MSSE SOFTWARE, INC.

# Test Plan for GolfScore

**revision 1.0**

**Klimov Andrey**

**07/03/2021**

# 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 as described in Section 2.4 Software Requirements Specification (SRS). The output from the program will consist of up to 3 reports as described in Section 2.5 SRS. The program is executed via a command line interface –there is no GUI associated with the application.

## 1.3. Process Tailoring

Please refer to software development and management processes as a guideline XXX.
For MSSE software requirements please see XXX.

Software verification tests are going to be divided into four categories:

-Regression testing.
-Main software testing.
    -Compatibility testing
    -Documentation testing
    -Functional testing
    -Performance testing

# 2.0 Assumptions/Dependencies

To start testing GolfScore software 1.1 the following has to happen:

-Stable version of GolfScore 1.0 or previous stable version has to be provided before the due date.
-Main functionality or complete GolfScore 1.1 has to be available for the QA team before the due date.
-Human resources are available to handle this activity,
-PCs and operating systems are available for the QA team to perform the Comparability testing.
-All the necessary documentation is available for the QA team.

# 3.0 Test Requirements

To verify the test we need following:

-Working GolfScore Realease 1.0 or below with main functionality in place as defined my MSSE requirements.
-PCs with at least Windows 2000, XP, VISTA, Windows 7 and others, the precise scope of operating systems is described in chapter Comparability testing.
-Existing examples of tournaments from previous revisions/years with output and input data if available.
-Any tools and tests from the previous release of the GolfScore software.
-One dedicated member of QA department per one working station.
-At least one hard copy of MSSE software requirements XXX and Design document GolfScore 1.1

# 4.0 Test Tools

Apart from tools requested form requirements:

-IDE compatible with C/C++;
-Quality assurance tracker to store and update tickets when quality issues are raised.
-Operating systems, number of PCs and human resources to be discussed with the development team, management, QA department and MSSE software requirements XXX.

# 5.0 Resource Requirements

Human resource requirements as it is estimated by the QA Engineering team. For this project Quality assurance two full-time QA specialists are required. See Appendix A for details.

Grand total resource requirements for the team of two QA engineers, hours: 2032 h.

# 6.0 Test Schedule

These are the estimates by QA team, the exact numbers are to be determined with cooperation of Development and Management.

| Test sequence | Start | Finish |
|---|---|---|
| 1. Regression testing | 01/06/2021 | 01/07/2021 |
| 2.1 Compatibility testing | 01/07/2021 | 31/07/2021 |
| 2.2 Documentation testing | 01/08/2021 | 31/08/2021 |
| 2.3 Functional testing | 01/09/2021 | 31/10/2021 |
| 2.4 Performance testing | 01/11/2021 | 20/11/2021 |

See Appendix B for details.

# 7.0 Risks/Mitigation

Potential risk might include:

-Main functionality of the software is not provided – hold the testing procedure until all the requirements are met/move planing dates if possible, assign more resources if this is not a feasible solution.

-PCs are not provided, operating systems are not as required – additional costs might be involved in delivering PCs and installing necessary software.

# 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

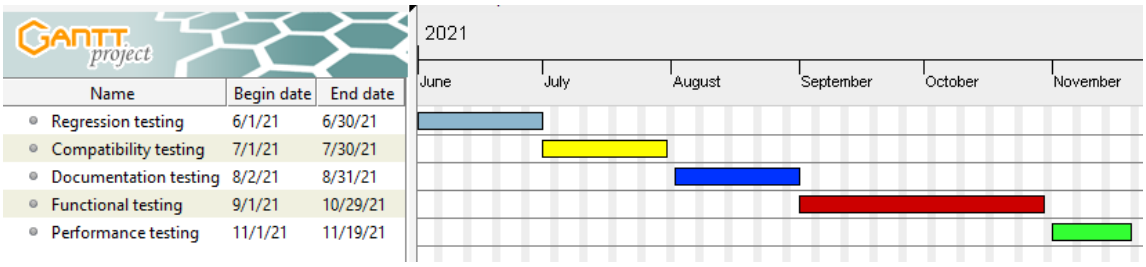
[To estimate the resource, all test activities must be identified and resources needed to accomplish the activities estimated. Detailed estimates will be shown here. This consists of identifying all project test activities by the Test Group and the number of hours estimated to accomplish these activities. Be specific. Show specific responsible test engineer's names, if possible. A grand total of the effort must be shown here, as well as in Section 5.0.]

Initial estimation by QA team is that we need 2 engineers to perform the testing, both from the availability standpoint in case one member is on vacation or disability and from effectivity and speed of the testing.

