OSEHRA

Open Source Electronic Health Record Agent

Software Quality Certification Plan – v1.0

Date Released:

16 September 2011

Prepared by:

Technical Lead:
Rick Avila
OSEHRA
900 N Glebe Rd
Arlington, VA
munsk@vt.edu
202-320-4613

Document Version Control

Revision	Editor	Date	Changes
0.90	Gustavo A. Benitez Jr	4 August 2011	Original Draft Document
0.91	Wesley Turner	17 August 2011	Revised Draft Document
0.92	Rick Avila	18 August 2011	Revised Draft Document
0.93	Wesley Turner	15 September 2011	Review to Create Final Document
1.0	Gustavo A. Benitez Jr	16 September 2011	Final edits and formatting

Table of Contents

Section **Page**

Exhibits

Section **Page**

1 Software Quality Certification Plan

This document contains the Software Quality Certification Plan for the Open Source Electronic Healthcare Record Agent (OSEHRA) project. It is intended to provide guidelines for the development of the Software Quality Certification system to ensure that the OSEHRA can attest that the VistA code is suitable for distribution and use for the development of VistA installations.

1.1 Executive Summary

Certification is initially the attestation and ultimately the verification that all executable artifacts managed by the OSEHRA meet the following criteria.

- **Safe**: Individual code units do not cause errors in other components of the system and the code is robust to all code paths and conditions.
- **Compliant:** Code meets agreed upon interface specifications and is adherent to all applicable laws and regulations.
- **Functional:** Code has a defined set of requirements that are met when the code executes.

Veterans Affairs (VA) has developed an extensive set of processes and procedures that are currently used to certify VistA and related software for internal use. VA maintains information resources on its secure intranet that facilitate testing and certification, and track the progress of the certification process. Much of this certification infrastructure and information is not currently shared with potential members of an open source development community external to VA.

It is not the goal of the OSEHRA to replace current VA testing and certification processes and procedures, nor is the OSEHRA inherently a testing body. Rather, the OSEHRA will establish a framework of tools, procedures, and educational materials to enable and actively engage a vibrant open source community in evolving a robust, community-based certification process.

2 Purpose

The central role of the OSEHRA in the ecosystem is to be steward of the VistA codebase. In this section we introduce how the purpose of the certification process aligns with the purpose of the OSEHRA and its principles.

2.1 OSEHRA's Purpose

"The OSEHRA's mission is to facilitate the development and maintenance of a VistA-based electronic health record information systems technology that is freely and spontaneously available for all medical beneficiaries through the promotion and use of state-of-the-art open source best practices."

2.2 OSEHRA's Open Source Principles

The OSEHRA has adopted the following governing principles as an Open Source entity:

- **Principle #1:** Innovation comes from the outside. It must be channeled inside.
- **Principle #2:** Software is knowledge transformed into code. It needs an engaged community to maintain it and operate it.
- **Principle #3:** Software is never a finished product, it evolves continuously, and that evolution requires an involved community.
- **Principle #4:** Attract interested people, with shared goals, earn their trust.
- **Principle #5:** Transparency; remove any obstacles to free flow of information.
- **Principle #6:** Meritocratic governance driven by: Autonomy, Mastery, and Purpose
- **Principle #7:** Release Early, Release Often
- **Principle #8:** Avoid Private Discussions
- **Principle #9:** Establish Credibility; build relationships with Open Source communities.
- **Principle #10:** Welcome the unexpected. Listen carefully to the community.

These principles will guide the decisions that will be made when considering different alternatives for the design and implementation of a software quality certification process.

2.3 The Purpose of Software Quality Certification

The purpose of the software quality certification process is to raise trust and confidence in the code that is to be introduced in the codebase. To address the final goal of channeling innovation into the codebase, the software quality certification process should create the mechanisms by which developers are encouraged to contribute bug fixes, improvements, and new functionalities to the codebase, while at the same time ensuring that those contributions are safe, compliant and functional before they are made available as part of a VistA distribution. The purpose of software quality certification is aligned with the purpose of the OSEHRA in the process of facilitating the development and maintenance of the VistA codebase, and in following best practices of open source communities.

3 Software Quality Certification Procedure

The software quality certification procedure defines the tools and mechanisms that comprise the code certification process for three types of code contributions:

1. Official Freedom of Information Act (FOIA) deliveries and associated patch streams.
2. Code contributions to be considered for integration into the VistA codebase.
3. Code contributions intended to be interoperable with VistA, but not intended for integration into the VistA codebase.

