Test Plan for GolfScore Release 1.1
Confidential and Proprietary – MSSE Software, Inc.

1.0 Introduction

1.1 Objective

The purpose of this Test Plan is to define the entire testing process for the GolfScore Release 1.1 software product.
It describes all planned testing activities—including unit testing (informal), integration tests, system verification testing (SVT), regression testing, resources, schedules, assumptions, risks, and acceptance criteria.
This plan ensures that all functional and performance requirements stated in the *GolfScore Software Requirements Specification, Rev 1.1* are fully validated.

1.2 Project Description

GolfScore is a command-line application written in C/C++ for Windows 2000 and later. It processes a tournament input file containing course records and golfer-score records and outputs up to three formatted text reports:

- **Tournament Ranking Report**
- **Golfer Report**
- **Course Report**

It also handles input parameter errors, input data errors, and output file overwrite behaviors.

1.3 Process Tailoring

This project follows a lightweight software development process suitable for a small, single-executable application.
Testing will consist of:

- Specification/requirements testing
- Functional and boundary testing
- Error-handling testing
- Performance testing (≤ 1 minute execution requirement)
- Compatibility testing (Windows 2000+ command-line)

The following standard test-process components are **tailored out**, with justification:

Process Component	Excluded?	Rationale
GUI usability tests	Yes	CLI-only program
Load/stress testing	Yes	System processes only one input file per invocation
Network/system integration	Yes	Standalone, no networking

1.4 Referenced Documents

- *GolfScore Software Requirements Specification, Rev. 1.1*
- *TestPlan_Template.docx*

576262b9-c61e-4599-9650-eabef74...

- *TestPlan_Example.pdf* (structure reference only)

TestPlan_Example

2.0 Assumptions and Dependencies

Assumptions

1. Input files conform *generally* to formatting rules (unless intentionally malformed for error testing).
2. The development team delivers a feature-complete executable before system testing begins.
3. Directory and file permissions allow reading input and writing output.
4. The Windows environment provides standard I/O behavior.

Dependencies

1. **Code delivery** by development must occur before SVT start.
2. **Error message specifications** remain unchanged after test case development.

3. Sample valid and invalid input files must be available no later than test development completion.
4. Test environment (Windows 10 or later) must be operational and stable.

3.0 Test Requirements

Derived from SRS sections 2.2–2.6.

Functional Requirements

1. **Command-line parameter handling**
    - Valid options: -h, -c, -t, -g
    - Combined options allowed (e.g., -ctg)
    - Incorrect options produce error
    - Missing or invalid filename causes error
    - Invalid output-directory causes error
    - Extra trailing params ignored
2. **Input file format**
    - Course Records (with correct columns and par values 3,4,5)
    - First delimiter record (non-blank column 1)
    - Golfer Records (with correct structure)
    - Ending delimiter record
3. **Tournament logic**
    - Courses: 1–5
    - Golfers: 2–12
    - Each golfer plays each course once
    - Score calculation follows table in SRS 2.3.2
4. **Output generation**
    - Tournament Ranking Report sorted by descending total score
    - Golfer Report sorted alphabetically
    - Course Report sorted by descending course score
5. **Error handling**
    - Non-numeric stroke/par input → fatal error
    - Par not in {3,4,5} → fatal error
    - Duplicate golfer record for same course → ignore duplicates with message
    - Output file overwrite prompt
6. **Performance**
    - Program finishes processing within 1 minute.

4.0 Test Tools

- **Windows command prompt**
- **Text editor** (Notepad++, VSCode)
- **File comparison utility** (WinDiff, FC, or diff)
- **Timer utility** (built-in PowerShell measure-command, or external stopwatch)
- **Custom input-file generator scripts** (optional, not required)

No additional paid or specialized tools are needed.

