MSSE SOFTWARE, INC.

**Test Plan for**

**GolfScore**

Confidential and Proprietary Information of Datacard Worldwide

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 3

4.0 Test Tools 4

5.0 Resource Requirements 4

6.0 Test Schedule 4

7.0 Risks/Mitigation 4

8.0 Metrics 4

Appendix A – Detailed Resource Requirements 5

Appendix B – Detailed Test Schedule 6

# Introduction

## 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 describes the testing activities for GolfScore Release 1.1, a software program designed to process golf tournament scores and generate reports. The plan outlines testing objectives, scope, tools, and resources to ensure the software meets its functional and performance requirements.

## Project Description

## GolfScore is a command-line program that processes input files containing tournament data to produce reports on golfer rankings, individual golfer performance, and course-specific results. The program handles up to 5 courses and 12 golfers and outputs formatted text files for printing.

## Process Tailoring

This project employs specification testing, functional testing, limits testing, performance testing, and error-handling validation. Tests are tailored to the GolfScore SRS requirements and include both manual and automated approaches.

## Referenced Documents

* GolfScore Software Requirements Specification (SRS) (Rev. 1.1, July 18, 2017)
* Test Plan Template
* Test Plan Example

# Assumptions/Dependencies

* Assumptions:
  + The input data file format adheres to the structure specified in the SRS.
  + The execution environment is a Windows PC running Windows 2000 or later.
* Dependencies:
  + The executable file for GolfScore must be delivered before testing begins.
  + Test data must be provided in the correct format as per the SRS.

# Test Requirements

**Functional Requirements**

1. Validate the correct parsing of input files, including course records, delimiter records, and golfer records.
2. Ensure accurate calculation of scores and rankings based on the scoring rules.
3. Verify that the three output report types (Tournament Ranking, Golfer Report, and Course Report) are generated correctly.
4. Test the handling of input errors, including file format violations and invalid data.
5. Confirm adherence to performance requirements, completing processing within one minute.

**Non-Functional Requirements**

1. Validate compatibility with the specified operating systems.
2. Ensure the robustness of error-handling mechanisms.

# Test Tools

1. Manual verification tools (e.g., text editor for reviewing output files).
2. Automated testing scripts for validating input/output correctness.
3. Logging tools to monitor execution and errors.

# Resource Requirements

* One Windows PC for executing tests.
* Two test engineers for execution and reporting.
* Sample input data files covering all test scenarios.
* Development support for resolving issues during testing.

# Test Schedule

* **Test Preparation**: 2 days
* **Functional Testing**: 5 days
* **Error Handling Testing**: 2 days
* **Performance Testing**: 1 day
* **Documentation and Reporting**: 1 day
* **Total Duration**: 11 days

# Risks/Mitigation

* **Risk**: Input file formatting errors causing test delays.
  + **Mitigation**: Provide clear examples and templates for input files.
* **Risk**: Limited testing resources.
  + **Mitigation**: Prioritize critical test cases and automate repetitive tasks.

# 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

|  |  |  |
| --- | --- | --- |
| **Resource** | **Quantity** | **Description** |
| Test Engineers | 2 | Responsible for test execution |
| Windows PC | 1 | Test execution environment |
| Test Data Files | N/A | Input files for various test cases |

Appendix B – Detailed Test Schedule

|  |  |  |
| --- | --- | --- |
| Task | Start Date | End Date |
| Test Preparation | Day 1 | Day 2 |
| Functional Testing | Day 3 | Day 7 |
| Error Handling Testing | Day 8 | Day 9 |
| Performance Testing | Day 10 | Day 10 |
| Documentation and Reporting | Day 11 | Day 11 |

**Appendix C - Test Cases**

### Test Case 1: Validate Course Record Parsing

* **Input**: Valid course record.
* **Expected Output**: Parsed course name, identifier, and par values for 18 holes.

### Test Case 2: Validate Golfer Record Parsing

* **Input**: Valid golfer record.
* **Expected Output**: Parsed golfer name, course identifier, and stroke counts for 18 holes.

### Test Case 3: Tournament Ranking Report

* **Input**: Data for 3 courses and 6 golfers.
* **Expected Output**: Tournament Ranking Report in descending order of scores.

### Test Case 4: Input Data Error Handling

* **Input**: File with non-numeric stroke count.
* **Expected Output**: Error message indicating invalid data.

### Test Case 5: Performance Validation

* **Input**: Maximum allowable input size (5 courses, 12 golfers).
* **Expected Output**: Processing completed within one minute.

### Test Case 6: Combined Report Generation

* **Input**: Options for generating all three reports.
* **Expected Output**: Three separate output files with correct content.

### Test Case 7: File Overwrite Prompt

* **Input**: Existing output file in the directory.
* **Expected Output**: Prompt to overwrite file, with correct handling of user response.