The plan is defined in six sections with Section 3.1 containing a definition of the term software quality certification for the purposes of the effort; Section 3.2 containing a description of the current VistA certification process; Section 3.3 containing a description of the final vision for software quality certification; Section 3.4 identifying gaps in the current process; Section 3.5 containing a description of the proposed implementation; and Section 3.6 containing a summary.

3.1 Definition

Software quality certification is the attestation that the following criteria are true. The code is:

- Safe
- Compliant
- Functional

Each of these terms is defined individually below.

3.1.1 Safe

For the purposes of the software quality certification process, code is safe when the individual code units do not cause errors in other components of the system and the code is robust to all code paths and conditions. We will certify code as safe using a combination of

- Regression Testing
  - Tests that each routine/function correctly performs operations
- Stress Testing
  - Tests the robustness of the system under all code pathways (exception handling)

3.1.2 Compliant

For the purposes of the software quality certification process, code is compliant when it meets the agreed upon code contract and is adherent to all applicable external laws and regulations. We will certify code as compliant using a combination of:

- Architectural Verification
  - Ensures that the code adheres to a defined community standard.
  - Verifies module-to-module dependencies.
  - Verifies module-to-data dependencies.
- Interoperability Testing

- Tests that all module application programming interfaces (APIs) are defined and met.
- Section 508 Testing
  - Tests that the user interface is compliant with Section 508 of the Rehabilitation Act.
- Governance Compliance Testing
  - Adherence to community Standards and Conventions (SAC).
  - Adherence to Licenses
- Documentation
  - User level documentation is provided.
  - Developer level documentation is provided.
- Privacy Protections and Systems Security
  - Tests that privacy and security requirements are defined and met.

3.1.3 Functional

For the purposes of the software quality certification process, code is functional when it has a defined set of requirements; when its execution satisfies the defined requirements; and when execution occurs within an acceptable performance window. We will certify code as functional using a combination of:

- Confirmation that requirements have been defined and met
  - Map functionality to test criteria
- Performance Testing
  - Verifies that the system meets its key performance (speed, availability) specifications

While not part of the software quality certification process, customer satisfaction is also an important metric of an open source community. Customer satisfaction will be monitored as part of the Community Engagement function and will be presented using the software quality certification dashboard.

3.2 Description of the "As Is" Environment

The current "As Is" environment is a thorough and detailed process involving multiple teams and following a set of stringent procedures that evaluate the code and its performance. These processes are laborious and intensive, and we believe they are dominated by attestation and manual validation.

The OSEHRA continues to evaluate the existing environment and is actively seeking procedures and documentation from within the VA and the external community that can be incorporated into the OSEHRA activities. Among the tools and processes we have currently identified are:

- The NOIS issue management system
- A ClearCase code repository
- The Forum system for patch management and certification
- A well-defined set of procedures for evaluating externally obtained (Class 3) code, and a detailed "***remediation list***" for managing required modifications

- The use of Beta testing sites at clinical facilities that have experience in testing the software
- A Section 508 Conformance Validation Statement and Section 508 checklists for Software Applications and Operating Systems; for Web-based Intranet and Internet Information and Applications; for Performance Criteria; and for Information, documentation, and support
- The ***MUnit*** testing facility
- The ***CATS*** tool for comparing a given VistA instance to a canonical instance
- IBM's ***Rational Testing Suite***
- Existing standards for coding in Delphi, MUMPS, Java, and Oracle.
- Product Requirements Document (PRD)
- Initial Requirements Analysis (IRA)
- Software Requirements Specification (SRS)
- Use Cases (UC)
- Supplemental Specifications (SS)
- Software Design Document (SDD)
- Master Test Plan
- ***LoadRunner*** from HP
- A VA "Hospital Simulator"

We are currently determining the prevalence and extent of the usage for these tools and procedures.

3.3 Description of the "To Be" Environment

The "To Be" environment will allow for increased automatic and manual testing in support of the flow of code contributions from VA to the external community, and from the external community back into VA. The testing and attestation of code changes will work in close conjunction with the code repository procedures defined in the *OSEHRA Code Repository Plan* via interactions with the Gerrit code review tool and the use and description of Gerrit is detailed in Section 4 of that document.

