Tools and Standards - Group 2

Introduction:

This document describes the various development tools and standards used throughout the project. These tools support implementation, collaboration, documentation, and testing.

Tools

- **Programming Language**
    - Java
- **Integrated Development Environment (IDE)**
    - Eclipse
- **Build Management**
    - Maven
- **Version Control**
    - GitHub
- **User Interface (UI) Framework**
    - JavaFX

Documentation Standards

- **Structure and Content**
    - Documentation must cover all items explicitly required in the project assignment PDF
    - Each section must provide sufficient detail such that a team unfamiliar with the project could understand, maintain, and extend the software.
- **Formatting**
    - Use Times New Roman, 12-point font.
    - Standard 1-inch margins on all sides.
    - 1.5 line spacing.
    - Include a linked Table of Contents for easy navigation through sections.
- **Traceability**
    - Each requirement must have a unique identifier (e.g., 1.2.0, 2.3.1) to allow tracing through design, implementation, and testing.
    - Source code and test cases must reference the requirement identifiers they satisfy.
- **Citations and References**
    - All external sources, standards, algorithms, or frameworks must be cited in APA format.

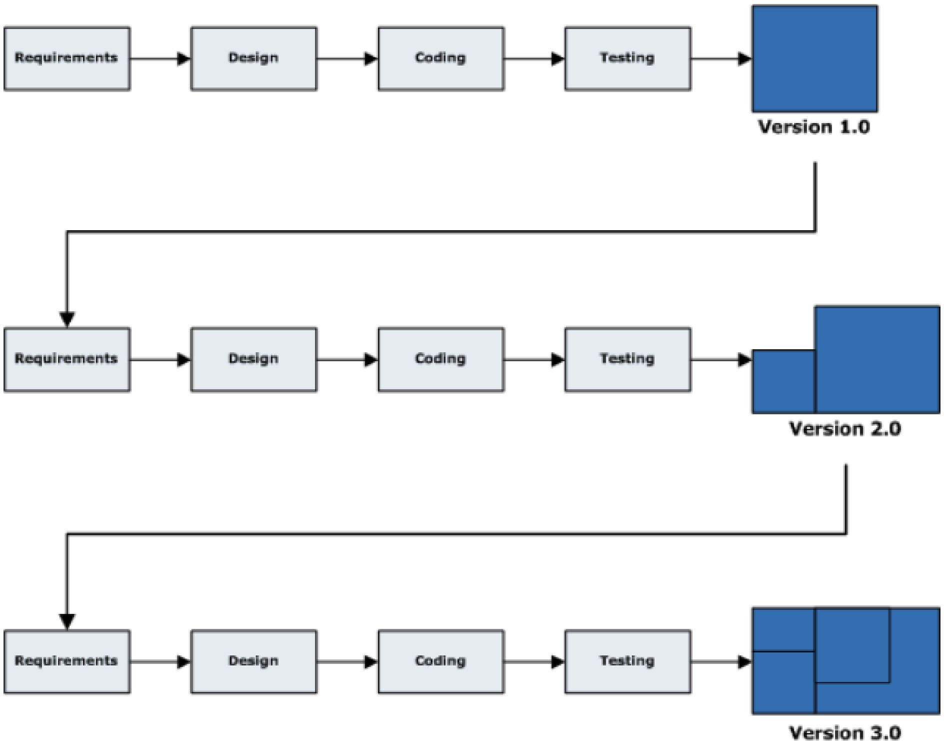
    - Include a complete References section at the end of each document.
- **Professional Presentation**
    - Maintain consistent formatting, headings, and numbering throughout.
    - Figures, diagrams, tables, and screenshots should be clearly labeled and referenced in text.
    - Avoid typographical errors, incomplete sections, or placeholder content.
- **Document Completeness**
    - Documentation should allow a team unfamiliar with the project to understand the system requirements and build or maintain the software without further assistance.
    - All mandatory deliverables listed in the assignment PDF must be fully addressed.
- **Submission**
    - All documents will be submitted using Word unless otherwise specified.
    - Supporting diagrams, spreadsheets, or other files must be embedded or properly referenced.

Coding Standards

- **Structure and Organization**
    - Follow standard object-oriented design principles.
    - Each class must have a well-defined responsibility.
    - Packages must be organized by functionality.
- **Naming Conventions**
    - PascalCase for class names (ex. GroupProject)
    - camelCase for variables and methods (ex. completeProject())
- **Comments**
    - All public classes and methods must include Javadoc comments.

Process Model

- **Incremental (Iterative) Model**



The incremental model is an iterative approach where the system is developed and delivered in several increments, which include:

1. **Requirements**
    a. Define the requirements for the increment, including functionality features and constraints that apply. Requirements are to be reviewed before development begins.
2. **Design**
    a. Develop or design the defined features using Java or JavaFX. This step includes defining class structures, object relationships, or user interface layout. Design features are to be documented before implementation.
3. **Coding**
    a. Implement the defined features in Java. Maven is used to maintain project structure. Code is written following established Java coding standards to ensure readability and consistency throughout the project.

4. **Testing**
    a. Test the features to ensure it meets requirements. Any errors identified during testing will be corrected before completing the increment.