******Style Guidelines for Final Year Project ReportsSmartWebCraft**

**Software Product Quality Metrics**

**Assignment No. 03**

BSc. (Hons.)BS in Software Engineering

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Group Members** | | | | | | |
| Sr.# | Registration ID- Sec | Name | CGPA | Email ID | Phone # | Signature |
| (i) | FA21-BSE-133-A | Aoun Haider | 0 | [FA21-BSE-133@cuilahore.edu.pk](mailto:FA21-BSE-133@cuilahore.edu.pk) | XXXXXX |  |
| (ii) | FA21-BE-127-A | Affan Ahmad | 0 | [FA21-BSE-127@cuilahore.edu.pk](mailto:FA21-BSE-127@cuilahore.edu.pk) | XXXXXX |  |
| (iii) | FA21-BSE-007-A | Muhammad Umer Atiq | 0 | [FA21-BSE-007@cuilahore.edu.pk](mailto:FA21-BSE-007@cuilahore.edu.pk) | XXXXXX |  |
| (iv) | FA21-BSE-152-A | Talha Shafique | 0 | [FA21-BSE-152@cuilahore.edu.pk](mailto:FA21-BSE-152@cuilahore.edu.pk) | XXXXXX |  |

Department of Computer Science

**COMSATS University Islamabad, Lahore Campus**

31 May 2024

Software Quality Metrics

SmartWebCraft

Project Code: SWC001

Internal Advisor: Talha Shafique

External Advisor: Muhammad Umer Atiq

Project Manager: Aoun Haider

Project Team:

Submission Date: 31 May 2024

Note: Task performed by each member has been mentioned in the format [FA21-BSE-XXX] before the start of the work.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Manager’s Signature

[FA21-BSE-121]

# Introduction

## Overview of SmartWebCraft

SmartWebCraft is an innovative web application development platform designed to simplify the development process by allowing programmers to use a visual drag-and-drop interface. This tool integrates advanced technologies such as AI and Machine Learning to enhance the user experience by providing smart suggestions and optimizing system performance. Unlike existing solutions such as IntelliJ and Microsoft Visual Studio, SmartWebCraft offers responsive design capabilities through integration with Bootstrap 5 and Tailwind CSS, ensuring the creation of user-friendly and visually appealing web applications.

## Significance of Software Quality Metrics

Software quality metrics are crucial in evaluating and ensuring the quality of software products. They provide quantitative data that helps in assessing various attributes of the software, such as functionality, reliability, usability, efficiency, maintainability, and portability. By applying these metrics to SmartWebCraft, we aim to identify areas for improvement, ensure high performance, and enhance user satisfaction.

## Objectives of Assignment

The primary objective of this assignment is to perform a thorough analysis of SmartWebCraft using software quality metrics. This analysis will help in:

* Identifying strengths and weaknesses in the software.
* Providing actionable insights for improvement.
* Ensuring that SmartWebCraft meets high-quality standards and user expectations.

# Project Description

SmartWebCraft is developed to streamline web application development by reducing the need for extensive coding. The platform allows users to drag and drop components such as buttons, text fields, and other UI elements to create their applications. It supports backend management via PHP and utilizes MongoDB for object-oriented database management. The project is designed to be incremental, allowing continuous improvements and updates.

## Core Functionalities and Features

* **Drag and Drop Interface:** Users can build web applications by dragging and dropping pre-built components.
* **AI and ML Integration:** Provides smart suggestions for UI/UX design, optimizing resource usage, and ensuring data integrity.
* **Responsive Design:** Integration with Bootstrap 5 and Tailwind CSS ensures that applications are responsive and visually appealing.
* **Backend Management:** Supports backend development via PHP and uses MongoDB for database management.
* **Heterogenous Architecture:** Combines functionalities from tools like WordPress, IntelliJ, and Microsoft Visual Studio.

## Comparison with Existing Tools

Compared to IntelliJ and Microsoft Visual Studio, SmartWebCraft offers enhanced features such as:

* Visual programming interface with drag-and-drop capabilities.
* Integrated AI for smart suggestions and resource optimization.
* Built-in support for responsive design using modern CSS frameworks.
* Lightweight operation with AI-driven resource management.

## AI and ML Integration

SmartWebCraft leverages AI and ML to provide several key features:

* **Smart Suggestions:** AI suggests UI components and design improvements based on user selections.
* **Resource Optimization:** AI ensures efficient use of system resources, putting the system to sleep when not in use.
* **Data Integrity:** AI maintains log files and supports cloud backups for data integrity and fast access.

## Enhancements in Responsiveness and Resource Optimization

By integrating Bootstrap 5 and Tailwind CSS, SmartWebCraft ensures that applications are responsive across different devices and screen sizes. The AI-driven resource optimization further enhances performance by efficiently managing system resources and reducing unnecessary load.

[FA21-BSE-133]

# Software Quality Metrics Overview

Software quality metrics are quantitative measures used to assess various aspects of software quality. These metrics help in evaluating the functionality, reliability, usability, efficiency, maintainability, and portability of software products. By providing objective data, quality metrics enable developers to identify areas for improvement, ensure high performance, and enhance user satisfaction.

