Department of Veterans Affairs

Product Build Checklist



**January 2018**

Version 1.1

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 10/18/2017 | 1.0 | Product Build Checklist Build-1 | Joe McGovern |
| 01/18/2018 | 1.0 | Product Build Checklist Build-2 | Joe McGovern |

Product Build Checklist

**Application:** OneVA Pharmacy **Version:**

**Project:** Clinical Ancillary Services (CAS) **Patch#:** PSD\*3\*83, PSO\*7\*505, PSS\*1\*219,

XU\_8\_689

Medical Permissions and Dispensing

Upgrades (MPDU)

**Date Review Closed:**

**Review Outcome**

| Item # | Checklist Question  Peer Review consists of items 1 -14. Formal Review consists of all items. | Yes | No | N/A |
| --- | --- | --- | --- | --- |
| 1 | Do all components follow the System Design Document? | X |  |  |
| 2 | Do all components satisfy the requests of the Requirements Specification Document? | X |  |  |
| 3 | Are the Use Case Specifications documented? | X |  |  |
| 4 | Is the Interface Control Document complete and current? |  |  | 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? | X |  |  |
| 8 | Are the results of the Product Component Testing documented? | X |  |  |
| 9 | Did Product Component Testing follow Product Development Standards? | X |  |  |
| 10 | Have the Test Scripts been completed? | 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? | X |  |  |
| 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? | X |  |  |
| 16 | Has the Component Integration Test Defect Log been completed? | 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? | X |  |  |

Review Findings Summary Instructions

A Review Findings Summary is a tool created to document and track anomalies and issues identified during reviews.

The Review Findings Summary contains the following information:

| Item | Definition |
| --- | --- |
| **Review Type** | Peer Review or Formal Review |
| **Artifact** | The category of the artifact under review, such as: Requirements Specification Document, Software Design Document, Prototype, Code, Documentation (Release Notes, User Manual, Technical Manual, Installation Guide, Security Guide), Patch Description (if released through National Patch Module), Master Test Plans, and Test Package. |
| **Author** | The person who created the work product under review. |
| **Project** | The official project name. |
| **Application** | The name of the software application to which this work product pertains. |
| **Version** | The version number of the software application pertinent to this work product. |
| **Patch** | If the software is to be released via the National Patch Module, enter the patch number. |
| **Date Review Started** | The date of the review meeting. |
| **Date Review Closed** | The date all anomalies, issues and action items are closed. |
| **Identifier** | A unique identifier that permits identification and sorting; suggested Project acronym + sequential number (i.e., SUR0001) |
| **Anomaly Category** | CM=Configuration Management, CO=Coding, CS=Coding Standards, DC=Documentation Content, DE=Design, DP=Documentation Presentation, IA=Integration Agreement, PE=Performance, SP=Specification, TR=Traceability, TP=Master Test Plan, TS=Test Script |
| **Anomaly or Issue** | Items identified and described during the review. |
| **Resolution** | The solution for the identified anomaly. |
| **Date Resolved** | The date an issue was resolved and the Review Team agrees it was resolved correctly. |
| **Status** | The various states through which an anomaly passes on the way to resolution and closure. The anomaly states are:   * **Submitted** – when an item is logged and reported for repair. * **Assigned** – when an item is assigned for repair. * **Opened** – when an anomaly is assigned for correction. * **Deleted** – when an item is originally reported as an anomaly, but later deleted because the item is either a duplicate or not an anomaly. * **Resolved** - when an anomaly is corrected and sent for review or verification. * **Re-Opened** – when an anomaly is closed and then reopened for modification. * **Returned** - when an anomaly is reviewed, verified as "incorrect", and returned to author. * **Verified** - when an anomaly is reviewed and verified as "correct". * **Closed** - when an anomaly is successfully reviewed and closed with a resolution and resolution date. * **Deferred** - when an anomaly is designated for correction at a later date. * **Duplicated** – when an item is assessed to be a duplicate of a prior record. * **Escalated** – when an item requires evaluation by management.   ***Note****: The statuses listed above reflect the use of Rational ClearQuest for anomaly tracking. Manual tracking may use a simplified list of statuses.* |
| **Impact** | The classification of anomalies according to their potential damage to the software, systems, patient, personnel or operating systems. They are classified in three levels:   * **High Impact** - an error or absence of functionality that may severely jeopardize patient or personnel safety; adversely impacts all users; represents a significant value or cost; is governed by Congressional mandate; affects either a large database or critical data; requires Food and Drug Administration (FDA) certification/approval; affects Veterans Integrated Service Network (VISN) issues; or negatively impacts the interdependence of applications. * **Medium Impact** - an error or absence of functionality that adversely affects the safety of veteran issues or users of large applications, i.e., Pharmacy, Lab, etc.; represents a high value or cost; sponsored or initiated by the National Program Office; or negatively impacts essential operational or business processing. * **Low Impact** - an error or absence of functionality that may cause operator/user inconvenience and minimally affects operational functionality. |

Review Findings Summary

**Artifact**: **Author:** **Project:**

**Application:** **Version:** **Patch:** **Date Review Begun:** **Date Review Closed:**

| Project acronym-number | Anomaly Category | Anomaly or Issue | Date Resolved | Status | Impact |
| --- | --- | --- | --- | --- | --- |
|  |  | Anomaly or Issue:  Location:  Resolution: |  |  |  |
|  |  | Anomaly or Issue:  Location:  Resolution: |  |  |  |
|  |  | Anomaly or Issue:  Location:  Resolution: |  |  |  |
|  |  | Anomaly or Issue:  Location:  Resolution: |  |  |  |
|  |  | Anomaly or Issue:  Location:  Resolution: |  |  |  |

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| April 2014 | 1.4 | Converted to MS Office 2007-2010 format | Process Management |
| October 2011 | 1.3 | Updated formatting to current ProPath documentation standards | Process Management |
| June 2011 | 1.2 | Changed OED to Product Development, where applicable, and changed test plan to master test plan | Process Management |
| July 2010 | 1.1 | Changed Software to System in item 1 | Process Management |
| June 2009 | 1.0 | Initial Version | Process Management |