sling-org-apache-sling-feature-apiregions
- https://github.com/apache/openmeetings
- https://github.com/apache/sling-org-apache-sling-engine
- https://github.com/apache/sling-org-apache-sling-distribution-journal-kafka
- https://github.com/apache/sling-org-apache-sling-repoinit-parser
- https://github.com/apache/sling-org-apache-sling-commons-cache-container-test
- https://github.com/apache/sling-org-apache-sling-api
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-repl
- https://github.com/apache/sling-org-apache-sling-jcr-oak-server
- https://github.com/apache/sling-org-apache-sling-launchpad-startupmanager
- https://github.com/apache/submarine
- https://github.com/apache/sling-org-apache-sling-adapter
- https://github.com/apache/sling-org-apache-sling-extensions-webconsolesecurityprovider
- https://github.com/apache/sling-org-apache-sling-provisioning-model
- https://github.com/apache/sling-org-apache-sling-fsresource
- https://github.com/apache/jspwiki-builder
- https://github.com/apache/sling-org-apache-sling-contentparser-json
- https://github.com/apache/sling-org-apache-sling-i18n
- https://github.com/apache/sling-org-apache-sling-models-caconfig
- https://github.com/apache/sling-org-apache-sling-testing-resourceresolver-mock
- https://github.com/apache/sling-org-apache-sling-scripting-java
- https://github.com/apache/plc4x
- https://github.com/apache/sling-org-apache-sling-servlets-post
- https://github.com/apache/sling-org-apache-sling-sitemap

- https://github.com/apache/poi-parent
- https://github.com/apache/sling-project-archetype
- https://github.com/apache/sling-org-apache-sling-commons-threads
- https://github.com/apache/sling-org-apache-sling-jcr-jackrabbit-accessmanager
- https://github.com/apache/sling-org-apache-sling-xss
- https://github.com/apache/sling-org-apache-sling-testing-sling-mock
- https://github.com/apache/sling-org-apache-sling-feature-modelconverter
- https://github.com/apache/sling-org-apache-sling-scripting-sightly
- https://github.com/apache/sling-org-apache-sling-resourceaccesssecurity
- https://github.com/apache/sling-kickstart-maven-plugin
- https://github.com/apache/sling-initial-content-archetype
- https://github.com/apache/sling-org-apache-sling-karaf-integration-tests
- https://github.com/apache/sling-org-apache-sling-security
- https://github.com/apache/commons-statistics
- https://github.com/apache/sling-scriptingbundle-maven-plugin
- https://github.com/apache/sling-org-apache-sling-distribution-journal-it
- https://github.com/apache/sling-org-apache-sling-feature-launcher
- https://github.com/apache/sling-org-apache-sling-scripting-jsp-api
- https://github.com/apache/sling-org-apache-sling-servlets-get
- https://github.com/apache/sling-org-apache-sling-scripting-jsp
- https://github.com/apache/sling-org-apache-sling-scripting-freemarker
- https://github.com/apache/sling-org-apache-sling-starter-content
- https://github.com/apache/sling-org-apache-sling-resource-presence
