



OSEHRA

Open Source Electronic Health Record Agent

OSHERA Final Review Checklist – v0.5 (DRAFT)

Date Submitted:

18 January 2013

Prepared for:

Department Of Veterans Affairs

Prepared by: Larry G. Carlson

Document Version Control

Table of Contents

| Section | Page |
|----------------|------|
| Overview | 1 |
| Checklist..... | 3 |
| Notes | 10 |

Overview

This document describes the set of checks that a Final Reviewer should perform as part of the process of approving a code contribution posted by a Primary Developer and reviewed by at least one Peer Reviewer, with the aim of including it in the OSEHRA code base.

- **Primary Developer** is the developer who is addressing an issue or a feature request that has been reported in the OSEHRA JIRA issue tracker.
- **Peer Reviewer** is another developer who is also knowledgeable on the VistA/MUMPS code base and its quality control system.
- **The Final Reviewer** is mostly focused on the Software Quality Assurance oversight of the patch.

This current OSEHRA document is an adaptation of the “*SQA Review Checklist*” used by the Product Development team at the Department of Veterans Affairs. Changes are very minor and are mostly intended to adapt to the software tools and procedures that have been setup for OSEHRA.

Final Reviewers are expected to perform the verifications listed in this checklist, as they study code submission, and to fill-up the entries in the checklist accordingly. The processed checklist must be included in a read-only format (for example a PDF document) as part of the review submission. Once processed, this document must be uploaded in the JIRA ticket that originated the action of the Primary Developer. Completed forms should be saved with the “Save As” option and uploaded to JIRA.

Checklist

This document describes the set of specific checks that Final Reviewer must perform as part of the process of reviewing a patch that has been submitted for integration into the OSEHRA code base.

It is the Final Reviewer's responsibility to complete the checklist and assure that it is uploaded to the JIRA ticket that initiated the action of the Primary Developer. This is a requirement for approving a patch in the Gerrit code review system.

When you mark Not Applicable (NA), then a justification must be added in the "Comments" column.

| | |
|--|---|
| JIRA Ticket Number: N/A. This is a new version of a package. | Test Version: 22.2T1 |
| Primary Developer: George Timson | Secondary Developers: Samuel Habel, Fredrick D. S. Marshall |
| Final Review Completed by: Larry G. Carlson | Date: 1/14/2013 |

| Item | Product Build Checklist Question | Review Outcome | | |
|------|---|----------------|----|-----|
| # | Peer Review consists of items 1 -14. Formal Review consists of all items. | Yes | No | N/A |
| | Compliant | | | |
| 1 | Do all components follow the System Design Document? <i>No System Design Document for upgraded File Manager. Most of the changes were incremental upgrades over the last 10 years.</i> | | | X |
| 2 | Do all components satisfy the requests of the Requirements Specification Document? <i>No Requirement Specification Document</i> | | | X |
| 3 | Are the Use Case Specifications documented? <i>No Use Case Specifications documents</i> | | | X |
| 4 | Is the Interface Control Document complete and current? <i>No such document needed</i> | | | X |
| 5 | Are the components required for the build identified? | X | | |
| 6 | Do all components follow Product Development Standards? | X | | |
| 7 | Is Product Component Testing (aka Unit Testing) complete for each component of the build? <i>Unit Test routines are being built, but not complete for all aspects of Filemanager</i> | | X | |
| 8 | Are the results of the Product Component Testing documented? | X | | |
| 9 | Did Product Component Testing follow Product Development Standards? <i>I think so. Certainly have been checking that all meet VA programming standards.</i> | X | | |
| 10 | Have the Test Scripts been completed? <i>Unit Tests being written. However, many more to develop. A lot of manual tests have been completed and accompany this submission.</i> | X | | |
| 11 | Do the Test Scripts conform to Product Development Standards? | X | | |
| 12 | Is the Master Test Plan completed according to the Test Preparation process? | X | | |
| 13 | Is Product Documentation available for the build? <i>Filemanager 22.2 Release Notes; Install Guide; Getting Started Manual. Rest of the documentation is being prepared.</i> | X | | |

| Item | Product Build Checklist Question | Review Outcome | | |
|------|---|----------------|----|-----|
| # | Peer Review consists of items 1 -14. Formal Review consists of all items. | Yes | No | N/A |
| | Compliant | | | |
| 14 | Has the sequence of integration of the Product Components been identified (see Test Preparation Process)? And Documented? | | | X |
| 15 | Has Component Integration testing been performed? Running Fileman, CPRS, VistA options after install in random stress testing for Fileman. | | | X |
| 16 | Has the Component Integration Test Defect Log been completed? No defects found. | | | X |
| 17 | Has the Component Integration Test Evaluation Summary been completed? | | | X |
| 18 | Has the Component Integration Test Execution Log been completed? | | | X |
| 19 | Has the Software Quality Assurance Review Checklist been started? See below. | X | | |