The "To Be" environment will employ a dashboard system where the status of the current VistA codebase will be displayed. A screenshot from the existing prototype dashboard is shown in Exhibit 1. As we discuss Software Quality Certification, keep in mind that the certification and attestations will come in three distinct varieties with distinct emphasis.

Most frequent will be the software quality certification of changes as part of a code review process. These will be Gerrit based reviews and attestations of incremental changes to the code base prior to merging the changes into the code repository. Review and attestation at this level will be continuous, but local. Automated tests will be run against the code base with the changes present, but the reviewer will not be expected to go beyond the context of the locally submitted changes and bug fixes during any manual attestations. The goal here is to keep bad code, code that has a significant performance penalty, code that is untested, and code that does not meet community compliance standards from being merged into the system; but not to require an exhaustive and time consuming review.

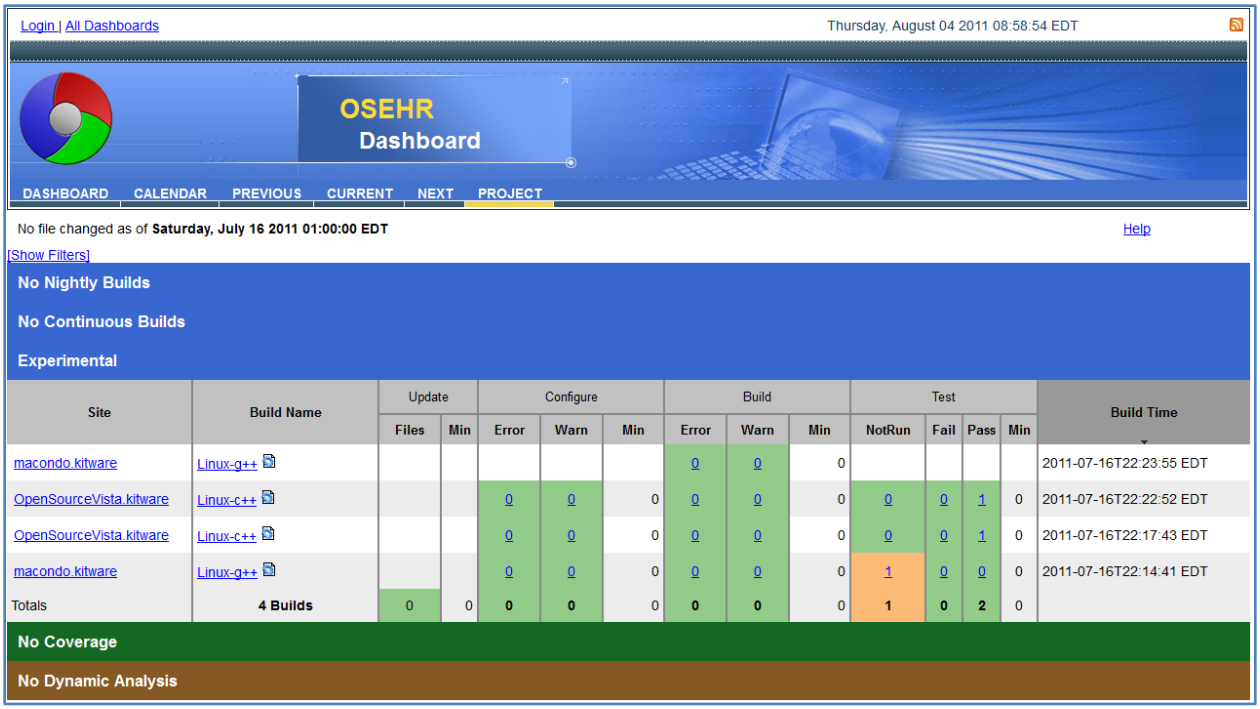

Exhibit 1 - Prototype OSEHRA Dashboard.

At the next level, will be reviews and attestations of larger code changes or the proposed contribution of new functionality or modules to the code base. These will be based on OSEHRA Technical Journal review and will require additional documentation and review consistent with the current VA requirements for evaluating third party (Class 3) code.