5.0 Resource Requirements

- **1 Test Lead** – responsible for test plan, design, execution, reporting
- **1 Test Engineer** – responsible for test case execution and issue logging
- **Hardware**:
    - One Windows PC (Windows 10 or later)
- **Software**:
    - GolfScore executable
    - Text editing tools

Estimated effort:

Task	Hours
Test planning	10
Test case design	14
Test environment preparation	4
Test execution (all cases)	20
Regression testing	10
Reporting	4
Total	**62 hours**

6.0 Test Schedule

Task	Start	End
Test Plan Development	Day 1	Day 3
Test Case Development	Day 3	Day 6
Test Environment Setup	Day 6	Day 7
System Testing	Day 7	Day 13
Regression Testing	Day 13	Day 15
Final Report	Day 15	Day 16

Schedule assumes 8-hour workdays.

7.0 Risks and Mitigation

Risk	Likelihood	Impact	Mitigation
Ambiguities in SRS	Medium	Medium	Clarify with instructor or interpret conservatively
Output formatting variances	Medium	Low	Use whitespace-tolerant comparison techniques
Late delivery of executable	High	High	Adjust schedule, parallelize test input creation
Misformatted sample files	Medium	Medium	Validate files manually before testing

---

8.0 Metrics

Metrics collected:

Before release:

- Number of defects by severity
- Number of test cases passed/failed/block
- Test execution time
- Defect discovery rate over time (S-curve)

After release:

- Number of defects found in use
- Severity per defect
- Escape rate (post-release defects / total defects)

Appendix A – Detailed Resource Requirements

Breakdown of effort:

Activity	Engineer	Hours
Test Plan Authoring	Lead	10
Test Case Writing	Lead + Engineer	14
Input-file creation	Engineer	6
Execution of Test Cases	Engineer	20
Regression Testing	Engineer	10

Activity	Engineer	Hours
Defect Logging	Engineer	2
Reporting	Lead	4
Total	—	**66 hours**

Appendix C – Test Cases
Each test case includes ID, description, input, expected output, and requirement link.

C.1 Command-Line Tests

ID	Description	Input	Expected Output	Req.
1	Display help	golf -h	Help text displayed; no file required	2.2
2	Missing options	golf input.txt outdir	Error: no options specified	2.2
3	Invalid option	golf -z in.txt out	Error: unrecognized option	2.6.1
4	Combined options	golf -ctg in.txt out	All three reports generated	2.2

C.2 Input File Validation

ID	Description	Input Case	Expected Output	Req.
10	Par value invalid	Par = 6 in course record	Fatal error: invalid par	2.6.2
11	Non-numeric stroke	Letter in stroke field	Fatal error	2.6.2
12	Duplicate golfer/course record	Same golfer twice for course	Second ignored, warning printed	2.6.2
13	Missing delimiter	No first delimiter record	Fatal input format error	2.4

C.3 Scoring Logic
ID	Hole case	Strokes vs. Par	Expected Score	Req
20	Par	4 strokes on par 4	1	2.3.2
21	1 under	3 strokes on par 4	2	2.3.2
22	Over par	6 strokes on par 5	0	2.3.2
23	≥3 under	2 strokes on par 5	6	2.3.2

C.4 Tournament Ranking Report

ID	Description	Input	Expected	Req
30	Ranking tie	Two golfers same score	Sorted alphabetically for tie	2.5.1
31	Multiple courses	3-course input	Total = sum of course scores	2.5.1

C.5 Golfer Report

ID	Description	Input	Expected	Req
40	Alphabetical order	Mixed names	Sorted by last name	2.5.2

C.6 Course Report

ID	Description	Input	Expected	Req
50	Sort by course score	Several golfers	Sorted descending score	2.5.3
51	Stroke listing	One golfer	18 strokes printed	2.5.3

C.7 Output File Handling

ID	Description	Input	Expected	Req
60	Overwrite prompt	Output file exists	Prompt Y/N	2.6.3

ID	Description	Input	Expected	Req
61	User says N	Existing file	No overwrite	2.6.3
62	User says Y	Existing file	Overwrite occurs	2.6.3

C.8 Performance Test

ID	Description	Input	Expected	Req
70	Max-size tournament	5 courses × 12 golfers	< 1 minute runtime	4