## **Appendix T – Quality Assurance Plan**

**1.0 Introduction**

This QA Plan outlines how Quality Assurance (QA) will be used across all phases of the *Network Performance Evaluation of Linux Based Operating Systems in a Physical Environment* R&D project. The plan details walkthroughs, QA/QC responsibilities, metrics, tools, and templates to maintain confidence in our results.

This QA plan follows the Software Testing Life Cycle (STLC) methodology, ensuring quality is embedded into each structured phase of the project. Each QA activity aligns with a corresponding STLC stage, supporting systematic planning, test design, execution, and closure for accurate and consistent evaluation results.

**2.0 QA Walkthrough Procedures by Project Phase**

|  |  |
| --- | --- |
| **Phase** | **Walkthrough Procedure** |
| **Requirement Analysis** | QA plans review all project documentation (proposal, WBS, risk register, etc.) to ensure that scope, evaluation metrics (throughput, delay, jitter, loss), and deliverables are clearly defined and feasible. |
| **Test Planning** | QA activities define test strategy, tool selection (D-ITG), responsibilities, and environments, supporting risk mitigation and stakeholder validation. |
| **Test Case Development** | QA creates test scripts and documentation, validated against requirements. Peer reviews align with STLC design verification. |
| **Test Environment Setup** | QA configures consistent hardware and OS routing environments, ensuring test readiness before execution. |
| **Test Execution** | QA plan ensures all test cases across all configurations are checked against failures or anomalies which will trigger re-runs, as outlined in STLC. |
| **Test Closure** | QA reviews, summarizes, and validates test results, ensuring deliverables are aligned with project goals. Supervisor/client feedback is documented. |

**3.0 Quality Assurance Metrics**

|  |  |
| --- | --- |
| **Metric** | **Description** |
| **Evaluation completion rate** | % of all planned evaluations completed |
| **Re-run frequency** | % of evaluations outside 95% confidence interval |
| **Setup bug count** | Number of environment/configuration-related issues encountered per OS |
| **Execution bug count** | Number of issues during or after evaluation runs |

**4.0 Quality Control Activities**

**Unit Evaluation**

* Verify NIC configurations and IP addresses.
* Ensure IP forwarding is enabled and functioning on routers.
* Confirm D-ITG and iPerf tools run properly on each OS.

**Integration Evaluation**

* Evaluate full packet flow across sender → router 1 → router 2 → receiver
* Validate correct routing and subnet communication between networks.

**System Evaluation**

* Execute full evaluation scenarios with 12 packet sizes per OS.
* Evaluate IPv4 and IPv6 separately across TCP and UDP.
* Review logs and verify data consistency across 10 repeated runs.

**Bug Tracking & Re-evaluating**

* Log bugs and produce daily and weekly bug reports (on each evaluation).
* Apply 95% confidence interval rule and re-run failed scenarios.
* Track and document issue resolution.

**5.0 Templates & Checklists (to be developed in Phase 3)**

|  |  |
| --- | --- |
| Item | Purpose |
| Evaluation case template | Define evaluate setup, packet size, OS, and expected output |
| Router configuration checklist | Ensure consistent dual NIC setup, IP forwarding, routing |
| Evaluation log template | Record evaluation attempt details, logs, tool used, results |
| Bug report form | Track defect info, reproduction steps, screenshots |
| Peer review checklist | Verify evaluation cases/scripts/configs before execution |

**6.0 Review & Update Policy**

This QA plan is a living document. It will be reviewed:

* After Test Planning (Phase 2)
* After first full OS evaluation
* After Evaluation Closure (Phase 6)

These three review points were chosen to match key moments in the project: once planning is complete, after running the first full evaluation, and at the very end of the project. They give the team a chance to reflect on what’s working, adjust the QA approach as needed, and improve the process as we go. All updates will be version-controlled and documented.

**7.0 Ensuring Quality**

The following quality attributes are prioritized in this project:

|  |  |
| --- | --- |
| Quality Attribute | Applied in project |
| Understandability | Evaluate plans, logs, and scripts are clearly documented and reviewed by all team members. |
| Correctness | 95% confidence threshold and re-run policy ensure that only accurate data is accepted. |
| Testability | Scenarios are repeatable across 3 OSs × 2 protocols × 2 IP versions × 12 packet sizes. |
| Learnability | Clear direction, training plan, and step-by-step evaluation to improve accessibility for team members. |
| Usability | Evaluation tools (D-ITG) are user-friendly for automated evaluating. |
| Reliability | Same hardware/software setup used per OS to ensure consistent performance evaluation. |
| Portability | Evaluation cases are designed to run identically across Ubuntu, Fedora, and Kali. |
| Efficiency | Scripts automate evaluations and logging, reducing manual effort and execution time. |
| Maintainability | Evaluation scripts and configuration files are version-controlled and peer reviewed. |
| Flexibility | Fallback evaluating tools (iPerf if D-ITG fails) are pre-planned. |