Software Quality Assurance Plan

Planr, an Agile Project Planning Application

Version 1.0

Submitted in partial fulfillment of the requirements of the degree of Master of Software Engineering

Prepared by

Michael Blakeman

CIS 895 – MSE Project

Kansas State University

Table of Contents

[1. Purpose 3](#_Toc81647829)

[2. Reference Documents 3](#_Toc81647830)

[3. Management 3](#_Toc81647831)

[3.1 Organization 3](#_Toc81647832)

[3.2 Responsibilities 3](#_Toc81647833)

[3.3 Tasks 4](#_Toc81647834)

[4. Documentation 4](#_Toc81647835)

[5. Standards, Practices, Conventions, and Metrics 4](#_Toc81647836)

[5.1 Documentation Standard 4](#_Toc81647837)

[5.2 Coding Standard 5](#_Toc81647838)

[5.3 Commentary Standards 5](#_Toc81647839)

[5.4 Documentation 5](#_Toc81647840)

[5.5 Metrics 5](#_Toc81647841)

[6. Reviews and Audits 5](#_Toc81647842)

[7. Test 5](#_Toc81647843)

[8. Tools, Techniques, and Methodologies 5](#_Toc81647844)

[9. Code Control 6](#_Toc81647845)

[10. Records Collection, Maintenance, and Retention 6](#_Toc81647846)

# Purpose

This document provides a guideline for ensuring that the Planr macOS application meets a high-quality standard. This is the initial Software Quality Assurance (SQA) Plan.

# Reference Documents

* Vision Document
* Project Plan
* IEEE Guide for Software Quality Assurance Planning
* IEEE Standard for Software Quality Assurance Planning

# Management

## 3.1 Organization

Supervisory Committee

* Dr. Mitchell Neilsen
* Dr. Daniel Andresen
* Dr. Arslan Munir

Major Professor

* Dr. Mitchell Neilsen

Developer

* Mike Blakeman

Formal Technical Inspectors

* Eric Haslag
* Austin Gray

## 3.2 Responsibilities

**Supervisory Committee**

The supervisory committee will be responsible for attending the presentations given by the developer. After each presentation the committee members will provide feedback and suggestions regarding the software.

**Major Professor**

The major professor will be responsible for supervisory committee duties as well as meeting with the developer on a weekly basis to evaluate progress and provide suggestions.

**Developer**

The developer is responsible for all documentation and software development tasks for the software. The project plan describes the tasks to be completed. The developer will also meet with the major professor on a weekly basis to report progress.

**Formal Technical Inspectors**

The formal technical inspectors will be responsible for a technical review of the architecture design artifacts and the formal requirements specifications. They are required to submit a report on their findings.

## 3.3 Tasks

All tasks to be performed have been documented in the Project Plan 1.0 document. This will be reviewed after the first phase to incorporate any necessary changes. A Gantt chart is included in the project plan to provide a schedule for each task.

# Documentation

The documentation will consist of a vision document, project plan, software quality assurance plan, formal requirements specification, architecture design, test plan, formal technical inspection, prototype, user manual, component design, source code, assessment evaluation, project evaluation, references, formal technical inspection letters. All documentation will be reviewed by the committee members for final approval. All documentation will be posted on the developers web site at <https://k-state.instructure.com/eportfolios/3125/Home>.

# Standards, Practices, Conventions, and Metrics

## 5.1 Documentation Standard

The IEEE standards will be used as the documentation guideline.

## 5.2 Coding Standard

The project will use traditional object-oriented analysis and design methods. Recommended Swift, SwiftUI, and Objective-C guidelines will be followed.

## 5.3 Commentary Standards

* Code comments must be used in the project to promote readability as well as implementation details for any interfaces created.
* All public methods must properly document input and output.
* All private methods must properly document the purpose of the method.
* Each file, or module, must contain the author’s name, date it was created, and purpose of the file.

## Documentation

All code comments must use proper Xcode Markup formatting.

## Metrics

Basic COCOMO will be used to estimate the effort and time for the project.

# Reviews and Audits

The committee members will be conducting reviews of the documentation as well as evaluating the developer’s work at each presentation. The committee will also comment on the software prototype demonstration to suggest changes and additions to the requirements specifications. Austin Gray and Eric Haslag will evaluate the architecture design artifacts and report on their findings.

# Test

The Test Plan will cover all testing documentation. Please refer to the Test Plan document.

# Tools, Techniques, and Methodologies

The following tools will be used for coding, testing, and documentation.

* Swift, SwiftUI, and Objective-C programming languages
* Xcode IDE
* XCTest Framework for unit testing
* Microsoft Word for documentation
* Microsoft Project for project planning
* Microsoft Excel for documenting Time Log
* Visual Paradigm for UML 7.0 for UML diagrams

# Code Control

Project artifacts produced using Xcode IDE (primarily source code, configuration files, and tests) will be kept under version control using a Git repository, remotely hosted on GitHub.

Project artifacts produced using the other defined tools in Section 8. Will be kept under version control using the same Git repository, remotely hosted on GitHub.

# Records Collection, Maintenance, and Retention

All project documentation (see 11. Deliverables below) will be available at <https://k-state.instructure.com/eportfolios/3125/Home> when completed. For access to the most current version of Planr artifacts please contact Dr. Mitchell Neilsen.