INCEPTA SOFTWARE, INC.

Comprehensive System Test (DVT and SVT) Plan for GolfScore

Revision 1.1

Md. Saidur Rahman Sujon

22, 2024

Contents

1.0 INTRODUCTION 3

1.1. Objective 3

1.2. Project Description 3

1.3. Process Tailoring 3

1.4. Referenced Documents 3

2.0 ASSUMPTIONS/DEPENDENCIES 3

3.0 TEST REQUIREMENTS 4

4.0 TEST TOOLS 4

5.0 RESOURCE REQUIREMENTS 4

6.0 TEST SCHEDULE 5

7.0 RISKS/MITIGATION 5

8.0 METRICS 5

APPENDIX A – DETAILED RESOURCE REQUIREMENTS 6

APPENDIX B – DETAILED TEST SCHEDULE 7

1.0 Introduction

1.1. Objective

This document explains the test plan for GolfScore Release 1.1. It includes information about what is to be tested, how testing is to be performed, and what is not to be tested. This document also contains a description of the testing schedule, tests to be performed, test dependencies, resources required, testing tools, and metrics. Changes in requirements and the structure of the team must always reflect in this document.

The main focus of this test it to verify the software requirements specification (SRS) for the GollfScore program as contained in Appendix A.

1.2. Project Description

GolfScore is a program that generates golf tournament results for golfers at each course. This program accepts an input text file (as stated in the SRS) and returns three output text files (as described in the SRS).

1.3. Process Tailoring

The GolfScore application has no external dependencies. Thus, the test plan is suited to Functional and Non-functional Testing within the context of Design Verification and System Validation. Testing is conducted in the following phases:

- Entrance Test: To ensure that the software executes correctly and handles input parameter errors as stated in the SRS.
  Appendix C describes the Entrance Testing test scenarios, while Appendix A contains the SRS.
- Main Test: To ensure that the software is being executed correctly. To determine whether the software accurately processes the input data as stated and delivers the appropriate output. Furthermore, the program's handling of input data errors and output faults is tested for accuracy.

  Appendix C provides a description of the Main Testing test scenarios.
- Exit Test: To verify if the program produced the required outputs, and saved them in the correct format and in the correct location.
  See Appendix C for Exit Testing test cases.
- Regression Test: After defects must have been identified during testing and processed, all tests are run again to ensure proper behavior.

The following references were used in creating this document:

a. Software Requirements Specification for GolfScore, Revision 1, July 18, 2017.
b. System Verification Test Plan for Advanced Color Module, Revision 2, 22 February, 2000.

2.0 Assumptions/Dependencies

It is assumed that the development team unit test their code while developing the software, and also perform integration testing. Customer validation testing is assumed to be carried out by field personnel together with the customers.

For conformation with the set schedule, the program must be made available by the development team by February 10, 2024.

3.0 Test Requirements

- Entrance Tests:
  - The program runs on a PC running Windows 2000 or any later version.
  - The program is written in either C or C++.
  - The program will run as a stand-alone executable.
  - The program can be run from the command line prompt.
  - The program is run with valid input parameters
- Main Tests:
  - There is a delimiter record that signals the end of the input file.
  - The number of golf courses specified for the tournament must be from 1 to 5.
  - Each golfer is expected to play each course once.
  - The number of golfers entered in the tournament can be from 2 to 12.
  - Par for holes on each course must be either 3, 4, or 5.
  - Score earned by a golfer for each hole played is between 0 and 6 (0 and 6 included).
  - The first set of records in the input file (course records) exist and follow the specified format for each entry.
  - There is a delimiter record that signals the end of course records.
  - A second set of records (golfer records) exist in the input file and each entry follows the specified format.
- Exit Tests:
  - The program should produce a number of reports corresponding to the specified options.
  - The generated reports should be saved as text files in the specified output directory (or if not specified, in the directory of the input file) with the extension ".rep".
  - If requested, the tournament ranking report should contain a list of all golfers in the specified format. The list should be in descending order of final score and should be saved with an output filename of trank.rep.
  - If requested, the golfer report should contain a list of all golfers in the specified format. The list should be alphabetical with respect to the golfers' last name and should be saved with an output filename of golfer.rep.
  - If requested, the course report should contain a section for each Golf Course listed in the input Course Records in the specified format. It should be saved with

4.0 Test Tools

To aid the testing process, the following testing tools are required:

- Defect reporting and tracking software
- Installation media for multiple Windows versions above 2000 (e.g. XP, Vista, 7, 8, 8.1, & 10)

5.0 Resource Requirements

The following resource would be required:

- GolfScore Program version 1.1
- Four PCs capable of hosting virtual machines
- A virtualization software
- Four Test Group personnel with at least 60% of his/her time available for this effort.

See Appendix A for details.

6.0 Test Schedule

No.	Test	Start	Finish
1	Test Development	28.01.2024	10.02.2024
2	Program Availability	10.02.2024	--
3	Entrance Testing	11.02.2024	17.02.2024
4	Main Testing	18.02.2024	28.02.2024
5	Exit Testing	29.02.2024	05.03.2024
6	Regression Testing	06.03.2024	10.03.2024

See Appendix B for details.

7.0 Risks/Mitigation

- Without a software to enforce input data structure conformity, there is a high risk of errors.

