

# **MSSE SOFTWARE, INC**

## **Test Plan Document for GolfScore Software (GolfScore)**

Revision 1.1

*8th, December, 2020*

# Contents

## **1. INTRODUCTION**

**3**

### **1. Objective**

**3**

### **2. Project Description**

**3**

### **3. Process Tailoring**

**3**

## **2. ASSUMPTIONS/DEPENDENCIES**

**4**

## **3. TEST REQUIREMENTS**

**4**

## **4. RESOURCE REQUIREMENTS**

**6**

## **5. TEST TOOL**

**6**

## **6. TEST SCHEDULE**

**6**

## **APPENDIX A – DETAILED RESOURCE REQUIREMENTS**

**8**

# 1. Introduction

## 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.

## 2. Project Description

GolfScore is a program used to generate reports of golfers' results for a golf tournament.

## 3. Process Tailoring

Process tailoring entails establishing a relationship between different levels of process definitions or between process requirements and process definitions.

In process tailoring, the way in which an object is altered can be specified by its attributes such as formality: the degree of attention to formal rules, procedures, and standards with which various process objects, such as activities and artifacts are performed, frequency: the number of times activities are performed, granularity: the level of detail with which process objects are defined, and scope: the area that software processes address.

This project would involve some analysis of the relationship between several processes and also some challenges involved in the development processes.

In this project, **functional testing** is the main testing activity carried out. In addition to functional testing, **performance testing, usability testing, regression testing and efficiency testing** are also considered.

**Functional testing** is the investigation of the main purpose of the software. These are the features of the system that cannot be avoided at any cost. For example, the correctness of the program via the type of inputs and outputs needed for the GolfScore software.

**Performance testing** involves how responsive the software is to the inputs supplied. The software should be able to process the tournament scores in a reasonable time frame.

**Usability testing** is necessary to examine how easy the software is in relative to how users perceive it. The GolfScore software should not be too difficult to operate and easy training should allow any user to be able to operate it perfectly.

**Regression testing** would inspect that the previously implemented and tested functional and non- functional features of the GolfScore software still work as expected.

**Efficiency testing** would examine if the GolfScore software is working efficiently. This involves investigating the functionalities specified in the requirement specification document to ensure the software complies with those functionalities.

#### **4. Referenced Documents**

The references used in producing this document are listed below:

1. Software Requirements Specification (SRS)
2. Product Definition Document (PDF)
3. Software Development Plan (SDP)

#### **2. Assumptions/Dependencies**

This software should be implemented in either C or C++ by which failure to comply with this may result in some failures. Also, it is assumed that the software is running on a PC that is running Windows operating system not earlier than 2000 version. Under this section, the tasks to be completed would be outlines together with the conditions we believe to be true to accomplish the tasks. Below are some of the assumptions:

- 1.The software components to be tested shall involve only functional requirements.
- 2.The software under test shall be ensured to be properly installed in the correct platform before carrying out the testing activities.
- 3.Since the testing plan involves only the functional requirements, no performance testing tools are installed before the testing activities.
- 4.During test execution, the system functionalities are not placed on use.

#### **3. Test Requirements**

In according to the Software Requirements Specification document, Product Definition Document and Software Design Document there are the following **general requirements**:

- a) the software shall be implemented via a command line interface.
- b) The software shall conform with the rules of golf tournament.
- c) The implemented software shall be executed on a PC running Windows 2000 or any later version.

The following is the list of requirements that will be tested as part of the current test plan execution. For a full description of the requirements refer to the GolfScore SRS Document.

### Functional Requirements

ID	Requirement	SRS Section
RF-001	Formats for calling GolfScore	2.2
RF-002	Program Functionality - Tournaments assumptions	2.3, 2.3.1
RF-003	Program Functionality - Scoring	2.3, 2.3.2
RF-004	Data Input - will consist of a formatted text file	2.4
RF-005	Data Input - Course Record	2.4.1
RF-006	Data Input - Course Record Delimeter	2.4.2
RF-007	Data Input - Golfer Records	2.4.3
RF-008	Data Input - Golfer Records Delimeter	2.4.2
RF-009	Data Input - GolfScore will generate up to 3 records each as a text file	2.5
RF-010	Data Output - Tournament Ranking Report	2.5.1
RF-011	Data Output - Golfer Report	2.5.2
RF-012	Data Output - Course Report	2.5.3
RF-013	Error Handling - Input Parameter Errors	2.6, 2.6.1
RF-014	Error Handling - Input Data Errors	2.6, 2.6.2
RF-015	Error Handling - Errors On Output	2.6, 2.6.3

### Non-Functional Requirements

ID	Requirement	SRS Section
RNF-001	Identification - the program will display its title and revision number on the screen at execution time	1.1

ID	Requirement	SRS Section
RNF-002	System Architecture - the program is intended to run on a PC running Windows 2000 or any later version	1.3
RNF-003	Deliverables - GolfScore is provided as an executable file	3
RNF-004	Performance - Once executed, GolfScore will complete its processing within one minute	

#### 4. Resource Requirements

1. PC running Windows 2000 or any later version
2. Imitation model of golf-tournament
3. One testing engineer with at least 80% of his/her time available for this effort.