Finally, there will be periodic formal releases of the OSEHRA code base. Each new release will be initiated by a Gerrit push containing a single modification to increase the system version. Review for formal releases will parallel the normal Gerrit code review process, but will not be local in scope and will be expected to more thoroughly evaluate the testing and attestations over the complete codebase paying particular attention to the complete corpus of changes since the previously accepted formal release. Following a successful release review, a signed Git tag will be created using an OSEHRA PGP key to provide an additional level of confidence to users downloading the release.

Gerrit reviews for incremental and formal releases will be reported to the OSEHR Software Quality Dashboard and indexed by hash code.

The "To Be" environment will also manage an open issues dashboard based on JIRA, which is analogous to the internal NOIS system of VA. Users and developers will both have access to the system to report, confirm, and resolve issues in any of the compliance procedures.

The OSEHRA software quality certification process is anticipated to evolve organically according to the best practices of open source communities. Initially the OSEHRA will be focused on the quality control processes that are closer to the field developers, which will help raise adherence of their code to the goals of safe, compliant, and functional; while simultaneously fostering their programming skills by providing constant feedback through the

community-wide platforms such as Git, Gerrit, and the quality control Dashboards. The certification effort will explore the spaces where the OSEHRA can provide value to VA software teams as well as to the external VistA communities, and will avoid overlapping with processes for which satisfactory solutions are already available in the ecosystem. Following Open Source best practices, the OSEHRA will work with the many actors of the ecosystem in a collaborative manner to identify the openings where focused efforts can deliver value to both VA and the external community.

3.3.1 Safe

Module safety status will be available from the dashboard.

3.3.1.1 Regression Testing

To ensure safety, a large part of the automated testing will consist of a set of unit and regression tests that capture the current state of the system. Individuals designing new code will be able to download the system test environment including test fixtures, test data, and expected results, and will be able to verify their installation reflects the "Gold VistA" system at the OSEHRA by executing the tests and submitting "Experimental Results" from their system back to the dashboard. As individuals continue to develop code, they will be expected to develop tests that exercise and validate their contributions. These tests become part of the existing test suite on their individual installations and will be executed by the test suite and reported to the Experimental dashboard. This capability allows the system to capture any interference between the newly developed and existing codebase and also ensures that newly developed code adds to the suite of tests available.

3.3.1.2 Stress Testing

The OSEHRA will provide educational materials, examples, and tools to aid developers in generating automated and manual tests that verify the operation of the system under all code pathways and exception conditions.

The OSEHRA will work with VA and the community to build a publicly available dataset that is a plausible representation of a VistA instance data in a realistic clinical facility. This type of dataset will be extremely useful to the open community to exercise stress testing on their code before they submit it to the certification process. This dataset will, of course, not contain any private information and will be reviewed carefully to ensure that it satisfies current regulations.

3.3.1.3 Peer Review System

Peer review is an important piece of the certification process and ties together the code repository with software quality certification. We identify three different mechanisms for peer review depending on the magnitude and goal of the requested code/schema change. These mechanisms provide expedited review when the code/schema change is small and localized, typically to address minimal changes or to fix a bug in the system. More thorough review is required to add more substantial capability such as a new module or package, or to attest to the software quality of a formal release. Each of these is discussed below.

For small, evolutionary code/schema changes or bug fixes, the OSEHRA will implement a code review system such as that in use for the NLM-sponsored Insight Segmentation and Registration Toolkit (ITK). This system will be based on Gerrit, a code review system developed by Google for the Android project and integrated with the Git environment. Peer review will require that one or more trusted individuals assert that the code meets publically available criteria developed by the community. At a minimum, this will include asserting that the code is safe (i.e., passes the regression and unit tests), meets the community standards, and is functional.

For larger code/schema changes, a review system based on the OSEHRA Technical Journal will be required. The Journal entry will include the code/schema change to be evaluated and a set of materials such as documentation, functional definition, test fixture, test data and test results, along with assertions that the code complies with a community developed Software Developers Kit (SDK). A prototype OSEHRA Technical Journal web page is shown in Exhibit 2.

Finally, formal releases will use a release process in conjunction with a system Gerrit review.