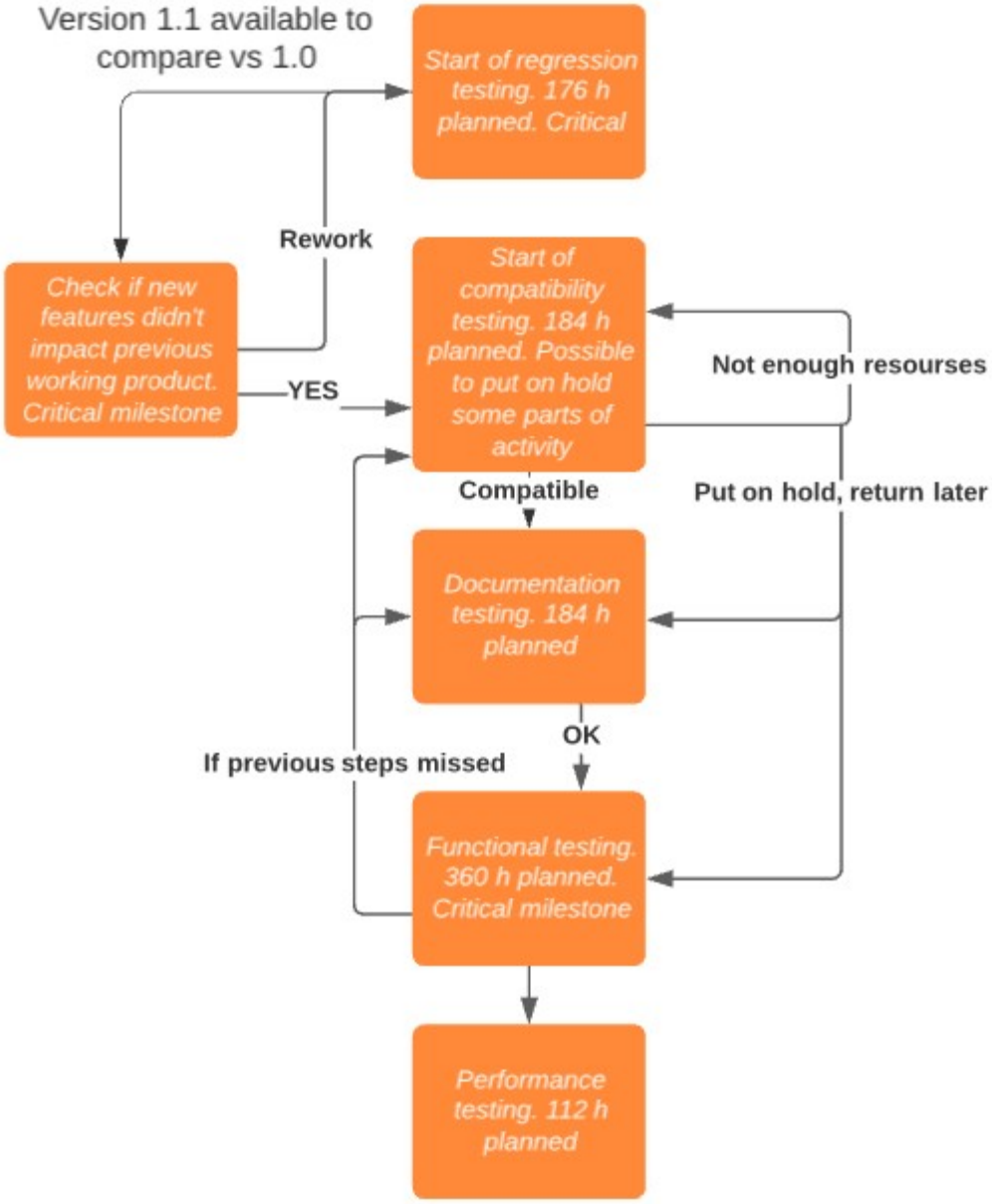
| Test sequence | Scheduled time, hours | Cost for the QA department = Scheduled time * Factor of HR resources |
|---|---|---|
| 1. Regression testing | 176 | 352 |
| 2.1 Compatibility testing | 184 | 368 |
| 2.2 Documentation testing | 184 | 368 |
| 2.3 Functional testing | 360 | 720 |
| 2.4 Performance testing | 112 | 224 |
| Total: | 1016 | 2032 |

This is raw time resources estimation, in case development team is facing some challenges, some activities might be put on hold and continued after some requirements are met.

# Appendix B – Detailed Test Schedule




| Name | Begin date | End date |
|---|---|---|
| Regression testing | 6/1/21 | 6/30/21 |
| Compatibility testing | 7/1/21 | 7/30/21 |
| Documentation testing | 8/2/21 | 8/31/21 |
| Functional testing | 9/1/21 | 10/29/21 |
| Performance testing | 11/1/21 | 11/19/21 |



## Appendix C  Test cases

### Regression stage:

1. Regression tests/previous tests shouldn't affect the current state of the project. Title of the software, revision number have to be checked on this test case.

### Compatibility stage:

1. Software is compatible with all the required versions of operating systems  Windows versions 2000 and above.

### Documentation stage:

1. Check that all the requirements of Software Requirements Specification (SRS),  MSSE software requirements, and other related documents are in line with the test plan and test schedule. Make sure that each critical requirement has at least one corresponding test to it.

### Functional stage:

1. Check for unrecognizable/unstated options for input parameters. Error message has to be present. see SRS 2.6.

2. Check for first filed following options for input parameters (filename). Error message has to be present. see SRS 2.6.1

3. Check for first filed following filename for input parameters (directory). Error message has to be present if directory doesn't exist. see SRS 2.6.1

4. Check that fields beyond the output are ignored. see SRS 2.6.1

5. Check for non-numeric data in numeric fields in the Input file. Error message has to be present if there is non-numeric data exists in those fileds. see SRS 2.6.1

6. Par values 3, 4, 5 do not return an error message. Par values that extend this scope are handled with the error message. Values -1, 0, are checked. see SRS 2.6.1

7. Errors on output are handled as described in 2.6.3 of SRS.

    7.1. Check names of the generated reports. SRS 2.5

    7.2. Check that course/golfer/tournament ranking reports  are in line with SRS 2.5.1-2.5.3 requirements.

11. Check data input as described in SRS 2.4

    11.1 Check name of the input file, check non-latin characters.

    11.2 Check different order of golfer records as described in SRS 2.4.3

12. Check boundaries and different combinations of strokes counts as per SRS 2.3.2

13.Check tournament assumptions as per SRS 2.3

    13.1 Check the program behavior for courses -1, 0, >5 SRS 2.3.1

13.2 Check number of courses -1, 0, >5 values SRS 2.3.1

14. Check that all stated calling expressions are in line with the requirements of SRS 2.2