- https://github.com/apache/sling-org-apache-sling-junit-teleporter
- https://github.com/apache/sling-org-apache-sling-karaf-configs
- https://github.com/apache/sling-content-package-archetype
- https://github.com/apache/sling-maven-plugin
- https://github.com/apache/sling-org-apache-sling-auth-core
- https://github.com/apache/incubator-hop
- https://github.com/apache/isis
- https://github.com/apache/sling-org-apache-sling-adapter-annotations
- https://github.com/apache/sling-org-apache-sling-scripting-el-api
- https://github.com/apache/sling-org-apache-sling-contentparser-xml
- https://github.com/apache/cxf
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-testing-content
- https://github.com/apache/sling-org-apache-sling-rewriter
- https://github.com/apache/ofbiz-plugins
- https://github.com/apache/groovy
- https://github.com/apache/sling-feature-converter-maven-plugin
- https://github.com/apache/sling-org-apache-sling-scripting-javascript
- https://github.com/apache/sling-org-apache-sling-installer-provider-file
- https://github.com/apache/sling-org-apache-sling-testing-clients
- https://github.com/apache/sling-org-apache-sling-nosql-launchpad
- https://github.com/apache/sling-org-apache-sling-feature
- https://github.com/apache/incubator-iotdb
- https://github.com/apache/sling-parent
- https://github.com/apache/sling-org-apache-sling-capabilities-jcr
- https://github.com/apache/commons-numbers
- https://github.com/apache/pdfbox-reactor
- https://github.com/apache/apache-dolphinscheduler
- https://github.com/apache/sling-org-apache-sling-testing-sling-mock-oak
- https://github.com/apache/struts
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-testing
- https://github.com/apache/sling-org-apache-sling-launchpad-test-fragment
- https://github.com/apache/sling-org-apache-sling-commons-jcr-file
- https://github.com/apache/sling-org-apache-sling-feature-extension-content
- https://github.com/apache/sling-org-apache-sling-commons-html
- https://github.com/apache/sling-whiteboard
- https://github.com/apache/sling-org-apache-sling-feature-applicationbuilder
- https://github.com/apache/sling-org-apache-sling-commons-mime
- https://github.com/apache/sling-org-apache-sling-servlets-resolver
- https://github.com/apache/sling-org-apache-sling-scripting-thymeleaf
- https://github.com/apache/hop
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-compiler
- https://github.com/apache/sling-org-apache-sling-commons-messaging-mail
- https://github.com/apache/sling-org-apache-sling-nosql-couchbase-resourceprovider
- https://github.com/apache/sling-org-apache-sling-reqanalyzer
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-compiler-java
- https://github.com/apache/sling-org-apache-sling-feature-analyser
- https://github.com/apache/sling-org-apache-sling-commons-cache-api
- https://github.com/apache/sling-org-apache-sling-servlet-helpers
- https://github.com/apache/sling-org-apache-sling-caconfig-api
- https://github.com/apache/sling-org-apache-sling-connection-timeout-agent

