Maine State Government
Dept. of Administrative & Financial Services
Office of Information Technology (OIT)

Application Deployment Certification Policy

I. **Statement**

Any computer application must undergo a battery of tests to determine if it is suitable to be deployed into production. Based on the test results, the Chief Information Officer (CIO) makes the final determination whether or not this application should be placed into production.

II. **Purpose**

As applications have become more complex, more interconnected, and more exposed to the external world, it has become even more important to thoroughly vet them before they are deployed into production. This policy establishes a uniform and objective battery of tests that enables the CIO to evaluate the suitability of an application to be deployed into production.

III. **Applicability**

This policy applies to all applications under the purview of the CIO. This includes both new applications as well as modifications to existing applications, irrespective of hosting location (applies to both OIT-Hosted and Remote-Hosted).

IV. **Responsibilities**

A. Application Directors: Enforce this Policy.

B. *Application Owners*: The *Application Owners* are responsible for executing this test battery. This certification consists of:

1. The names and signatures of the Project Manager, the Product Manager, and the Executive Sponsor; and
2. A summary result (Passed/Failed/Not Applicable) and a short paragraph clarifying that summary result, for each of the tests specified below.

Any part of the testing required by this policy may be outsourced to a third-party without affecting the responsibility or the prerogative of the *Application Owners*. Irrespective of who actually executes a test, the *Application Owners* still remain in charge of its execution. The *Application Owners* are not answerable to the third-party regarding the nature or the result of any outsourced test. Further, the third-party exclusively conveys the test results directly back to the *Application Owners*.

For OIT –Hosted applications, State personnel will generally perform any applicable application tests.

For Remote-Hosted applications, it is a generally a combination of vetting vendor-provided test results and State personnel performing applicable tests. Provided that vendor-provided results for a specific application test are deemed acceptable by the Applications Director and Subject Matter Experts (Enterprise Security Officer for Security, etc.), no further State personnel testing is required for that item. Should there be deficiencies, then additional testing must be conducted by either the vendor or by State personnel, until acceptable results are achieved.

C. Associate CIO for Applications: Owns and interprets this Policy.

D. Chief Information Officer (CIO): The CIO may delegate authority to certify or approve applications for deployment. Regardless of approving authority, certification of applications will be based on advice from the Director, PMO, the Chief Technology Officer (CTO), the Associate CIO for Applications, and/or other subject matter experts.

E. Enterprise Security Officer (ESO): Interprets Security Test results, reviews vulnerability remediation plans, and determines pass/fail results for Security Test. The ESO may accept vendor-provided test results in lieu of OIT testing.

V. **Directives**

A. The following list defines the battery of application tests:

1. *Use Cases* Test: Ensures proper functioning of all the features of the application.
2. Accessibility Test: Ensures compliance with the Maine I.T. accessibility policies and standards.
3. Data Conversion Test: Ensures the accurate migration of appropriate legacy data.
4. Interfaces Test: Ensures proper functioning with all companion applications.
5. Security Test: Ensures the confidentiality, integrity, and availability of the application.
6. Performance Test: Ensures responsiveness under projected average and peak processing loads.
7. Restoration Test: Ensures full functioning of the application following an infrastructure rollback/restoration.
8. Regression Test: Applies exclusively to modifications of existing applications. Ensures that a new version does not compromise existing functionality.
9. Operating Platform Test: Ensures proper functioning of the application across all combinations of relevant hardware and software components.

B. Brief general descriptions of the tests are provided below:

1. *Use Cases* Test: An application must have a complete, stable, and up-to-date documentation of the full set of its *Use Cases*. Each of the *Use Cases* must be executed individually, and verified that it indeed delivers as expected. Beyond individual *Use Cases*, *Application Owners* must also know which *Use Cases* are

likely to be invoked simultaneously with one another. All such likely combinations of *Use Case* interactions must be tested. Finally, it is also important to test a representative sample of actual end-users performing their daily jobs holistically, using the entirety of an application. At the completion of the *Use Cases* Test, the end-users must be satisfied that the application does indeed meet all their expectations, or alternatively, that they are willing to accept any deficiencies thereof. At the discretion of the Associate CIO for Applications, alternative requirements definition artifacts may be acceptable in lieu of *Use Cases*.

