Performance Testing Phase

Automated Network Request Management in ServiceNow

1. Introduction to Performance Testing Phase

The **Performance Testing Phase** is a crucial stage in the lifecycle of the Automated Network Request Management system in ServiceNow. This phase ensures that the application performs reliably, securely, and efficiently under expected and peak usage conditions. It validates that the system can handle real-world workloads while maintaining data accuracy, access control, and service quality.

For an enterprise ITSM solution like ServiceNow, performance testing is not limited to speed alone. It also evaluates **data handling efficiency, role-based access control, quality assurance (QA) testing, and data integrity**. This phase confirms that the solution is production-ready and compliant with organizational standards.

2. Objectives of Performance Testing

The main objectives of this phase are: - Validate system performance under normal and peak load conditions - Ensure efficient data handling for large volumes of requests and tasks - Verify role-based access control and security enforcement - Identify performance bottlenecks and system limitations - Ensure data integrity across workflows and transactions - Validate overall quality through structured QA testing

Successful completion of this phase ensures system stability, scalability, and user trust.

3. Data Handling Performance

3.1 Overview of Data Handling

Data handling performance refers to how efficiently the system processes, stores, retrieves, and updates data related to network requests, tasks, approvals, and audit logs. In Automated Network Request Management, the system must handle a high volume of records without impacting user experience.

3.2 Data Volume Scenarios

The following scenarios are considered during testing: - High number of concurrent network request submissions - Multiple tasks generated from a single request - Large approval histories and audit logs - Frequent updates to task states and work notes

3.3 Performance Validation Areas

- Record creation time for Network Requests
- Task generation latency
- Approval processing time
- Search and filter performance in lists
- Report and dashboard data loading time

3.4 Optimization Strategies

- Use indexed fields for frequently searched columns
- Avoid unnecessary synchronous business rules
- Optimize Flow Designer actions
- Use reference fields instead of string values
- Archive old records when applicable

4. Access Control Testing

4.1 Purpose of Access Control

Access control ensures that users can only view and modify data that is relevant to their role. Since network requests often contain sensitive information, strict access enforcement is essential.

4.2 Role-Based Access Control (RBAC)

Roles tested include: - Network Request User - Network Approver - Network Engineer - ServiceNow Administrator

Each role is tested to verify: - Correct form visibility - Field-level access restrictions - List and record access permissions

4.3 Access Control Rules (ACL) Testing

- Read, Write, Create, and Delete permissions validated
- Sensitive fields restricted to authorized roles
- Approval actions limited to approvers only
- Backend access tested using impersonation

4.4 Security Validation Scenarios

- Unauthorized access attempts
- Cross-role data visibility checks
- Approval bypass prevention
- Secure handling of firewall and VPN request data

5. Quality Assurance (QA) Testing

5.1 Overview of QA Testing

QA testing ensures that the Automated Network Request Management system meets functional, non-functional, and business requirements. It verifies correctness, usability, reliability, and performance.

5.2 Types of QA Testing Performed

5.2.1 Functional Testing

- Validate request submission process
- Verify approval workflows
- Confirm task creation and assignment
- Test notification triggers

5.2.2 Performance Testing

- Load testing with multiple concurrent users
- Stress testing under peak conditions
- Response time measurement

5.2.3 Usability Testing

- Ease of request submission
- Clarity of form labels and instructions
- Navigation flow testing

5.2.4 Regression Testing

- Ensure new changes do not affect existing functionality

5.3 Test Case Management

- Test cases documented with expected results
- Test execution recorded for traceability
- Defects logged and tracked until closure

6. Data Integrity Testing

6.1 Importance of Data Integrity

Data integrity ensures that information remains accurate, consistent, and reliable throughout the system lifecycle. This is critical for auditability, compliance, and reporting.

6.2 Integrity Validation Scenarios

- Mandatory fields validation
- Prevention of duplicate network requests
- Parent-child relationship consistency
- Approval and task synchronization

6.3 Transaction Integrity

- Atomic execution of flows
- Rollback behavior on failure
- No partial record updates

6.4 Audit and Logging Validation

- Approval history recorded correctly
- Work notes and comments retained
- Change tracking enabled

7. Performance Metrics and Evaluation

7.1 Key Performance Indicators (KPIs)

- Average request creation time
- Task assignment time
- Approval turnaround time

- SLA compliance rate
- Error and failure rate

7.2 Acceptance Criteria

- Response time within acceptable limits
- Zero critical security defects
- No data inconsistency issues
- All high-priority test cases passed

8. Risk Identification and Mitigation

Risk	Impact	Mitigation Strategy
Performance degradation	High	Query and flow optimization
Unauthorized access	High	Strong ACL enforcement
Data inconsistency	Medium	Business rules and validations
High defect rate	Medium	Multiple QA cycles

9. Test Environment and Tools

- ServiceNow Developer Instance
- Test users with different roles
- Impersonation for role validation
- Built-in ServiceNow performance tools

10. Documentation and Reporting

- Test summary reports
- Performance benchmark reports
- Defect logs and resolution status
- Final test sign-off document

11. Compliance and Best Practices

- Alignment with ITIL standards
- Adherence to ServiceNow best practices
- Configuration over customization

12. Conclusion

The Performance Testing Phase validates that the Automated Network Request Management system in ServiceNow is stable, secure, and efficient. By thoroughly testing data handling, access control, QA processes, and data integrity, this phase ensures the solution is ready for production deployment.

Successful performance testing minimizes operational risks, enhances user confidence, and guarantees long-term system reliability and scalability.