- https://github.com/apache/sling-org-apache-sling-commons-scheduler
- https://github.com/apache/JMeter
- https://github.com/apache/sling-org-apache-sling-javax-activation
- https://github.com/apache/sling-org-apache-sling-jcr-packageinit
- https://github.com/apache/sling-org-apache-sling-validation-test-services
- https://github.com/apache/sling-org-apache-sling-servlets-annotations
- https://github.com/apache/jackrabbit-filevault
- https://github.com/apache/sling-org-apache-sling-serviceusermapper
- https://github.com/apache/sling-org-apache-sling-jcr-registration
- https://github.com/apache/sling-org-apache-sling-installer-core
- https://github.com/apache/sling-org-apache-sling-installer-it
- https://github.com/apache/sling-adapter-annotations
- https://github.com/apache/sling-org-apache-sling-auth-saml2
- https://github.com/apache/sling-org-apache-sling-event
- https://github.com/apache/sling-org-apache-sling-scripting-groovy
- https://github.com/apache/sling-org-apache-sling-resourcemerger
- https://github.com/apache/sling-org-apache-sling-karaf-features
- https://github.com/apache/sling-org-apache-sling-testing-rules
- https://github.com/apache/sling-htl-maven-plugin
- https://github.com/apache/sling-org-apache-sling-feature-cpconverter
- https://github.com/apache/sling-org-apache-sling-jcr-repoinit
- https://github.com/apache/sling-org-apache-sling-feature-inventoryprinter
- https://github.com/apache/sling-org-apache-sling-feature-extension-apiregions
- https://github.com/apache/sling-org-apache-sling-caconfig-integration-tests
- https://github.com/apache/sling-org-apache-sling-caconfig-bnd-plugin
- https://github.com/apache/sling-org-apache-sling-commons-johnzon
- https://github.com/apache/sling-org-apache-sling-models-api
- https://github.com/apache/sling-org-apache-sling-testing-hamcrest
- https://github.com/apache/sling-org-apache-sling-nosql-generic
- https://github.com/apache/sling-org-apache-sling-commons-messaging
- https://github.com/apache/sling-slingstart-maven-plugin
- https://github.com/apache/sling-org-apache-sling-testing-jcr-mock
- https://github.com/apache/sling-org-apache-sling-models-integration-tests
- https://github.com/apache/sling-org-apache-sling-junit-performance
- https://github.com/apache/sling-org-apache-sling-testing-caconfig-mock-plugin
- https://github.com/apache/sling-org-apache-sling-launchpad-integration-tests
- https://github.com/apache/sling-org-apache-sling-commons-osgi
- https://github.com/apache/apache-ratis
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-runtime
- https://github.com/apache/sling-org-apache-sling-feature-resolver
- https://github.com/apache/sling-org-apache-sling-jcr-contentparser
- https://github.com/apache/sling-org-apache-sling-distribution-api
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-js-provider
- https://github.com/apache/sling-org-apache-sling-scripting-core
- https://github.com/apache/sling-org-apache-sling-jcr-jackrabbit-usermanager
- https://github.com/apache/sling-org-apache-sling-installer-provider-jcr
- https://github.com/apache/sling-org-apache-sling-commons-content-analyzing
- https://github.com/apache/sling-org-apache-sling-auth-form
- https://github.com/apache/apache-daffodil
- https://github.com/apache/commons-geometry
- https://github.com/apache/sling-org-apache-sling-startupfilter-disabler
- https://github.com/apache/commons-math
- https://github.com/apache/sling-org-apache-sling-junit-core
- https://github.com/apache/sling-org-apache-sling-thumbnails
- https://github.com/apache/sling-org-apache-sling-jcr-webconsole
- https://github.com/apache/sling-org-apache-sling-commons-compiler
- https://github.com/apache/sling-org-apache-sling-jcr-filetransfer
- https://github.com/apache/sling-org-apache-sling-installer-factory-feature
- https://github.com/apache/sling-org-apache-sling-tooling-support-install
- https://github.com/apache/sling-org-apache-sling-scripting-spi
- https://github.com/apache/sling-org-apache-sling-discovery-oak
- https://github.com/apache/sling-org-apache-sling-contentparser-testutils
- https://github.com/apache/sling-org-apache-sling-testing-serversetup
- https://github.com/apache/sling-org-apache-sling-nosql-mongodb-resourceprovider
- https://github.com/apache/sling-org-apache-sling-jcr-base
- https://github.com/apache/sling-org-apache-sling-junit-healthcheck
- https://github.com/apache/sling-org-apache-sling-capabilities