## Categories of Software Quality Metrics

* **Product Quality Metrics:** Assess the inherent qualities of the software product, such as functionality, reliability, usability, efficiency, maintainability, and portability.
* **In-Process Quality Metrics:** Measure the quality of the development process, including defect density, code churn, review effectiveness, test coverage, and build success rate.
* **Maintenance Quality Metrics:** Evaluate the ease of maintaining and updating the software, including metrics like MTTR, MTBF, maintainability index, and change request completion rate.

## Selection of Relevant Metrics for SmartWebCraft

The selection of metrics for SmartWebCraft is based on its specific features and functionalities. Relevant metrics include:

* **Functionality Metrics:** Functional completeness, correctness, and appropriateness.
* **Reliability Metrics:** Maturity, availability, and fault tolerance.
* **Usability Metrics:** Learnability, operability, and user error protection.
* **Efficiency Metrics:** Time behaviour and resource utilization.
* **Maintainability Metrics:** Modularity, reusability, and analysability.
* **Portability Metrics:** Adaptability and installability.

## Selecting Specific Metrics

The criteria for selecting specific metrics include:

* **Relevance to the Project:** Metrics should directly relate to the features and functionalities of SmartWebCraft.
* **Measurability:** Metrics should be quantifiable and provide objective data.
* **Actionability:** Metrics should provide insights that can lead to actionable improvements.

## Applying Functional Metrics

### Functional Completeness

1. **Description and Relevance**

Functional completeness measures the extent to which the software includes all the required functionalities.

1. **Application to SmartWebCraft**

Ensuring that all expected features, such as drag-and-drop components, AI suggestions, and responsive design, are fully implemented.

1. **Measurement Methods**

* Feature checklist
* User feedback
* Requirements traceability matrix

### Functional Correctness

1. **Description and Relevance**

Functional correctness assesses the accuracy of the software in performing its intended functions.

1. **Application to SmartWebCraft**

Verifying the accuracy of drag-and-drop components and AI-generated suggestions.

1. **Measurement Methods**
   * Automated testing
   * User acceptance testing
   * Defect tracking

### Functional Appropriateness

1. **Description and Relevance**

Functional appropriateness evaluates how well the software's functionalities align with user needs and expectations.

1. **Application to SmartWebCraft**

Ensuring that AI suggestions and UI components meet user expectations and enhance the user experience.

1. **Measurement Methods**
   * User surveys
   * Usability testing
   * Contextual inquiry

## Reliability Metrics

### Maturity

1. **Description and Relevance**

Maturity measures the stability and reliability of the software over time.

1. **Application to SmartWebCraft**

Assessing the stability of the platform during development and usage.

1. **Measurement Methods**

* Historical defect data
* System logs
* Stability testing

### Availability

1. **Description and Relevance**

Availability evaluates the extent to which the software is operational and accessible when required.

1. **Application to SmartWebCraft**

Ensuring high system uptime and reliable access to the platform.

1. **Measurement Methods**

* Uptime monitoring
* Availability reports
* User access logs

### Fault Tolerance

1. **Description and Relevance**

Fault tolerance assesses the software's ability to continue operating in the presence of errors or failures.

1. **Application to SmartWebCraft**

Evaluating how well SmartWebCraft handles user errors and system crashes.

1. **Measurement Methods**

* Fault injection testing
* Error handling logs
* User error recovery reports

## Reliability Metrics

## Learnability

1. **Description and Relevance**

Learnability measures how easy it is for new users to learn and use the software.

1. **Application to SmartWebCraft**

Assessing the ease with which new users can learn to use the drag-and-drop interface and AI features.

1. **Measurement Methods**

* User onboarding metrics
* Training completion rates
* User feedback surveys

## Operability

1. **Description and Relevance**

Operability evaluates the ease of use and user interface design of the software.

1. **Application to SmartWebCraft**

Ensuring that the user interface is intuitive and easy to navigate.

1. **Measurement Methods**

* Usability testing
* User interface evaluations
* Task completion rates

## User Error Protection

1. **Description and Relevance**

User error protection assesses the software's ability to prevent and correct user errors.

1. **Application to SmartWebCraft**

Implementing features that prevent user errors and provide easy recovery options.

1. **Measurement Methods**

* Error rate tracking
* User feedback on error messages
* Usability testing

## Efficiency Metrics

## Time Behaviour

1. **Description and Relevance**

Time behavior measures the response time and processing speed of the software.

1. **Application to SmartWebCraft**

Evaluating the system's response time and performance under different loads.

1. **Measurement Methods**

* Performance testing
* Load testing
* Response time monitoring

## Resource Utilization

1. **Description and Relevance**

Resource utilization assesses the efficiency of the software in using system resources.

1. **Application to SmartWebCraft**

Ensuring efficient use of system resources, such as memory and CPU, through AI-driven optimization.

1. **Measurement Methods**

* Resource usage monitoring
* Performance profiling
* Efficiency benchmarks

## Maintainability Metrics

## Modularity

1. **Description and Relevance**

Modularity evaluates the degree to which the software's components are separated and independent.

