Test Plan

1. **Introduction**

**This system level test plan acts as the instruction manual that explains how we will verify the console-based delivery routing program.**

1. **Testing Methodology**

Planning/Analysis – Draft this plan. Review requirements and starter code.

Design – Identify manual scenarios and C++ unit tests.

Implementation – Create Visual Studio test project and write test data sheets.

Execution- Run unit tests, then manual console scenarios. Log any defects to Jira.

Closure – Ensure exit criteria has been met. Add results to scrum report.

**Techniques**

Black-box scenarios: end-to-end console interactions.

White-box scenarios: small functions (e.g. A\*step counter, capacity check)

1. **Objectives**

Correct truck selection – with tie breaking by capacity.

Valid diver paths – avoid buildings and staying on grid.

Capacity enforcement

Robust input validation – weight, size, destination.

1. **Scope**

*Included* – Command-line I/O, data structures, routing logic, A\* path finding on 25x325 grid. Also included is capacity, validation and error messaging.

*Excluded* – GUI/web front-end, performance stress tests, and security.

1. **Schedule & Sequence**

|  |  |  |
| --- | --- | --- |
| **Day** | **STLC Phase** | **Key Tasks** |
| 0 | Planning | Final code pull & compile check |
| 1 | Design/Implementation | Write unit test and manual scenario sheet |
| 2 | Execution | Run unit tests and fix failed tests |
| 3 | Execution | Run manual test cases and log defects |
| 4 | Closure | Retest fixes. Update Jira and scrum report. |

1. **Entry and Exit Criteria**

Entry – Code compiles, test data ready, starter functions integrated.

Exit – All high/medium defected closed or deferred (with documentation). Must pass all critical unit tests. The manual tests should show expected output.

1. **Risks and Mitigation**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Impact** | **Mitigation** |
| A\* algorithm complexity | Wrong paths | Peer review and tiny unit tests |
| Tight timeline | Missed tests | Prioritize objects 1-4 first |

1. **Resources**

Hardware: Windows 10/11 PCs

Software: Visual Studio 2022 Community, GitHub, Jira, Excel for test matrix

1. **Roles and Skills**

|  |  |
| --- | --- |
| **Roles** | **Skills Needed** |
| QA Lead | Jira issue creation, defect severity triage |
| Developer/Tester | C++ basics, VS Debugger, write and assert tests |
| Recorder | Update scrum doc, test matrix, commit results |

1. **Defect Management**

* New issues logged as Jira Bug
* Status flow: In progress – In review – Done.
* Severity: Critical, Major, Minor

1. **Test Deliverables**
2. This test plan in TestPlan/
3. VS Unit-Test project in SourceCode/Tests/
4. Manual Test case sheet in TestDocuments/
5. Jira defect export attached to scrum submission
6. Updated Scrum Report and Traceability Matrix
7. **Assumptions and Dependencies**

* Starter code compiles on VS 2022 without external libs
* Team members have laptop or lab access
* Instructor feed back on data structures is received by Day 1