- https://github.com/apache/sling-org-apache-sling-testing-paxexam
- https://github.com/apache/sling-org-apache-sling-app-cms
- https://github.com/apache/sling-org-apache-sling-validation-examples
- https://github.com/apache/sling-org-apache-sling-auth-xing-oauth
- https://github.com/apache/sling-org-apache-sling-discovery-standalone
- https://github.com/apache/sling-org-apache-sling-models-impl
- https://github.com/apache/sling-archetype-parent
- https://github.com/apache/sling-org-apache-sling-scripting-bundle-tracker-it
- https://github.com/apache/sling-org-apache-sling-junit-scriptable
- https://github.com/apache/sling-org-apache-sling-contentparser-api
- https://github.com/apache/sling-org-apache-sling-installer-factory-packages
- https://github.com/apache/sling-org-apache-sling-dynamic-include
- https://github.com/apache/camel
- https://github.com/apache/sling-org-apache-sling-feature-r2f
- https://github.com/apache/sling-jspc-maven-plugin
- https://github.com/apache/sling-org-apache-sling-installer-factory-configuration
- https://github.com/apache/sling-org-apache-sling-commons-crypto
- https://github.com/apache/sling-org-apache-sling-servlets-annotations-it
- https://github.com/apache/ofbiz-framework
- https://github.com/apache/sling-slingfeature-maven-plugin
- https://github.com/apache/sling-org-apache-sling-scripting-esx
- https://github.com/apache/sling-org-apache-sling-validation-core
- https://github.com/apache/sling-org-apache-sling-scripting-api
- https://github.com/apache/sling-org-apache-sling-contentparser-xml-jcr
- https://github.com/apache/sling-org-apache-sling-superimposing
- https://github.com/apache/sling-org-apache-sling-commons-metrics-rrd4j
- https://github.com/apache/sling-org-apache-sling-distribution-journal-messages
- https://github.com/apache/sling-org-apache-sling-pipes
- https://github.com/apache/sling-org-apache-sling-launchpad-base
- https://github.com/apache/sling-slingstart-archetype
- https://github.com/apache/sling-org-apache-sling-models-validation-impl
- https://github.com/apache/sling-org-apache-sling-installer-console
- https://github.com/apache/sling-org-apache-sling-jcr-resource
- https://github.com/apache/sling-org-apache-sling-commons-metrics
- https://github.com/apache/gora
- https://github.com/apache/sling-org-apache-sling-karaf-distribution
- https://github.com/apache/sling-org-apache-sling-jcr-resourcesecurity
- https://github.com/apache/sling-org-apache-sling-distribution-journal
- https://github.com/apache/sling-org-apache-sling-event-dea
- https://github.com/apache/sling-org-apache-sling-commons-classloader
- https://github.com/apache/sling-apache-sling-jar-resource-bundle
- https://github.com/apache/sling-org-apache-sling-bundleresource-impl
- https://github.com/apache/sling-org-apache-sling-testing-logging-mock
- https://github.com/apache/sling-org-apache-sling-commons-clam
- https://github.com/apache/sling-org-apache-sling-nosql-couchbase-client
- https://github.com/apache/sling-org-apache-sling-models-jacksonexporter
- https://github.com/apache/jackrabbit-filevault-package-maven-plugin
- https://github.com/apache/sling-org-apache-sling-graphql-core
- https://github.com/apache/sling-org-apache-sling-settings
- https://github.com/apache/sling-org-apache-sling-scripting-jsp-jstl
- https://github.com/apache/org.apache.nemo:nemo-project
- https://github.com/apache/any23
- https://github.com/apache/sling-org-apache-sling-jcr-contentloader
- https://github.com/apache/sling-org-apache-sling-scripting-jsp-taglib
- https://github.com/apache/sling-launchpad-standalone-archetype
- https://github.com/apache/sling-feature-launcher-maven-plugin
- https://github.com/apache/sling-org-apache-sling-scripting-jsp-taglib-compat
- https://github.com/apache/sling-org-apache-sling-installer-factory-model
- https://github.com/apache/sling-org-apache-sling-commons-log-webconsole
- https://github.com/apache/sling-org-apache-sling-karaf-launchpad-oak-tar-integration-tests
- https://github.com/apache/sling-org-apache-sling-feature-diff
- https://github.com/apache/sling-org-apache-sling-discovery-support
- https://github.com/apache/sling-org-apache-sling-testing-email
- https://github.com/apache/sling-org-apache-sling-clam
- https://github.com/apache/commons-rng
- https://github.com/apache/sling-org-apache-sling-launchpad-test-bundles

