**Quality Assurance Plan**

**Prepared by Larissa & Zafar**

**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.

**2.0 QA Walkthrough Procedures by Project Phase**

1.0 Draft Quality Assurance Plan

1.1 Introduction

1.2 Purpose

1.3 Policy Statement

1.4 Scope

2.0 Management

2.1 Organizational Structure

2.2 Roles and Responsibilities

2.2.1 Technical Monitor/Senior Management

2.2.2 Task Leader

2.2.3 Quality Assurance Team

2.2.3 Technical Staff

3.0 Required Documentation

4.0 Quality Assurance Procedures

4.1 Walkthrough Procedure

4.2 Review Process

4.2.1 Review Procedures

4.3 Audit Process

4.3.1 Audit Procedures

4.4 Evaluation Process

4.5 Process Improvement

5.0 Problem Reporting Procedures

5.1 Noncompliance Reporting Procedures

6.0 Quality Assurance Metrics

Appendix

Quality Assurance Check List Forms

|  |  |
| --- | --- |
| **Phase** | **Walkthrough Procedure** |
| **Requirement Analysis** | The team reviews 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** | The team confirms tool compatibility (D-ITG), defines evaluation scenarios and environments, and ensures upskilling align with planning needs. Feedback from the client and supervisor is used to validate planning outcomes. |
| **Test Case Development** | Test cases are reviewed against project requirements. Test scripts are peer reviewed, and Test cases are validated with the client. Templates for logs, bug reports, and scenario documentation are also prepared during this phase. |
| **Test Environment Setup** | The team sets up the test hardware and software, verifies router configurations and network topology, and ensures a consistent environment for each operating system. Setup is reviewed before any execution begins. |
| **Test Execution** | Each scenario (OS, 12x packet sizes, 2x IP versions, 2x protocols) is executed 10 times. Results are logged and reviewed daily. Tests outside the 95% confidence range are re-run. Issues and bugs are tracked, and regular peer review ensures consistency. |
| **Test Closure** | Results and analysis are verified and compared across all operating systems. Reports, graphics, and the final poster are created. Closure documentation is reviewed and submitted with feedback from the client and supervisor. |

**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 tool runs 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 Project Feasibility**

This Quality Assurance Plan support project feasibility by ensuring all technical activities align with the project’s goals and given constraints. Compatibility checks, structured upskilling, validated configurations, and evaluation results checks are used to upkeep the project’s feasibility throughout its lifestyle.

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

|  |  |
| --- | --- |
| Item | Purpose |
| Router configuration checklist | Ensure consistent dual NIC setup, IP forwarding, routing |
| Test log template | Record evaluation attempt details, logs, tool used, results |
| Peer review checklist | Verify evaluation cases/scripts/configs before execution |

**7.0 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 Test 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.

**8.0 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. |