| Final Review Checklist | Complete? Pass /Fail | Comments |
|----------------------------------|-------------------------|----------|
| Before Patch Installation | | |

| Final Review Checklist | Complete? Pass /Fail | Comments |
|--|---|--|
| <ol style="list-style-type: none"> 1. Save the patch description (Checkboxes) <ol style="list-style-type: none"> 1. Perform editorial review 2. Ensure patch subject is clear 3. Ensure acronyms are defined in the first occurrence 4. Run spelling and grammar check 5. Ensure all required Remedy JIRA ticket(s) are listed 6. Ensure patch priority is listed 7. Ensure patch category(s) are listed 8. Ensure instructions for disabling options/protocols are included 9. Confirm there is a statement regarding whether or not users can be on the system 10. Ensure time required to install patch is included 11. Question manual deletion of routines 12. Validate accuracy of patch name(s) found in the description 13. Ensure format is correct for files: FILE NAME (#number) 14. Ensure format is correct for fields: FIELD NAME (#number) 15. Ensure format is correct for options: Menu Text [INTERNAL OPTION NAME] 16. Ensure patch number list is in order of release 17. Ensure full name is used for other packages such as Fileman, TaskMan, MailMan, etc. 18. Ensure dependencies on other packages/patches listed 19. Ensure database changes have been approved by database administrator 20. If this is an ENHANCEMENT type patch, compare the description to the New Service Request (NSR), if applicable 21. Ensure estimate of disk space and journal file consumption is included, if applicable 22. Ensure reference is made to any documentation that will be sent out separately, if applicable 23. Compare the Installation Guide to the patch description's installation instructions, if applicable. 24. Ensure Patient Safety Issues (PSIs) addressed in this patch are identified to include the PSI number. | <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>N/A</p> <p>N/A</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>N/A</p> <p>PASS</p> <p>PASS</p> <p>PASS</p> <p>N/A</p> <p>PASS</p> <p>PASS</p> <p>N/A</p> <p>N/A</p> | <p>Release Notes</p> <p>Completely replaces File Manager</p> <p>Completely replaces File Manager</p> <p>Complete overwrite of File Manager</p> <p>File Manager is now a stand-alone package</p> <p>Not a patch, no NSR</p> <p>Install Guide refernces Release Notes.</p> <p>N/A – Fileman has no separate patch description.</p> <p>No Patient Safety Issues addressed</p> |

| Final Review Checklist | Complete? Pass /Fail | Comments |
|--|-------------------------|--|
| 2. Ensure test plan/problem duplication is available, if applicable | PASS | |
| 3. List Remedy Ticket(s) associated with the patch, if applicable. <ol style="list-style-type: none"> 1. Ensure status field is “Work in Progress” and the Pending Field is empty 2. Ensure developer has posted a note for patch reference 3. [Is this too VA-specific?] Ensure TeamPlay info is in Keywords section (TeamPlay ID.WBS) | FAIL | Remedy ticket number: 447336 (Do not reindex) included in this package. Will be included in the next version of the release notes. |
| 4. Understand the problem or new features of the patch | PASS | |
| 5. Perform ^INDEX | PASS | See exemption descriptions in XINDEX ERRORS spreadsheet |
| 6. Collect the before patch installation checksum(s), if applicable | N/A | Complete update of File Manager |
| 7. Ensure that modifications of routines and files have been reviewed by the Blood Bank Team for potential effects on Blood Bank software in accordance with Directive 2007-038, if applicable [Too VA specific ?] | PASS | Clearance: EFFECT ON BLOOD BANK FUNCTIONAL REQUIREMENTS: VA FILEMAN contains changes to a package referenced in VHA OI SEPG SOP 192-023 Review of VISTA Patches for Effects on VISTA Blood Bank Software. This patch does not alter or modify any VistA Blood Bank software design safeguards or safety critical elements functions. RISK ANALYSIS: Changes made by |
| 8. If Health Level Seven (HL7) segments are included, check that new segment approval has been obtained from the Message Administration | N/A | No HL7 segments included |
| 9. Retrieve the MailMan message/host file, if applicable [Too VA specific ?] | N/A | No Mailman message/host file |
| 10. Review file/global changes and save a “before” capture, if applicable <ol style="list-style-type: none"> 1. Check for use of privileged access (R, W, P/D) 2. Check for current nodes | N/A | Not applicable with DIFROM |