- https://github.com/apache/sling-org-apache-sling-validation-api
- https://github.com/apache/sling-org-apache-sling-fragment-ws
- https://github.com/apache/sling-org-apache-sling-commons-log
- https://github.com/apache/sling-org-apache-sling-jms
- https://github.com/apache/sling-launchpad-comparator
- https://github.com/apache/commons-math
- https://github.com/apache/sling-org-apache-sling-graphql-schema-aggregator
- https://github.com/apache/sling-org-apache-sling-distribution-sample
- https://github.com/apache/sling-org-apache-sling-resourceaccesssecurity-it
- https://github.com/apache/sling-org-apache-sling-serviceuser-webconsole
- https://github.com/apache/sling-org-apache-sling-commons-contentdetection
- https://github.com/apache/sling-org-apache-sling-jcr-jcr-wrapper
- https://github.com/apache/incubator-tamaya
- https://github.com/apache/sling-org-apache-sling-installer-factory-subsystems-base
- https://github.com/apache/roller-master
- https://github.com/apache/sling-org-apache-sling-scripting-bundle-tracker
- https://github.com/apache/sling-org-apache-sling-hapi-samplecontent
- https://github.com/apache/sling-org-apache-sling-caconfig-impl
- https://github.com/apache/sling-org-apache-sling-distribution-core
- https://github.com/apache/sling-org-apache-sling-extensions-logback-groovy-fragment
- https://github.com/apache/incubator-tamaya-extensions
- https://github.com/apache/sling-launchpad-debian
- https://github.com/apache/sling-org-apache-sling-feature-extension-unpack
- https://github.com/apache/sling-org-apache-sling-auth-xing-login
- https://github.com/apache/sling-org-apache-sling-resource-editor
- https://github.com/apache/sling-org-apache-sling-hc-it
- https://github.com/apache/sling-org-apache-sling-urlrewriter
- https://github.com/apache/sling-org-apache-sling-jcr-jackrabbit-base
- https://github.com/apache/sling-org-apache-sling-discovery-commons
- https://github.com/apache/sling-org-apache-sling-commons-logservice
- https://github.com/apache/sling-org-apache-sling-commons-threaddump
- https://github.com/apache/sling-org-apache-sling-resourcecollection
- https://github.com/apache/karaf
- https://github.com/apache/sling-org-apache-sling-junit-remote
- https://github.com/apache/sling-org-apache-sling-feature-io
- https://github.com/apache/sling-org-apache-sling-jcr-classloader
- https://github.com/apache/sling-org-apache-sling-jobs
- https://github.com/apache/incubator-ratis
- https://github.com/apache/sling-org-apache-sling-tooling-support-source
- https://github.com/apache/xmlbeans
- https://github.com/apache/sling-org-apache-sling-jcr-js-nodetypes
- https://github.com/apache/sling-org-apache-sling-discovery-base
- https://github.com/apache/knox-gateway
- https://github.com/apache/sling-org-apache-sling-caconfig-spi
- https://github.com/apache/sling-org-apache-sling-scripting-xproc
- https://github.com/apache/sling-org-apache-sling-launchpad-api
- https://github.com/apache/sling-org-apache-sling-jobs-it-services
- https://github.com/apache/sling-org-apache-sling-jcr-webdav
- https://github.com/apache/sling-org-apache-sling-distribution-kryo-serializer
- https://github.com/apache/sling-org-apache-sling-event-api
- https://github.com/apache/sling-org-apache-sling-file-optimization
- https://github.com/apache/sling-org-apache-sling-mongodb
- https://github.com/apache/sling-org-apache-sling-jmx-provider
- https://github.com/apache/sling-org-apache-sling-jcr-contentparser
- https://github.com/apache/sling-org-apache-sling-launchpad-test-services
- https://github.com/apache/sling-org-apache-sling-hc-samples
- https://github.com/apache/sling-org-apache-sling-installer-factory-deploymentpackage
- https://github.com/apache/sling-jcrinstall-bundle-archetype
- https://github.com/apache/sling-org-apache-sling-fragment-activation
- https://github.com/apache/sling-org-apache-sling-jcr-davex
- https://github.com/apache/sling-org-apache-sling-distribution-avro-serializer
- https://github.com/apache/sling-org-apache-sling-commons-fsclassloader
- https://github.com/apache/sling-org-apache-sling-installer-hc
- https://github.com/apache/sling-org-apache-sling-bnd-models
- https://github.com/apache/sling-maven-launchpad-plugin
- https://github.com/apache/sling-servlet-archetype
- https://github.com/apache/sling-org-apache-sling-crankstart-test-model
- https://github.com/apache/sling-org-apache-sling-commons-cache-impl
- https://github.com/apache/sling-org-apache-sling-tracer