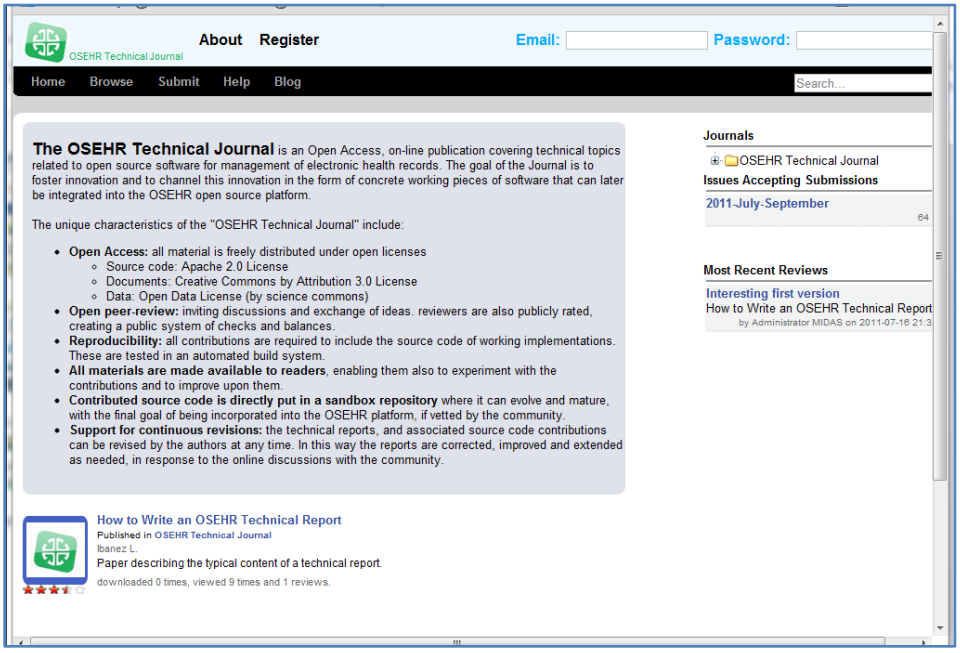

Exhibit 2 - Prototype OSEHRA Technical Journal

3.3.2 Compliant

Module compliance will be displayed on the dashboard and will establish that the code is compliant with community standards, architectural requirements, and applicable laws. An SDK will be published to the open source community and will be the basis for the following automated architectural verification:

3.3.2.1 Architectural Verification

Architectural verification ensures that the code adheres to defined Community SAC, and that the actual code is consistent with the architecture defined module-to-module and module-to-data dependencies. To the extent possible, this testing will be automated.

3.3.2.2 Interoperability Testing

Interoperability Testing verifies that all module APIs defined within the coded modules are consistent with the architecture. To the extent possible, this testing will be automated.

3.3.2.3 Section 508 Testing

Section 508 Testing verifies that that the User Interface is compliant with section 508 of the Rehabilitation Act. The OSEHRA will provide educational materials, examples, and tools to aid developers in meeting these requirements.

For example, the VA provides a set of documents and checklists for validating section 508 compliance including:

- A Section 508 Conformance Validation Statement
- A Section 508 checklist for Software Applications and Operating Systems
- A Section 508 checklist for Web-based Intranet and Internet Information and Applications
- A Section 508 checklist Performance Criteria
- A Section 508 checklist for Information, documentation, and support

These documents and checklists will be integrated into the Gerrit Code Review and OSEHRA Technical Journal and made part of the attestation process required of new code and code changes.

3.3.2.4 Governance Compliance Testing

Governance Compliance Testing is based on the SDK, Community SAC, and Licenses. To the extent possible, this testing will be automated.

3.3.2.5 Privacy Protections and Systems Security

Privacy Protections and Systems Security testing verifies that privacy and security requirements are defined and met. To the extent possible, this testing will be automated.

3.3.3 Functional

Functional testing results will be shown on the dashboard.

3.3.3.1 Requirements defined and met

Every module will have requirements defined using the Health Level 7 (HL7) EHR system functional model (EHR-S FM) and will have automated test fixtures, test data, and test results defined. The test fixture will be used by the dashboard support systems to execute the tests and capture and display them on the OSEHRA dashboard. Code changes will need to update and define changes to any module documentation. New modules will be required to submit a set of descriptive documents to the OSEHRA Journal as defined in the published SDK.

3.3.3.2 Performance Testing

A performance specification will be developed for each module and semi-automated tests will capture module performance status and provide it to the OSEHRA dashboard for display.

3.4 Identified Gaps

The purpose of a gap analysis is to identify key issues that must be addressed in order to move the open source development community toward the "To Be" environment in a controlled and expeditious manner. We note here that VA has an extensive and rigorous test process that is applied during development and prior to the official release of any VistA code. This gap analysis is concerned only with qualifying those areas where the current procedures need to be supplemented to support the open source community and to enable VistA development over a wider group of developers.

