Table of Contents

List of Tables

1. Introduction

This report gives a brief explanation of the Software Quality Assurance Plan (SQAP) for transit electrification simulation software project. It is well organized into goals, commitments, abilities, activities containing the standards and procedures that administrators may use review and perform the audit for software products and activities to verify that the software meets standards. In addition, it includes all supporting processes that provide the independent assurance that all work products, activities, and processes comply with the predefined plans.

1.1. Purpose

This report summarizes the results of the test performed on the Transit Electrification Simulation Software as stated in this document. The different input gives us different results and according to the standards for building the software its conformance to specifications. So, the test cases and test plan will be used to evaluate the whole simulation software working with reliable results.

- This SQAP encompasses the entire software development process which is used in developing the simulation software.

- This software development process further includes requirements definition, software design, coding, code reviews, source code control, software configuration management, testing, release management and product integration.

- Analysts use application technology to achieve high-quality specifications and designs.

- It identifies the responsibilities of the SQA Planner and Developer.

- Tests are created by developers and/or dedicated testers for each significant use case. Standards violations are identified and addressed throughout development.

- Product or program managers ensure that feature, architecture, and component changes are made only after appropriate review.

- It gives information on which product and process are going to be reviewed by the SQA member.

- Our team monitors problems related to software quality through meetings.

2. Testing

2.1. Test plans

This report includes During Post Execution various tests were performed as follows:

- **Unit Testing:** This testing will ensure that each performs the designated function.

- **Integration Testing:** A systematic technique can also be conducted to construct the software architecture and conduct tests to uncover the errors associated with interfacing. Bottom-up and top -down are the integration types used in our project.

- **Validation Testing:** This testing in our project focuses on software requirements, user-visible actions and output from the system Validation succeeds when software performs functions in a manner that can be reasonably expected by the customer (SRS).

- **System Testing:** This testing is conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. By taking its input, all the integrated components that have passed integration testing.

- **Debugging:** The process is also called a diagnostic process which identifies and removes errors from computer hardware or software per project.

- **Performance testing**: It is a practice which strives to build performance standards into the implementation, design, and architecture of a system. It is also in general a testing practice. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage that can be reasonably expected by the customer (SRS).

  There is no separate test plan that exists which is taken into consideration taken in for reference.

3. Test Assessment

The test assessment gives information about the extent to which the component or system under test meets the requirements that guided its design and development and responds correctly to all kinds of inputs taken into consideration.

There are some minor bugs which is acceptable but with major errors in the simulation software has detected by using the test cases in the algorithm and rework has been performed by the testing team to get reliable results. The system is tested on both front – end and back – end with detailed review of the cases and under given circumstances how well the system works on both ends according to the stated inputs by the user. It facilitates information to stakeholders about the quality of the simulation software product under test by investigation which can be carried as soon as executable software even if partially complete exists.

4. Test Results

There are various tests which are conducted to expose the errors and detailed results are shown in the following Table 1. The passed test does not expose any data integrity and function error but some fail tests require assessment and need to do re-work to make the algorithm run smoother and give the optimal results. Therefore, most crucial test was the algorithm test in which the reliability of results is tested and to avoid such wrong results produced by software is not acceptable.

4.1. Unit/Integration/Debugging/System Testing

We have done unit, integration, debugging, system testing during and after the development of the simulation software to expose the errors in the trip and scheduling algorithm. Therefore, debugger is also used to expose the front - end errors in the graphical user interface (GUI) through the functions that lie in the script. The build of the system is checked after the release and timely updated the local server to avoid any further errors.

Test Case ID	Tester	Effect algorithm	Test Run	Error Description	Pass / Fail
T – 01	Dhaval	None	Initial run of application	-	Pass
T – 02	Himal	None	Selecting the charger and bus model	-	Pass

Test Case ID	Tester	Effect algorithm	Test Run	Error Description	Pass/Fail
T – 03	Dhaval	None	Enter the unit cost of charger and bus model	-	Pass
T – 04	Dhaval	Low	Display the selected charger models from selected brand	It shows the default charger models specified.	Fail
T – 05	Dhaval	High	Pre – configured plan using optimal solution algorithm	It shows only one battery capacity solution not according to selection	Fail
T – 06	Himal	Medium	Cost change according to battery selection and charger capacity	Wrong calculation in the algorithm	Fail
T – 07	Dhaval	None	Show Schedule for first bus with its corresponding charging schedule	-	Pass
T – 08	Dhaval	None	Showing the schedule for all the buses with charger schedule	-	Pass
T – 09	Himal	High	The schedule always displayed without clicking on show schedule	JavaScript debugger found the error in scripting of function	Fail
T – 10	Himal	None	Update the data to generate different schedule	-	Pass

Table 1 Test Cases of Simulation Software

5. Variances

The test cases will be created by the tester conducted in the scripts and algorithms of the simulation software which are used in current software corporates and these test cases are enough to test all the defined cases including major flaws and errors in the system. However, the test plan which is stated is executed in various phases by the team of testers which is involved and most important thing that completes the verification and validation of the system in the first place.