Proposer’s Name:  Enter Proposer Name Here  
Date:  ______________

Instructions:
1. A response is required for each requirement contained within this document.
2. If additional space is needed Proposer should clearly label their response with the requirement identifier.
3. NYS reserves the right to allow NYSDOT and/or the Proposer to correct obvious errors of omission.
4. Within each of the responses, identify which requirements from Attachment 1 Functional Requirements will be met.

For each of the following business challenges, provide detailed responses as to how the Proposer’s system meets the requirement.

<table>
<thead>
<tr>
<th>Rqmt. No.</th>
<th>Requirement Description – System Testing and Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA1</td>
<td>The Proposer must describe their system testing and acceptance strategy proposal which shall include, but not be limited to, validation of system functions against requirements, performance of system functions and security features, how the system interoperates with all interfaces in terms of accuracy and performance. Upon certification of all system testing done by the Proposer, NYS requires a minimum of 6 weeks for user acceptance testing, system integration testing and production readiness testing. The Consultant shall warranty all work for 30 days post system acceptance. The below scenarios represent a minimum sampling of acceptance testing scenarios expected by NYS.</td>
</tr>
</tbody>
</table>

- The Proposer shall prepare an overall test strategy document which defines the levels of testing to be performed (e.g. unit testing, system integration testing). This document should include the appropriate metrics (e.g. no show stopper defects, entry and exit criteria to different testing environments) as well as any tools used to assist with testing. The test strategy document must be approved by NYS prior to acceptance testing.
- The Proposer shall prepare and document test plans which will include expected results and validation techniques for performance, functionality, interoperability, backup and restore procedures and high availability design requirements.
- The Proposer shall participate in the execution of acceptance testing under the guidance of NYS.
- The Proposer shall provide a full and complete audit trail for all acceptance testing and the requisite reporting from this audit trail.
- The Proposer shall perform system backup, restore and recovery testing prior to implementation under the guidance of NYS.
• working with NYS to schedule the upgrades when necessitated by 3rd party upgrades (e.g.; O/S upgrades)

Proposer’s TA1 Response:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OITS’s QA/QC team at NYSDOT is responsible for facilitating, coordinating and managing Integration, Functional and Load/Performance Testing for DOT systems. Integrators will work with this team to create test scripts and analyze the test results. Integrators are fully responsible for remediating failed test results until successful. RT DOT uses the following tools for:</td>
</tr>
<tr>
<td></td>
<td>1. Testing Repository is supported using Application Life Cycle Management (HP ALM);</td>
</tr>
<tr>
<td></td>
<td>2. Integration and Functional testing is supported using Unified Functional Testing (UFT), also known as Quality Center;</td>
</tr>
<tr>
<td></td>
<td>3. Load/Performance testing is supported using HP LoadRunner;</td>
</tr>
<tr>
<td></td>
<td>4. Application Code Scanning is supported using Fortify, or an approved EISO tool;</td>
</tr>
<tr>
<td></td>
<td>and,</td>
</tr>
<tr>
<td></td>
<td>5. Vulnerability Scanning is supported using Qualys, or an approved EISO tool.</td>
</tr>
</tbody>
</table>

Any system developed for use by or on behalf of NYS must undergo vulnerability scanning as described in NYS-S15-002 ([https://its.ny.gov/document/vulnerability-scanning-standard](https://its.ny.gov/document/vulnerability-scanning-standard)). This standard addresses pre-deployment scans, implementation scans, and frequency of scans (determined by information classification and connectivity/accessibility). The standard also provides remediation timeframe expectations based on risk and severity of identified vulnerability.

In addition, all application code that resides in the Development Tier must undergo application scanning prior to being migrated to the Test or QA tier at any time. Vulnerabilities discovered during the application scan must be remediated in accordance with NYS-S13-002 ([https://its.ny.gov/document/secure-coding-standard](https://its.ny.gov/document/secure-coding-standard)). The standard requires mitigating controls for medium and high risk threats identified during a system’s life cycle. Un-remediated vulnerabilities must have a formal remediation plan and documented approval of the SE executive responsible for risk management.

Scanning tools must be approved by the EISO and be able to generate a summary of findings, provide remediation suggestions and be able to associate a severity value to each vulnerability discovered.

Scanning results are considered confidential and sensitive and should be protected in a manner appropriate with that classification.

**1.1.1 Development Tier**

The Development Tier is the environment where integrators will develop, modify, and test software code. The development environment is used to perform unit testing. Unit testing is the most ‘micro’ scale of testing and validates a particular function or specific section of code. These tests are written and performed by developers, require detailed knowledge of the...
internal program design, and cover things like statements, branches, conditions and paths. Unit tests should also include the preliminary testing of interfaces (exchanges of data or control) with other systems. Integrators will have the appropriate accounts and permissions to perform the aforementioned tasks in this tier.