1. **Application to SmartWebCraft**

Ensuring that the platform is modular, allowing for easy updates and maintenance.

1. **Measurement Methods**

* Code analysis
* Module independence tests
* Maintainability index

## Reusability

1. **Description and Relevance**

Reusability assesses the extent to which software components can be reused in different contexts.

1. **Application to SmartWebCraft**

Ensuring that UI components and code can be reused across different projects.

1. **Measurement Methods**

* Code reuse metrics
* Component reuse rates
* Developer feedback

## Analysability

1. **Description and Relevance**

Analysability measures the ease with which software issues can be diagnosed and resolved.

1. **Application to SmartWebCraft**

Ensuring that issues can be quickly identified and resolved through effective logging and diagnostic tools.

1. **Measurement Methods**

* Debugging efficiency
* Issue resolution time
* Diagnostic tool effectiveness

[FA21-BSE-152]

## Portability Metrics

## Adaptability

1. **Description and Relevance**

Adaptability measures the software's ability to run on different platforms and environments.

1. **Application to SmartWebCraft**

Ensuring that the platform can operate across various operating systems and devices.

1. **Measurement Methods**

* Cross-platform testing
* Environment compatibility checks
* User feedback on different platforms

## Installability

1. **Description and Relevance**

Installability assesses the ease with which the software can be installed and configured.

1. **Application to SmartWebCraft**

Ensuring that the platform can be easily installed and set up by users.

1. **Measurement Methods**

* Installation success rates
* User feedback on installation process
* Installation time metrics

## In-Progress Quality Metrics

## Adaptability

1. **Description and Relevance**

Defect density measures the number of defects per unit size of the software.

1. **Application to SmartWebCraft**

Tracking the number of defects found during development and testing.

1. **Measurement Methods**

* Defect tracking
* Code analysis
* Test results

## Code Churn

1. **Description and Relevance**

Code churn assesses the frequency of code changes during development.

1. **Application to SmartWebCraft**

Monitoring the rate of code changes to ensure stability and quality.

1. **Measurement Methods**

* Version control analysis
* Code change logs
* Developer reports

## Review Effectiveness

1. **Description and Relevance**

Review effectiveness measures the efficiency of code reviews in identifying and resolving issues.

1. **Application to SmartWebCraft**

Ensuring that code reviews are thorough and effective in catching errors.

1. **Measurement Methods**

* Review coverage metrics
* Review defect rates
* Developer feedback

## Test Coverage

1. **Description and Relevance**

Test coverage assesses the extent to which the software is tested.

1. **Application to SmartWebCraft**

Ensuring that all features and components are adequately tested.

1. **Measurement Methods**

* Code coverage tools
* Test case analysis
* Test results

## Build Success Rate

1. **Description and Relevance**

Build success rate measures the frequency of successful builds during development.

1. **Application to SmartWebCraft**

Monitoring the success rate of builds to ensure stability and quality.

1. **Measurement Methods**

* Build logs
* Continuous integration reports
* Developer feedback

## Maintenance Quality Metrics

## Mean Time to Repair (MTTR)

1. **Description and Relevance**

MTTR measures the average time taken to repair issues in the software.

1. **Application to SmartWebCraft**

Tracking the time taken to resolve defects and issues.

1. **Measurement Methods**

* Issue resolution logs
* Repair time analysis
* Developer reports

## Mean Time Between Failures (MTTR)

1. **Description and Relevance**

MTBF measures the average time between software failures.

1. **Application to SmartWebCraft**

Monitoring the time between system failures to ensure reliability.

1. **Measurement Methods**

* Failure logs
* System monitoring
* Reliability testing

## Maintainability Index

1. **Description and Relevance**

The maintainability index evaluates the ease of maintaining the software.

1. **Application to SmartWebCraft**

Assessing the overall maintainability of the platform.

1. **Measurement Methods**

* Code analysis tools
* Developer feedback
* Maintenance logs

## Change Request Completion Rate

1. **Description and Relevance**

Change request completion rate measures the rate at which change requests are completed.

1. **Application to SmartWebCraft**

Tracking the completion rate of change requests to ensure responsiveness to user needs.

1. **Measurement Methods**

Change request logs

Developer reports

User feedback

## Methodology

## Selection of Tools for Metrics Collection and Analysis

To effectively measure and analyze the quality metrics for SmartWebCraft, we will use the following tools:

***SonarQube:*** For code quality analysis, including maintainability and defect density.

***JUnit:*** For automated testing and test coverage analysis.

***Selenium:*** For usability and functionality testing through automated UI tests.

***JIRA:*** For defect tracking and change request management.

***Google Analytics:*** For monitoring user behavior and performance metrics.

## Data Collection Procedures

Data will be collected through various means:

**Automated Testing:** Using JUnit and Selenium to run automated tests and collect performance and functionality data.

**User Feedback:** Conducting surveys and usability tests to gather user feedback on learnability, operability, and error protection.

**System Monitoring:** Using tools like Google Analytics to monitor resource utilization, response times, and availability.