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

Size of software

Appendix A – Detailed Resource Requirements

No.	Test	No of Personnel	No of Hours
1	Test Development	4	90
2	Entrance Testing	4	90
3	Main Testing	4	80
4	Exit Testing	3	40
5	Regression Testing	2	40

- PCs that are capable of hosting virtual machines are required such that the program can be tested on multiple versions of Windows.
- A virtualization software is required such that multiple versions of Windows can be installed to test the program.

Appendix B – Detailed Test Schedule

No.	Test	Start	Finish
1	Test Development	28.01.2024	10.02.2024
2	Program Availability	10.02.2024	--
3	Entrance Testing	11.02.2024	17.02.2024
4	Main Testing	18.02.2024	28.02.2024
5	Exit Testing	29.02.2024	05.03.2024
6	Regression Testing	06.03.2024	10.03.2024

No.	Test	Dependencies
1	Test Development	4 PCs 4 Personnel
2	Program Availability	GolfScore Program
3	Entrance Testing	4 PCs 4 Personnel Virtualization Software
4	Main Testing	4 PCs 4 Personnel Virtualization Software
5	Exit Testing	3 PCs 3 Personnel Virtualization Software
6	Regression Testing	2 PCs 2 Personnel Virtualization Software

Appendix C – Test Cases

Test No.	Test Case	Test Type
1	The program shall be written in C or C++	Non-functional
2	The program shall run on a PC running Windows 2000	Non-functional
3	The program shall run on a PC running Windows XP	Non-functional
4	The program shall run on a PC running Windows Vista	Non-functional
5	The program shall run on a PC running Windows 7	Non-functional
6	The program shall run on a PC running Windows 8	Non-functional
7	The program shall run on a PC running Windows 10	Non-functional
8	The program shall run as a stand-alone executable	Non-functional
9	The program shall run from the command line prompt	Non-functional
10	Command line options "-ctg" shall be accepted	Functional
11	Command line option "-c" shall be accepted	Functional
12	Command line option "-t" shall be accepted	Functional
13	Command line option "-g" shall be accepted	Functional
14	Command line options "-c –t -g" shall be accepted	Functional
15	Command line option "-k" shall display an "unrecognizable options" message	Functional
16	Command line option "-j" shall display an "unrecognizable options" message	Functional
16	Command line option "-kj" shall display an "unrecognizable options" message	Functional
17	Command line option "-ckj" shall display an "unrecognizable options" message	Functional
18	Specifying an input filename that does not exist shall display an "input parameter error"	Functional
19	Specifying an output directory that does not exist shall display an "input parameter error"	Functional
20	Command line option "-g" shall be accepted and shall display help information	Functional
21	Calling the program as "golf -ctg in.txt golfout" where "in.txt" exists and is valid and folder "golfout" exists shall be accepted	Functional
22	Calling the program as "golf -ctg in.txt golfout dis" where "in.txt" exists and is valid and folder "golfout" exists shall be accepted	Functional
23	Calling the program as "golf -ctg in.txt golfout" where "in.txt" exists and is valid and folder "golfout" does not exist shall display an "input parameter error"	Functional
24	Calling the program as "golf -ctg in.txt golfout" where "in.txt" does not exist shall display an "input parameter error"	Functional
25	The number of golf course "1" shall be accepted	Functional
26	The number of golf course "5" shall be accepted	Functional
27	The number of golf course "-5" shall return an error	Functional
28	The number of golf course "6" shall return an error	Functional
29	The number of golf course "0" shall return an error	Functional
30	Having multiple records for a golfer on the same golf courses shall be accepted, although a message should be displayed to indicating	Functional

	this. The first record shall be used and processing shall continue.	
31	The number of golfers “0” shall return an error	Functional
32	The number of golfers “1” shall return an error	Functional
33	The number of golfers “2” shall be accepted	Functional
34	The number of golfers “12” shall be accepted	Functional
35	The number of golfers “13” shall return an error	Functional
36	Par for hole “2” shall return an error	Functional
37	Par for hole “6” shall return an error	Functional
38	Par for hole “3” shall be accepted	Functional
39	Par for hole “4” shall be accepted	Functional
40	Par for hole “5” shall be accepted	Functional
41	Golfer score per hole “7” shall return an error	Functional
42	Golfer score per hole “-1” shall return an error	Functional
43	Golfer score per hole “0” shall be accepted	Functional
44	Input data with non-numeric data where numeric data is expected shall return an error	Functional
45	Input data with numeric data where non-numeric data is expected shall return an error	Functional
46	Input data that violates delimiter constraints shall return an error	Functional
47	Input file that does not contain course records shall return an error	Functional
48	Input file that does not contain golfer records shall return an error	Functional
49	Calling the program with command line options “-ctg” shall generate 3 output files: “trank.rep”, “golfer.rep”, “course.rep”. If any of the files already exist, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
50	Calling the program with command line option “-c” shall generate an output file: “course.rep”. If the file already exists, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
51	Calling the program with command line option “-t” shall generate an output file: “trank.rep”. If the file already exists, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
52	Calling the program with command line option “-g” shall generate an output file: “golfer.rep”. If the file already exists, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
53	If output cannot be saved due to insufficient permissions, the program shall display an error.	Functional