- https://github.com/apache/sling-org-apache-sling-hc-api
- https://github.com/apache/sling-org-apache-sling-extensions-classloader-leak-detector
- https://github.com/apache/sling-org-apache-sling-installer-provider-installhook
- https://github.com/apache/sling-org-apache-sling-query
- https://github.com/apache/sling-org-apache-sling-resourcebuilder
- https://github.com/apache/sling-launchpad-webapp-archetype
- https://github.com/apache/commons-compress
- https://github.com/apache/incubator-daffodil
- https://github.com/apache/sling-org-apache-sling-jcr-api
- https://github.com/apache/sling-org-apache-sling-launchpad-testing
- https://github.com/apache/sling-org-apache-sling-launchpad-test-services-war
- https://github.com/apache/sling-org-apache-sling-launchpad-installer
- https://github.com/apache/sling-maven-jcrocm-plugin
- https://github.com/apache/sling-org-apache-sling-discovery-api
- https://github.com/apache/sling-org-apache-sling-oak-restrictions
- https://github.com/apache/sling-org-apache-sling-auth-xing-api
- https://github.com/apache/sling-bundle-archetype
- https://github.com/apache/sling-org-apache-sling-bnd-plugins
- https://github.com/apache/sling-org-apache-sling-scripting-sightly-models-provider
- https://github.com/apache/ant-master
- https://github.com/apache/sling-org-apache-sling-fragment-xml
- https://github.com/apache/sling-org-apache-sling-extensions-webconsolebranding
- https://github.com/apache/sling-org-apache-sling-resource-inventory
- https://github.com/apache/sling-org-apache-sling-commons-testing
- https://github.com/apache/sling-org-apache-sling-tenant
- https://github.com/apache/sling-org-apache-sling-mom
- https://github.com/apache/sling-org-apache-sling-discovery-impl
- https://github.com/apache/sling-org-apache-sling-resource-filter
- https://github.com/apache/sling-org-apache-sling-hapi
- https://github.com/apache/sling-org-apache-sling-scripting-console
- https://github.com/apache/sling-org-apache-sling-hapi-client
- https://github.com/apache/sling-org-apache-sling-jcr-repository-it-resource-versioning
- https://github.com/apache/sling-org-apache-sling-tail
- https://github.com/apache/sling-org-apache-sling-fragment-nashorn
- https://github.com/apache/sling-taglib-archetype
- https://github.com/apache/sling-site
- https://github.com/apache/sling-org-apache-sling-extensions-slf4j-mdc
- https://github.com/apache/sling-org-apache-sling-servlets-resolver-api
- https://github.com/apache/sling-org-apache-sling-datasource
- https://github.com/apache/sling-org-apache-sling-bnd-plugin-headers-parameters-remove
- https://github.com/apache/sling-org-apache-sling-installer-factory-subsystems
- https://github.com/apache/sling-org-apache-sling-fragment-transaction
- https://github.com/apache/sling-org-apache-sling-featureflags
- https://github.com/apache/shiro
- https://github.com/apache/sling-org-apache-sling-jobs-it
- https://github.com/apache/sling-org-apache-sling-crankstart-test-services
- https://github.com/apache/sling-org-apache-sling-startupfilter
- https://github.com/apache/sling-org-apache-sling-kickstart
- https://github.com/apache/sling-org-apache-sling-hc-junit-bridge
- https://github.com/apache/incubator-milagro-MPC
- https://github.com/apache/sling-org-apache-sling-paxexam-util
- https://github.com/apache/fineract-cn-group-finance
- https://github.com/apache/milagro
- https://github.com/apache/sling-org-apache-sling-distribution-it
- https://github.com/apache/sling-org-apache-sling-starter-startup
- https://github.com/apache/log4cxx
- https://github.com/apache/incubator-seatunnel
- https://github.com/apache/incubator-tamaya-sandbox
- https://github.com/apache/servicecomb-toolkit
- https://github.com/apache/servicecomb-pack
- https://github.com/apache/jspwiki
- https://github.com/apache/poi
- https://github.com/apache/pdfbox-jbig2
- https://github.com/apache/dolphinscheduler
- https://github.com/apache/ratis
- https://github.com/apache/daffodil
- https://github.com/apache/incubator-nemo
- https://github.com/apache/roller
- https://github.com/apache/knox
- https://github.com/apache/ant
- https://github.com/apache/logging-log4cxx