NOIS is an identified system for issue tracking within VA that is not available outside of VA. It will be necessary to replicate this capability in the external community.

3.4.1 Safe

Automated unit and regression testing coupled to test driven programming have been shown to be essential components of successful open source development processes. A primary goal of the OSEHRA community enablement program will be to educate the community, provide appropriate incentives, and make tools and techniques available to facilitate a smooth transition from current practices. The entire community benefits from a codebase that is certified to be comprised of interoperable components and minimal coding errors. As the codebase evolves with the addition of new components developed by all members of the open source community, the cost and effort required to maintain today's testing and certification approach will be prohibitive and a major impediment to innovation.

Unit tests are a part of VistA code development and testing within VA and the "As Is" environment contains a significant number of tools and procedures. However, the tests that exist are internal to VA and are not part of the current release to the broader VistA community. Currently, publically available test fixtures are insufficient for supporting effective automated unit, regression, and stress testing. As the OSEHRA procedures and acceptance criteria are institutionalized, developers will be required to submit appropriate documentation, architectural artifacts, test fixtures, and test results along with codes.

The migration path begins by increasing the flow of information from VA into the external community so that all developers understand currently defined coding standards, interface agreements, and module dependencies. Through community enablement and interactions, a SDK will be developed and continuously evolved in order to provide the community with guidelines and tools to facilitate the transition to automated testing and increased reliance on standards based interfaces. The OSEHRA dashboard will provide a framework to allow the entire community to monitor the progress of this transition. The OSEHRA will work closely with VistA developers to identify the tools and resources that can be shared with the community, as well as tools that may need to be developed.

3.4.2 Compliant

VA has a code process that is used internally and that feeds current VistA development. This process is not available to the broader open source community and was not designed to meet the needs of a team outside VA. The existing VistA System Architecture is currently being

documented for the "As-Is" VistA product, at which point we will evaluate and adjust the procedures in accordance with community needs and based on community feedback and involvement. These will be published as a SDK to the community. New code contributions including bug fixes and new modules will be required to submit SDK compliant code packages to include the appropriate architectural artifacts, test fixtures, and test results along with any code submission contributions.

3.4.3 Functional

Current functional testing is, by necessity, manual and localized to VA. The VistA System Architecture is maintained internally, and while it appears to be a rigorous and well defined community, visibility outside of VA is limited. To address this, we are mapping the system architecture to the HL7 EHR-S FM. Once complete, the EHR-S conformance criteria will be provided to developers and testers to help them define and test future modules. To the extent that it is feasible, automated testing tools will be introduced into the process in order to test compliance with functional requirements. Such tools will largely rely on script driven test cases defined in conjunction with the functional requirements specifications.

The key to certification of system functionality is clearly defined and broadly communicated functional specifications. By facilitating the creation and community-wide communication of a product road map and requirements specifications, the open source community will be able to develop functional testing processes and procedures, as well as test fixtures.

3.5 Implementation

Following best practices of Open Source communities, the OSEHRA will take a staged approach to the establishment of the certification procedures, starting by putting in place the necessary support framework at the time the OSEHRA becomes operational and continuing through to a full implementation in the "To Be" environment. Note that the certification procedures establish the tools and framework for the certification process, but the OSEHRA cannot generate all of the tests that are required for full certification by itself. The OSEHRA will rely heavily on an active and enthusiastic open source community to create the full test suite and to correct the codebase. Many of the OSEHRA activities will be focused on creating the conditions for this active community to be productively engaged.

3.5.1 Operational Goals

The OSEHRA became operational on 19 August 2011. The initial FOIA delivery was received and brought into a Git-based version control environment. Software quality certification of this initial version will focus on proving that the version controlled codebase can be used to construct a working VistA instance. Accordingly, the OSEHRA software quality certification process will:

- Extract the VistA code base and globals from the repository
- Stand up (create) a VistA installation based on the extract
- Execute a set of manual tests to ensure that the system is operational and functional

The OSEHRA operational date precedes the software quality certification Environment operational date of 18 October 2011. Third party contributions will not be accepted until the software quality certification environment becomes operational.

