# Assessment 2: Brownfield

# Development - Continuous Integration Report A

## Cohort 1 Team 2

**Team Members:**
**Trace Chinelle**
**Vidhi Chohan**
**Apollo Cowan**
**Siyuan Liu**
**Aryaman Marathe**
**Charlie Mason**

# Continuous Integration Methods and Approach

## CI Pipelines Used

1) **Automated Build and Testing Pipeline**
    - **Input:** Commits and pull requests targeting the main branch
    - **Processes:**
        - Utilises a GitHub Actions workflow configured with Gradle to automate the compilation of the project that uses dependency caching to optimise execution time
        - The build task ensures that all modules and their dependencies re compiled correctly and checked for errors
    - **Outputs:**
        - Test results, including failures logged in the GitHub Actions interface
        - Creates a dependency graph to monitor for outdated dependencies
    - **Trigger:** Automatically triggered by commits and pull requests targeting the main branch
    - **Appropriateness:** Ensures every change can and will integrate seamlessly, reducing likelihood of bugs in the main branch
2) **Artifact Management and Binary Distribution**
    - **Input:** Successful build outputs
    - **Processes:**
        - Packages the code into distributable JAR file
        - Uploads the JAR to a designated releases page
    - **Outputs:** A ready-to-download binary accessible to developers
    - **Trigger:** Runs upon command after a successful build
    - **Appropriateness:** Simplifies distribution and allows easy access to latest working versions
3) **Release Management**
    - **Input:** Tagged commits (e.g using dates, ‘mark’, ‘patch’)
    - **Processes:**
        - Runs a build and testing workflow to ensure the release is stable
        - Creates a new version in the repository with the binary attached
    - **Outputs:** A release version available on the repository
    - **Trigger:** Automatically triggered by tagging a commit with a version label
    - **Appropriateness:** Guarantees stable releases and provides a clear version history

## Justification for CI Approach

- **Tooling:** GitHub Actions was selected as the CI platform for its seamless integration with our version control system and extensive library of reusable actions
- **Efficiency:** Automated processes reduce manual work and allowed developers to focus on feature implementation and bug fixes