#### 5. Test tool

The list of tools required to perform and/or aid in the testing process is the following:

- Jira (Task management and defect tracking tool).
- Text File editor program

#### 6. Test Schedule

For a detailed view of the Schedule see: Appendix B - Detailed Test Schedule

Task	Start date	End date
Requirements Analysis	08-Dec-2020	9-Dec-2020
Test Planing	9-Dec-2020	14-Dec-2020
Test Case Development	15-Dec-2020	18-Dec-2020
Environment Setup	21-Dec-2020	22-Dec-2020
Test Case Execution	22-Dec-2020	30-Dec-2020
Test Cycle Closure	30-Dec-2020	31-Dec-2020

#### 7. Risks/Mitigation

Some of the risks are the possibility of not delivering the software on time and misinterpretation of the user requirements.

## **8. 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 Size of software

## Appendix A – Detailed Resource Requirements

Requirement ID	Requirement title	SRS Section
RF-001	Invoking GolfScore	2.2
RF-002	Tournament Assumption	2.3.1
RF-003	Tournament Scoring	2.3.2
RF-004, RF-005, RF-006, RF-007, RF-008, RF-009	Data Input	2.4
RF-005, RF-006	Course Records	2.4.1
RF-006, RF-008	Delimiter Records	2.4.2
RF-007, RF-008	Golfer Records	2.4.3
RF-010, RF-011, RF-012	Data Output	2.4.4
RF-010	Tournament Ranking Report	2.5
RF-011	Golfer Report	2.5.1
RF-012	Course Report	2.5.2
RF-013, RF-014, RF-015	Error Handling	2.6
RF-013	Input Parameter Error	2.6.1
RF-014	Input Data Error	2.6.2
RF-015	Errors on Output	2.6.3



## Appendix B - Detailed Test Schedule

Task	Start date and end date	Task Responsible	Task Suporter
Requirements analysis	08-09 Dec 2020	Eleonora Belova	Yulia Doe
Test Planing	9-14 Dec 2020	Eleonora Belova	Yulia Doe
Test Case Development	15-18 Dec 2020	Dmitry Bokov	Eleonora Belova, Yulia Doe, James Rom
Environmental Setup	21-22 Dec 2020	Dmitry Bokov	Yulia Doe
Test Case Execution	22-30 Dec 2020	Yulia Doe	Eleonora Belova, Dmitry Bokov, James Rom
Test Cycle Closure	30-31 Dec 2020	Eleonora Belova	

## Appendix C - Test Cases

Test Case	Description	Type of Test
TC-001	The command line >golf -h c:\in.txt c:\golfout shall be accepted	Functional
TC-002	The command line >golf -a c:\in.txt c:\golfout shall return an error	Functional
TC-003	If input files exist shall be accepted	Functional
TC-004	If input files don't exist shall return an error	Functional
TC-005	If output directory exist shall be accepted	Functional
TC-006	If output directory don't exist shall return an error	Functional
TC-007	The number of golf course '1' shall be accepted	Functional
TC-008	The number of golf course '5' shall be accepted	Functional
TC-009	The number of golf course '0' shall return error	Functional
TC-010	The number of golf course '6' shall return error	Functional
TC-011	The number of golf course '2' shall be accepted	Functional
TC-012	The number of golf course '12' shall be accepted	Functional
TC-013	The number of golf course '1' shall return error	Functional
TC-014	The number of golf course '13' shall return error	Functional
TC-015	The number of par value '3' shall be accepted	Functional
TC-016	The number of par value '4' shall be accepted	Functional
TC-017	The number of par value '5' shall be accepted	Functional
TC-018	The number of par value '2' shall return error	Functional
TC-019	The number of par value '6' shall return error	Functional
TC-020	A text file shall be accepted as data input	Functional
TC-021	A file that is no text shall return error	Functional
TC-022	Course Records at the beginning of the text file shall be accepted	Functional
TC-023	Course Records delimited by a delimiter record shall be accepted	Functional
TC-024	Course Records with numeric value in the columns 21 to 38 shall be accepted	Functional

Test Case	Description	Type of Test
TC-025	Course Records with no numeric value in the columns 21 to 38 shall return error	Functional
TC-026	Tournament ranking report shall display golfer's name	Functional
TC-027	Tournament ranking report shall display golfer's score for each score	Functional
TC-028	Tournament ranking report shall display golfer's total score of the tournament	Functional
TC-029	Tournament ranking report shall display golfer's final standing (1st place, 2nd place, etc.	Functional
TC-030	Tournament ranking report filename shall be thank.rep	Non-Functional
TC-031	The program shall display its title at execution time	Non-Functional
TC-032	The program shall display its revision number at execution time	Non-Functional
TC-033	The program shall run in Windows 2000 PC	Non-Functional
TC-034	The GolfScore program shall run from executable file	Non-Functional
TC-035	The GolfScore program shall run from command line	Non-Functional
TC-036	The GolfScore program shall run from IDE	Non-Functional
TC-037	The GolfScore program shall complete processing of 1 report under 1 minute	Non-Functional