3.5.1.1 Safe

To satisfy the safety requirements, the OSEHRA will:

- Attest safety based on the results of the manual tests undertaken on the VistA environment after extraction from the Git-based repository

3.5.1.2 Compliant

To satisfy the compliant requirements, the OSEHRA will:

- Attest compliance based on the results of the manual tests undertaken on the VistA environment after extraction from the Git-based repository

3.5.1.3 Functional

To satisfy the functional requirements, the OSEHRA will:

- Attest functionality based on the results of the manual tests undertaken on the VistA environment after extraction from the Git-based repository

3.5.2 Initial Surge Goals

On 18 October 2011 the OSEHRA Software Quality Certification Environment becomes operational. On this day, the software quality certification Environment will exist in an initial state. This state will comprise implementations of a software quality certification tool set, and tool and procedural documentation for the certification process.

Certification Tools

Certification tools will be provided to perform regression and unit tests, compliance attestation, and functional attestation allowing the OSEHRA to meet the needs of Safe, Compliant, and Functional. The tools will consist of:

- An automatic test driver (CTest)
- A code review tool (Gerrit)
- A technical journal for proposing and introducing new functionality (OSEHRA Technical Journal)
- A mechanism for accepting issue reports and action requests (JIRA)
- A reporting tool (CDash dashboard)

An initial set of module level tests based on the MUMPS/VistA cross reference generator (XINDEX) will be implemented as both an example of module level testing and as a substantive indicator of the health of the VistA delivery. Other test status will be determined by attestation based on expert opinion or by established procedures and checklists. For example, 508 compliance certifications will be supported on the Gerrit code review tool and the OSEHRA Technical Journal using procedures and checklists derived from VA documents such as:

- Section 508 Conformance Validation Statement
- Section 508 Checklist for Software Applications and Operating Systems
- Section 508 Checklist for Web-based Intranet and Internet Information and Applications
- Section 508 Checklist for Performance Criteria
- Section 508 Checklist for Information, documentation, and support

The VistA environment does not contain a substantial corpus of automated tests that will be immediately available to the open-source community. As such, the automatic testing environment on 18 October 2011 outside of the XINDEX check will be limited.

Gerrit and the OSEHRA Technical Journal will be the primary mechanisms for accepting new code into the OSEHRA repository. Of these, Gerrit is best suited to small code changes and will be used for issue resolution and incremental functionality. The OSEHRA Technical Journal will be used to introduce significant new functionality. When the OSEHRA Software Quality Certification Environment becomes operational:

- Code review for the peer review site and the technical journal will rely heavily on manual attestation by knowledgeable VistA community members;
- Both review mechanisms will provide a method for a trusted reviewer to attest that the submission is safe, compliant, and functional.

JIRA, an issue tracking system that is free for use for open source projects, will be established as a mechanism for reporting issues and requesting new functionality.

The dashboard will provide a community facing report on the status of the VistA codebase to include the results of automatic and manual unit and regression tests; and manual compliance and functional testing. Results of the automated and manual module testing, and the current state of compliance and functional attestation derived from Gerrit and the OSEHRA Technical Journal will be captured and display on the dashboard.

Educational Materials

Maturation of the software quality certification process within the open-source community requires that the community provide regression and unit tests, publish documents describing the use of the system, and establish community norms. As part of the Software Quality Certification Environment status on 18 October 2011, the OSEHRA will provide an example automated test based on XINDEX; working versions of documents defining the use of certification tools; and proposals for the community standards. These documents will include:

- A safety testing document describing the use of the testing system to generate, execute and submit unit and regression tests for safety compliance
- A SDK describing the preparation of architectural artifacts and attestation of architectural compliance
- A functional testing document describing the use of the testing system to generate EHR-S FM conformance criteria and attestation of functional compliance

Each of these documents will be considered a working document until the community has had the opportunity to review and comment on them. The OSEHRA will also begin soliciting additional material from the open-source community in support of Safe, Compliant, and Functional:

- Regression and stress tests (Safe)
- Architectural artifacts from the user community (Compliant)
- Functional change proposals (Functional)
- Performance test requirements and performance testing toolsets (Functional)

3.5.2.1 Safe

To satisfy the safety requirements, the OSEHRA will:

- Establish an OSEHRA dashboard ready for submissions
- Prepare educational materials on how to design stress and regression tests, and on how to use the testing environment to submit dashboards
- Establish the Gerrit peer review system
- Establish the OSEHRA Technical Journal
- Begin soliciting regression and stress tests from the user community