| Final Review Checklist | Complete? Pass /Fail | Comments |
|---|---------------------------------|---|
| 11. Save a copy of the “before” components (other than routines), if applicable <ol style="list-style-type: none"> 1. Data Dictionary 2. Data Values 3. Protocols 4. Options | N/A | Not useful in the case of Fileman as the old version can be retrieved from any database running Fileman 22.0. |
| 12. Duplicate problem and save the results, if applicable | N/A | New File Manager version mostly introduces new features, not address current bugs. |
| Installation | | |
| 1. Load the patch | N/A | N/A. New File Manager comes as DIFROM |
| 2. Verify checksums in Transport Global | N/A | N/A. New File Manager comes as DIFROM. DINTEG routine supplied to perform checksums. |
| 3. Print Transport Global <ol style="list-style-type: none"> 1. Confirm Type – Single Package 2. Confirm National Tracking = Yes 3. Confirm National Package = package of patch 4. Confirm presence of appropriate Required Builds 5. Confirm all expected routines are included in the build (listed in the routine multiple) 6. Confirm exported routine list is correctly namespaced 7. Confirm alpha/beta testin is blank or set to “no” | N/A | N/A. New File Manager comes as DIFROM. |
| 4. Compare Transport Global to Current System | N/A | N/A. New File Manager comes as DIFROM |
| 5. Back a Transport Global | N/A | N/A. New File Manager comes as DIFROM |
| 6. Install Package(s) | PASS | Followed Installation Guide |
| 7. Compare time required to install with installation instructions | PASS | Time shown in examples in Install Guide. Install time depends on Mumps Implementation and CPU |
| 8. Environment check, if applicable | N/A | No Environment Check for package. |
| Post-Installation | | |
| 1. Verify the “after” patch installation checksums, if applicable | N/A | Not a patch. Use ^DINTEG routines. |

| Final Review Checklist | Complete? Pass /Fail | Comments |
|--|---------------------------------|--|
| 2. Perform ^%RCHECK on all patch routines and save results, if applicable | N/A | Not on GT.M or CACHE instances |
| 3. Check first, second, and third routine lines <ol style="list-style-type: none"> 1. First line 2. Second line 3. Third line | PASS | With exemptions for compiled routines and DIFROM found in : XINDEX_ERRORS.xlsx Also captured: CheckFirstSecondThirdLineOfRoutines_20130114.rtf |
| 4. Spell check routines for anything displayed to the user | FAIL | All spelling errors noted. At this point they will not be fixed in this version. See Fileman22SpellingErrors.pdf |
| 5. Perform ^INDEX <ol style="list-style-type: none"> 1. Ensure there are no variables, new with this patch, that were not explicitly killed. 2. Ensure there are no warnings or errors listed | FAIL | With exemptions for compiled routines and DIFROM found in : XINDEX_ERRORS.xlsx There are several first line routine formal lists that will need specific exemption from the SAC committee. |
| 6. Check for necessary integration agreements | PASS | No changes to APIs or externally called APIs. |
| 7. Ensure exported protocols are properly attached to menus, if applicable | PASS | N/A |
| 8. Ensure exported options are properly attached to menus, if applicable | PASS | N/A |
| 9. Review file changes, if applicable <ol style="list-style-type: none"> 1. Save a copy of “after” components (other than routines), if applicable 2. Confirm no erroneous nodes are exported 3. If patch alters data, verify that new values adhere to data dictionaries | PASS | Reviewed all files# <2 for attributes |
| 10. Save copy of “after” components (other than routines), if applicable | N/A | N/A |

| Final Review Checklist | Complete? Pass /Fail | Comments |
|---|-------------------------|--|
| 11. Do a routine search (^%RFIND) for the following items, if applicable <ol style="list-style-type: none"> 1. //// - yes 2. DIC(0) - yes 3. ^UTILITY - yes 4. ^TMP - yes 5. ^XTMP -yes 6. % - yes 7. \$I - yes 8. U= - yes 9. K^ - yes 10. ^(- yes | PASS | Fileman is exempt from these requirements. |
| 12. Review sets and kills of IO variables | PASS | Fileman is exempt from this requirement. |
| 13. Confirm fields which contain executable code are write protected in the DD with “@” or are defined as VA FileMan data type of “M”, if applicable. | PASS | All Mumps nodes are protected with either an @ or ^ access code. |
| 14. Test the patch to confirm that the patch has corrected the problem, if applicable | N/A | Not a patch |
| 15. Check error log (D ^XTER) | PASS | No errors found. |
| 16. Deliver report of findings to the developer and development manager | PASS | |
| 17. Ensure Patch Tracking Message (JIRA) has been delivered to appropriate staff, if applicable. | N/A | DIFROM not patch |
| 18. Perform user documentation review, if applicable | PASS | |
| 19. Ensure the documentation file names in the patch description match the actual file names, if applicable | PASS | |

Notes

- OSEHRA Patch Module Repository (OPMR): code.osehra.org/files/patches
- OSEHRA Technical Journal: <http://code.osehra.org/journal>
- OSEHRA Git Repository: <http://code.osehra.org/gitweb>
- OSEHRA Gerrit Code Review: <http://review.code.osehra.org>