A PROJECT INSTRUCTIONS

Project instructions are included in the appendix on the next page.

Software Development, Maintenance and Operations
811372A

Fall 2024
Course Project - Option B

October 15, 2024

INSTRUCTIONS: You may use any books, references, notes and programs. The course already provides some supporting material to complete the course projects. Please provide your codes (preferably *Python*), output, and brief comments in a file and submit it to Moodle. Likewise, write your *name* and *student number* in your file. (*Recommendation*: For a structured submission procedure, consider using a *version control* platform (e.g. GitHub) to keep track of your work and provide it within the submission file).

PRESENTATION: Source code **refactoring** is a well-established approach to improving source code quality without compromising its external behaviour. Providing refactoring **recommendations** closer to what developers perceive as relevant may support the broader application of refactoring in practice. Therefore, you will **mine** the refactoring activity of the involved software projects, and further, you will capture the developers **effort** performing such activity. Similarly, you will put hands on the **issue tracking system** as an advanced data mining research experience.

Consider the dataset provided in the following link [1].

(a) Download the zipped file. Take a look on the *project* column. Get the unique instances. Find the pattern to obtain the GitHub links of the listed projects. (**Hint:** All the included projects are inside the *Apache* source foundation.)

(b) Clone the GitHub projects into your machine from the built GitHub links. (**Hint:** *subprocess* library might be *one* option.)

(c) Mine the refactoring activity applied in the history of the cloned projects. For that, you might use *RefactoringMiner* library [2]. The provided link contains detailed steps for using this tool. **Hints:**

- First, *build* the tool into your machine, it can be anywhere but make sure to remember the path.
- Looking at the data dimension to be mined, you may consider using the CLI options for RefactoringMiner (You should mine all the commits from each repository.)
- If you are using Python, take a look, for instance, at the *subprocess* library to run commands in the CL.
- Provide for each project a table presenting the total number of refactorings made of each refactoring type and the average time of the inter-refactoring periods.

(d) **Calculate** (The *modification numerical data* for ADD and DEL within each file diff) and **collect** (literally the modified code in text) the *diff* change between the detected commit and the previous commit. (**Hint:** You might use *pydriller* library if you are using Python for this project.) From the submission format section of this project:

[1] https://filesender.funet.fi/?s=download&token=2435fa06-afd0-4ff2-a6d4-7d9c493bc0a0
[2] https://github.com/tsantalis/RefactoringMiner?tab=readme-ov-file#general-info

- The output for the commit *diff* could be a json file with attributes "commit_hash", "previous_commit_hash", "diff stats", "diff content".

(e) Collect developers *effort*. Collect the total count of touched lines of code (TLOCs) for each refactoring and each developer. (**Hint:** You might need *subprocess* library and the *scc* tool seen in Weekly exercise lectures.) **Important notes:**

- Get the LOC for the Refactoring Commit (RC) being analyzed.
- Checkout to the previous commit (not the previous RC but the previous commit in the version history right before the RC analyzed) and get the LOC as well.
- TLOC shall be the absolute Δ between both numbers.
- **NOTE:** When calculating the LOC with tools like *scc*, be careful with the "programming languages" the tool considers (it considers all as programming languages). To discard those "non-programming languages", you can find the languages considered to be programming languages in the TIOBE index (check the link attached to the name).

(f) Mine *bug-fixing* commits.

- Check in GitHub whether the projects are using GitHub as Issue Tracking System (ITS) or not (**Hint(s):** Check out GitHub's REST API documentation. You might need *Requests* library to work with APIs.)
- For projects using GitHub as ITS, mine **all** (It's fine dumping the entire output) the issue data. (**Hint:** Beware of the API request rate limit. More info in the GitHub docs.)
- For projects not using GitHub as ITS, a special lab lecture will be announced for JIRA API data collection.

(g) **NOTES:**

- Think about the logic to perform the data mining process. (**Hint:** Cloning all the projects may overload your machine.)
- The submission of this part must consider all the collected data, the scripts utilized and a brief report on each with the reasoning of each of the steps performed.

(h) **SUBMISSION FORMAT:**

- The output from the RMiner should create a **json** file, please name is with the repository name, or within a folder with the repository name (e.g. "../accumulo/rminer-output.json" or "../rminer-outputs/accumulo.json")
- The output for the commit *diff* could be a json file with attributes "commit_hash", "previous_commit_hash", "diff stats", "diff content".
- The output for the developers' effort could be a csv with headers such as ("refactoring_hash", "previous_hash", "TLOC").
- Consider for instance compression the submission through the attached file compression engine [3]. Choose the link generation option.

BONUSES: Anyone in the course completing the following points should be confident on passing the course with high grades.

- UNIQUE BONUS: Reporting the data collection process and the structure of the mined data in the following LATEX template [4] (You can open it directly in Overleaf for instance).

[3] https://filesender.funet.fi

[4] https://www.overleaf.com/latex/templates/acm-conference-proceedings-primary-article-template/wbvnghjbzwpc