2. Accessibility Test: The application must be tested to ensure its compliance with the State I.T. accessibility policies and standards[1]. The OIT Accessibility Team provides guidance and the final determination regarding testing tools, etc.

3. Data Conversion Test: It is likely that record structures and formats of the legacy application were modified as a result of the migration into the new application. It must be ensured via actual testing that all business-critical data have indeed survived the migration. It is left to the discretion of the *Application Owners* to determine exactly what constitutes 'business-critical data.' But once so determined, it must be ensured that such data are indeed accessible from the new application. Should the new application cause modifications to the existing workflows, then this step must also include testing the new workflows.

4. Interfaces Test: An application must have a complete and up-to-date documentation of all the data and workflow dependencies between itself and all other applications that it interacts with. All such interactions must be tested. Interfaces must anticipate errors, and therefore, incorporate robust error-handling and error-logging capabilities. While it is desirable to exclusively utilize the Test environments of the various applications when testing the interfaces, it may be necessary under certain circumstances to pair the Test environment of this application with other environments of companion applications, as long as such other applications participate in the interface on a read-only basis.

5. Security Test: The application must ensure the highest levels of Confidentiality (No unauthorized access), Integrity (No tampering), and Availability (No denial-of-service). All personal, medical and financial data, in motion, must be encrypted end-to-end, both inside and outside the State firewall. All personal, medical, and financial data must be encrypted at rest in the Demilitarized Zone. Data hosted on servers inside the firewall are not subject to encryption, but data resident in portable computing devices must be encrypted at all times. A full vulnerability assessment and penetration test must be performed on the application. Applications should not only guard against standard security vulnerabilities (such as Weak Credentials, Injection Attacks, Buffer Overflows, Cross-site Scripting, etc.), but they should also be designed to thwart denial-of-service attacks. Beyond such generic requirements, an application may also need to satisfy additional

[1] http://www.maine.gov/oit/accessibility/policy/index.html

specific, statutory requirements, as set forth by CJIS, HIPAA, FISMA/FIPS, SOX, GLBA, CROMERR, USA Patriot Act, etc. By default, High/Critical vulnerabilities must be remediated prior to go-live, and Medium/Moderate vulnerabilities must have a remediation plan and an approved waiver prior to go live. In all cases, regardless of classification level (high, medium, low) the Enterprise Security Officer has the final word regarding which vulnerabilities require remediation prior to go-live and which require a remediation plan/approved waiver prior to go-live.

6. Performance Test: Performance testing determines the responsiveness of the application to its users, and therefore, its acceptance and adoption. The application must respond adequately under the projected average load, as well as expected peak load. The application must not cause unreasonable adverse impact on either network throughput or server loading. In order to safeguard against adverse user perception, the application must establish a two-tiered response time specification, one for data inquiry/lookup, and another for data modification transactions, assuming Ethernet or broadband connectivity end-to-end. For performance testing, the application may consider using automated tools that simulate user behavior, including simultaneous and staggered loading. Beyond response times, other aberrations that must be investigated include non-linear performance, i.e., response time increasing disproportionately with loading, and response time varying during periods of constant load. This is a test that requires close cooperation with the *Software as a Service (SaaS)* provider and considers joint tenancy.

7. Restoration Test: Subsequent to a point-in-time recovery of the entire suite of application components (the client-device, the webserver, the application server, the file server, and the database server), the application must be tested to ensure that it functions exactly as expected. The restoration test should encompass all components represented on the application's architecture diagram. This test demonstrates that in the event of a catastrophic failure, all system components ("all boxes") are recoverable. Thus, unless an application is contained wholly within the database layer, all application tiers should be included in the test. Any dependencies on enterprise components outside the application should be considered for inclusion in this test, for example secure file transfer, schedulers, reporting tools, but may be considered out of scope if their restoration has been previously certified. It is left to the discretion of the *Application Owners* to determine whether the entire suite of *Use Cases* or merely a core suite of essential *Use Cases* will be executed in the restoration test. Either way, the endgame is to ensure that the application functions entirely to the satisfaction of its end-users following an infrastructure rollback/restoration. Equally important is to negotiate with the infrastructure provider and the *Application Owners* the two metrics of recovery: *Recovery Point Objective* and *Recovery Time Objective*. This is a test that requires close cooperation with the *Software as a Service (SaaS)* provider.

8. Regression Test: This test applies whenever there is a modification to an existing application, either an upgrade to the application proper, or an upgrade to an

embedded, third-party component. This is to ensure that the modification did not adversely affect previously working functionality. A two-pronged regression test strategy must be undertaken. One prong is based upon the release notes and the known module dependencies. A focused test suite must be administered for those *Use Cases* that are actually affected as part of this upgrade. At the same time, the other prong, a core suite of essential functions must also be tested, irrespective of whether or not they underwent any modification as part of this upgrade. It is left to the discretion of the *Application Owners* to determine exactly what constitutes a 'core suite of essential functions.' Such a two-pronged strategy provides a safety net against inadequate release notes and incomplete knowledge of module dependencies.

9. Operating Platform Test: The application must be actually loaded, or configured, on all combinations of hardware, operating systems, network configurations, terminal emulators, browsers, etc., that are planned for production deployment, and verified that it works as expected, end-to-end, and across the board. This includes all relevant client devices, network configurations, as well as the full complement of web, application and database servers. It will not suffice to accept the product vendors' compliance statements in lieu of actual testing. It is often the case that the pre-production environment is different from the production environment. The extent of such variance could be subtle, e.g., the pre-production environment could be Oracle on Windows, whereas, the production environment could be Oracle on AIX. Nonetheless, the application must be actually tested under Oracle on AIX prior to its deployment. And the same holds for alternative network configurations, terminal emulators, browsers, etc. The consensus determination of the Hosting and Application Director provides the final word as to what extent this can be fulfilled in practice.

VI. **Definitions**

1. Application Owners: With respect to the application being considered for deployment, the Project Manager, the Product Manager, and the Executive Sponsor are jointly and collectively identified as the *Application Owners*. If, for any reason, any of the roles turns out to be vacant, or the same person fulfils more than one role, or there arises a difference-in-opinion with respect to this Policy among the three roles, for the purpose of this Policy, the decision of the Associate CIO for Applications, will be final and binding.

2. Recovery Point Objective: The *Recovery Point Objective* is the point-in-time *to which* an application must be restored subsequent to a disaster or disruption.

3. Recovery Time Objective: The *Recovery Time Objective* is the duration-of-time *within which* an application must be restored subsequent to a disaster or disruption.

4. Software as a Service (SaaS): End-user application consumed from either the Public Cloud or OIT-Hosted infrastructure.

5. Use Case: A *Use Case* is a well-defined sequence of actions undertaken jointly by the

user and the application, which produces a predictable result of value to the user. Thus, a *Use Case* captures a discrete functionality of an application completely independent of the underlying implementation. Beyond the expected outcomes, a *Use Case* must anticipate errors, and therefore, incorporate robust error-handling and error-logging capabilities. The full set of all the *Use Cases* for an application constitutes the complete value added by that application.

VII. **References**

VIII. **Document Information**

Initial Issue Date: September 22, 2010
Last Revision Date: August 13, 2015

Point of Contact: Henry Quintal, Architecture-Policy Administrator, OIT, 207-624-8836.
Approved By: James R. Smith, Chief Information Officer, OIT, 207-624-7568.
Enforced By: Paul Sandlin, Associate CIO for Applications, OIT, 207-624-7568.

Legal Citation: Title 5, Chapter 163: Office of Information Technology[2]
Waiver Process: See the Waiver Policy[3].

[2] http://legislature.maine.gov/statutes/5/title5ch163sec0.html

[3] http://maine.gov/oit/policies/waiver.htm