MSSE SOFTWARE, INC.

Test Plan for

GolfScore

Contents

1.0   Introduction

1.1. Objective

The Test Plan is an aggregation of information, which describes the entire test activity for this project. It covers the entire testing effort (unit, development test, system verification test, and Beta). It identifies the product requirements, schedules, resource requirements (people, effort and equipment), quality, assumptions, exclusions, and risks.

A preliminary Test Plan is prepared for the Project Team during the System Phase of PEAQ Process. This Test Plan will be updated in the earliest possible time of the Implementation Phase, so that progress can be tracked during implementation.

1.2. Project Description

GolfScore is a program used to generate reports of golfers' results for a golf tournament. The input to the program will consist of a file containing two types of records. The output from the program will consist of up to 3 reports. The program is executed via a command line interface – there is no GUI associated with the application. The program will be run as a stand-alone executable, and can be run from a command line prompt, from within an IDE (Integrated Development Environment), etc. Input to the program will come from an input record file, and output from the program will go to output record files in a format suitable for printing.

1.3. Process Tailoring

This document in intended for the GolfScore testing team, this project in written un either C or C++ and is intended to rum on a PC running Windows 2000 or any later. So, we as team, are going to test hardware and software. This way GolfScore can be related to quality.

The initial process is based on organize, schedule and manage testing effort. After the unit test phase, we can relate that our product es ready for son design verification test and system validation test. Some of the type of testing that we are including are specification, functional, limits and performance.

The references used in this document are:

1. Software Development Process Handbook (no revision, no date)
2. Software Requirements Specification for the Advanced Color Module, May 13, 1999

2.0   Assumptions/Dependencies

We believe that are efforts can be focused on all the requirements specify on the SRS. Assuming that we are short in time the priority requirements to test are going to be the system score points, the command lines in prompt to the executable file and the reports.

We depend on the code completion before one moth to deliver the final product. With a month of testing we can guarantee regression, integration and SVT tests

3.0   Test Requirements

Test Requirements will require the following resources:

- PC capable of running Windows 2000 or any later version.

- Printer

- The program should run independently

- The input to the program must be from a text file

- The program should output record files in a format suitable for printing.

- The program must be run by a command line interface: there is no GUI associated with the application.

- The reports generated must be stored as text files in the directory determined

- If any of the requested output reports already exist, the program should stop and ask the user if the file should be overwritten.

4.0   Test Tools

Various hardware and software test tools apart from the deliverable product are used to assist in the testing process. These include:

- defect control reporting and tracking software (MantisBT)

- web-based test management system (TestLink)

- Software de Project Management Software (ClickUp)

- hardware computer (mouse, keyword, monitor, PC, printer)

5.0   Resource Requirements

System Verification Test will require the following resources:

- PC capable of running Windows 2000 or any later version.

- Printer

- One SVT – DVT test personnel with at least 80% of his/her time available for this effort.

- Real data to compare reports issued

6.0   Test Schedule

The test schedule must be a coordination between the test that we can be doing and the termination of the product. The schedule must be between weeks of testing, this way the test team can determined error or bugs in each test module.

The first week our mean goal is to realize all the unit test with the developer team, prepare some test case for the second week. For the second week we are been doing integration and functional testing. Leaving for the third week the system test, nun-functional testing and the test report. On the last week we planning to have rework and validation of the system.

Test Sequence	Start
1. Test Development and unit test	First week
2. Integration and functional testing.	Second week
3. System test, nun-functional testing and the test report	Third week

4. Rework and validation of the system	Fourth week

7.0 Risks/Mitigation

List of risks/mitigation:

- the performance of the system could be over one minute / Test the program under stress and for many data

- Printer compatibility / Prepare test configuration and brands of printers

- Failure to read the reports / Create tests cases for de reading of the output file

- Cannot enter or read the input file / Testing the input and see the correct message of error on an invalid file

- Overestimated the time spended on development / Create a schedule with time to spare about test rework

8.0 Metrics

The following metrics data will be collected. Some will be collected prior to, and some after product shipment.

Prior to shipment:

Effort expended during DVT, SVT and Regression

# of defects uncovered during DVT, SVT and Regression, and development phase each defect is attributable to

Test tracking S-Curve

PTR S-Curve

After shipment:

# of defects uncovered and development phase each defect is attributable to

9.0 Document history

The references used in this document are:

1. Software Development Process Handbook (no revision, no date)

2. Software Requirements Specification for the Advanced Color Module, May 13, 1999

3. MSSE SOFTWARE, INC. Software Requirements Specification/ Design Document GolfScore Revision 1.1

Appendix A – Detailed Resource Requirements

To estimate the resource, all test activities must be identified and resources needed to accomplish the activities estimated. Detailed estimates will be shown here:

Requirement number	Requirement name	Relevant section(s) of SRS	Notes
1	Ability to generate reports	Section 2.5	See File Management test cases
2	The input to the program will consist of a file	Section 2.4	ACM-03-1, ACM-03-10, etc.
3	The output from the program will consist of up to 3 reports	Section 2.5	See File Management test cases
4	The program is executed via a command line interface	Section 2.2	ACM-03-1, ACM-03-10, etc.
5	The program will be run as a stand-alone executable	Section 2.2	See File Management test cases
6	Managing errors	Section 2.6	ACM-03-1, ACM-03-10, etc.

Appendix B – Detailed Test Schedule

Test case id	Description	Type of Test
1	Test with Non-numeric data where numeric data is expected: the program must stop with an appropriate error message	Functional
2	Test values for Par that are not 3, 4, or 5: the program will stop with an appropriate error message	Functional
3	Any golfer that has two or more records for the same golf course must be ignore, just taking the first record	Functional
4	If any of the requested output reports already exists. The system must be print on screen "File already exists. Do you want to overwrite it? (Y/N)"	Functional
5	The command line ">golf -h", must display help information on the screen	Functional
6	The command line ">golf -c", must generate the Course Report	Functional
7	Test the score for "over par"	Functional
8	Test the delimiter record for an input file with two Course Records	Functional
9	Test the delimiter record for an input file with two Golfer Records	Functional
10	See and validate the Tournament Ranking Report	Validation
11	See and validate the Tournament Ranking Report	Validation
12	Test once executed, GolfScore will complete its processing within one minute.	Performance
13	GolfScore must be executable file	Validation
14	The first field following filename must be assumed to be a directory	Functional

Appendix B.2 Input test

The input is a document with several test case about the format to enter the input file, is in a separate document titled InputFormat.doc. There are a total of 56 test cases defined.

Appendix B.3 Main Test

The Main Test is in the reports, is in a separate document titled Report.doc. There are a total of 72 test cases defined.

Appendix B.4 Regression Test

Regression Testing will consist of a subset of Main Testing based on those areas where problems were found. The Test Case of calculi score Regression Test is in a separate element titled acmreg.doc; a total of 44 test cases are defined.

Appendix B.5 Test Case Procedures

The Test Case Procedures for Entrance Testing and Main Testing are in Procedures.DOC.