3.5.2.2 Compliant

To satisfy the compliant requirements, the OSEHRA will:

- Establish an OSEHRA dashboard ready for submissions
- Prepare a SDK on how to prepare appropriate architectural artifacts and on how to use the testing environment to submit dashboards
- Establish the Gerrit peer review system
- Establish the OSEHRA Technical Journal
- Begin soliciting architectural artifacts from the user community

3.5.2.3 Functional

To satisfy the functional requirements, the OSEHRA will:

- Establish an OSEHRA dashboard ready for submissions
- Prepare educational materials on how to use and test to the EHR-S FM conformance criteria and on how to use the testing environment to submit dashboards
- Establish the Gerrit peer review system
- Establish the OSEHRA Technical Journal
- Begin soliciting functional change proposals from the user community

Gerrit and the OSEHRA Technical Journal will be the primary mechanisms for establishing requirements and for certifying that developed code is safe, compliant, and functional. When the OSEHRA becomes operational, we will have Gerrit and the OSEHRA Technical Journal sites available, but will not have a complete and community approved SDK. The review capability will rely heavily on manual review by knowledgeable VistA community members and may

require an iterative submission process to fully define the set of requirements and code constructs that need to be part of each contribution.

Both review mechanisms will provide a method for a trusted reviewer to attest that the submission is safe, compliant, and functional.

At the OSEHRA Certification Environment operational time, performance testing will be supported in the automated test framework and dashboard, but the actual performance tests and test environment will require further investigation to determine the appropriate toolsets; to generate the specific tests; and to specify the performance goals.

3.5.3 Early Maturation Goals

At the early maturation stage, the OSEHRA envisions a more complete system with active automatic testing and mature, manual procedures.

The JIRA issue tracker should be relatively complete and in active use to support code quality.

3.5.3.1 Safe

To satisfy the safety requirements, the OSEHRA will:

- Require regression test contributions as part of new code and code modification procedures
- Require stress test contributions as part of new code and code modification procedures
- Fully support an experimental section in the OSEHRA dashboard to support and capture independent testing by developers
- Require the use of the Gerrit peer review system for code modifications
- Require new code submissions to be submitted to and reviewed within the OSEHRA Technical Journal

3.5.3.2 Compliant

To satisfy the compliant requirements, the OSEHRA will:

- Use the OSEHRA dashboard to attest and monitor code compliance status
- Use the SDK on how to prepare appropriate architectural artifacts as the basis for automating the testing environment to submit dashboards
- Institutionalize the Gerrit peer review system
- Use the OSEHRA Technical Journal
- Manage architectural artifacts from the user community

3.5.3.3 Functional

To satisfy the functional requirements, the OSEHRA will:

- Use the OSEHRA dashboard to attest and monitor code functional status
- Require the use of educational materials on how to use and test to the EHR-S FM conformance criteria and on how to use the testing environment to submit dashboards
- Use the Gerrit peer review system

- Use the OSEHRA Technical Journal
- Manage functional change proposals from the user community

As the certification authority reaches maturation, the SDK defining requirements for bug fixes will be refined to meet the defined needs of the community and will be made concrete in the establishment and use of the Gerrit peer review system and the OSEHRA Technical Journal.

At early maturation, the OSEHRA will have a mechanism in place for performance testing and will report the results of performance tests to the dashboard. This will include test timings in terms of absolute time and in terms of deviations from the norm. For more critical modules, the SDK will require developers to specify automated and manual test suites that exercise and verify the performance specifications. These test suites will become part of the complete test suite of the system and will be exercised along with other compliance tests.

To the extent that the OSEHRA can identify appropriate tools for profiling VistA code in an open source environment, these tools will be made available to the community and will be augmented by use documents and examples. In particular, the OSEHRA will work with VA and the open source community to build a realistic dataset representing an instance of a clinical site that is suitable for public distribution. The availability of such a dataset will facilitate testing by field developers early in their development process.

3.6 Summary/Conclusion

A software quality certification framework will be established to verify that submitted code is safe, compliant, and functional. First, these processes and products will be brought up to modern software engineering best practices and will used by developers to attest to the codebase's status. As we go forward, the software quality certification procedures will be institutionalized and automated within the OSEHRA community.