Security Testing - Software Code will be scanned in this tier supported using the Fortify tool. Integrators will be responsible for analyzing and remediating all Medium and High Risk Vulnerabilities until resolved, and prior to promoting code to the Test tier.

1.1.2 Test Tier

The Test Tier is the environment used for Functionality, User Interface, Security, and System Interface testing. Functionality testing is a collaborative effort by developers, testers, and end users, which requires no knowledge of the internal design. It is software testing that evaluates the system's overall compliance with its specified functional requirements and behavior versus the expectations of the customer. Integrators will be responsible for analyzing and correcting all failed test results until all test results are successful, and prior to promoting code to the QA tier. Intra-system testing includes:

- Functionality testing - Verification that the individual components flow and function effectively as one integrated system.

- User Interface testing - Verification that the Graphical User Interface (GUI) meets its written specifications, including look and feel ('user-friendliness'), handicap compliance, drop downs, lists of values, validity checking, etc.

- Security testing - Verification that the system protects data while enabling functionality as intended. The six basic elements that should be addressed, regardless of the method used to implement security are: confidentiality, integrity, authentication, authorization, availability, and non-repudiation.

- Static testing – The Intra-system test phase is where activities such as desk checks, code reviews, and peer reviews intended to find and fix mistakes should be conducted. Code reviews allow the transfer of knowledge and best practices from the more experienced staff to the less experienced.

- System Interface testing - System Interface testing seeks to validate the quality of the interfaces between independent systems. This includes all exchanges of data or control and file extracts and/or reports into the system being tested, as well as to all other internal and external systems. System Interface testing may also be performed utilizing the Quality Assurance Tier.

1.1.3 Quality Assurance Tier

The Quality Assurance (QA) tier is the environment used for system interface, integration and functional testing comprising Performance, Recovery, and Operational testing. Integration and Functional testing validates that all necessary interfaced services and components will function as designed in the production tier. Integrators will be responsible for analyzing and correcting all failed test results until all test results are successful, and prior to promoting code to the Prod tier. This testing includes:
Performance testing - a set of tests (stress, load, etc.) used to determine the stability of the system under a variety of operational conditions, from normal to extreme, often to a breaking point. Performance tests seek to reveal the system’s robustness, availability, and error handling under heavy loads to ensure that the software doesn't crash due to insufficient resources (memory, disk space, number of connections, network capacity, etc.), high concurrency, denial of service attacks and the like. Performance tests are also used to determine the speed or effectiveness of a computer, network, software program, or device.

Recovery testing - the activity of testing how well an application is able to recover from crashes, hardware failures, and other similar problems.

Operational testing - verifies that all components that were insufficiently tested, or were not tested at all, during prior phases are functioning as intended, including: VDI, GIS, LDAP, OID, mobile devices, Iws (Tivoli), OBIEE, etc.

User Acceptance testing - User acceptance is the point in time when the customer, using as a basis the collective results of prior testing and exposure to the system, acknowledges that the system meets the predefined criteria for that phase and may move to the next phase of an implementation.

Regression testing - Regression testing focuses on finding defects (previously working functionality that has stopped working) after changes to the code or the operating environment have taken place. The extent of testing will depend on the magnitude of the changes and/or the level of risk to NYSDOT should the software fail. NYSDOT strives to fashion easily repeatable, preferably automated, tests to help streamline regression testing efforts. A sanity test, or sanity check, is a basic type of regression test used to quickly assure that the application or system works as expected.

Security testing – Vulnerability scanning must be performed on the application prior to migration to the Production tier. In the Quality Assurance Tier an In-depth Web Application Scan should be performed on web applications.

1.1.4 Production Tier

The Production Tier is the operational environment operational application utilization. Per NYS-S15-002 (https://its.ny.gov/sites/default/files/documents/nys-s15-002_vulnerability_scanning_1.pdf), Vulnerability Scans are run periodically in accordance with the applicable application risk rating as shown in Table 3: Frequency of Scans (NOTE: Tables 3 and 4 are located inside this website). A Vulnerability Scan will be run supported by the Qualys tool. Integrators will be responsible for analyzing and remediating all Medium and High risk Vulnerabilities until resolved and in accordance with Table 4 in the aforementioned standard.

The Proposer shall state its ability to comply with and abide by all of Attachment 5’s requirements.

TA2 – Proposer’s Response:
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TA3</td>
<td>The Proposer must describe their strategies for working with the RT DOT QA/QC Team to create test scripts and analyze test results, in accordance with Requirement #TA2.</td>
</tr>
</tbody>
</table>

Proposer’s TA3 Response:

|-----------|--------------------------------------------------------------------|
| TA4       | The Proposer must describe their requirements for NYS resource participation in the acceptance testing activities identified for each their plans. Included must be:  
- Key NYS resources roles and responsibilities  
- List of milestones and artifacts requiring NYS approval with a timeframe for review and feedback |

Proposer’s TA4 Response: