Swift & Staley Team/DOE Paducah Site
Classification Office (CO)/Technical Information Office (TIO)
and Operations Security (OPSEC) Release Form

To Be Completed by Person Requesting Release
(Required to be completed prior to submittal to CO/TIO/OPSEC)

Date 05/08/2009 Date Release is Required 05/13/2009

Person Requesting Release Teresa Overby Phone Number (270) 441-5188

Mailing Address 761 Veterans Avenue, Kevil, KY 42053

Organization Paducah Remediation Services, LLC, SPCI

Document Number _____ Number of Pages ~~300~~ 11 pages

Accession Number (DMC only) _____

Document Title/Date PRS-BFM-0078, Software Quality Assurance, 05/26/2008

Author _____ Corporate Author _____

Media (Check all that apply)

☒ Paper ☐ Photo ☐ Diskette ☒ Drawing ☐ Video ☐ CD ☐ Report/Letter ☐ Other _____

Project Subcontract/Task Order _____

Requestor/Purpose of Release public

DO NOT WRITE BELOW THIS LINE – CO/TIO/DC/OPSEC USE ONLY

Classified	☐ Yes	☒ No	**UCNI**	☐ Yes	☒ No
ECI	☐ Yes	☒ No	**OUO**	☐ Yes	☒ No

Type of Release (Check One)

☒ Public Release ☐ Internal Release Only ☐ Limited Release

Reason (Check all that apply)

☐ Circumvention of Statute ☐ Statutory Exemption ☐ Commercial/Proprietary

☐ Personal Privacy ☐ Privileged Information ☐ OPSEC

Other/Comments _____

DC M Brenneman Date MAY 12 09

TIO M Brenneman Date MAY 12 09

Reviewing Official _____ Date _____

OPSEC M Brenneman Date MAY 12 09

SSTF-106 Rev 0
10/4/06

PADUCAH
Remediation Services
A Portage Shaw Joint Venture Company

OWNER: Information Technology	**PRS-BFM-0078**	**REV. NO.**
SUBJECT MATTER AREA: Software QA	**PREPARER: Tom Barron & Jim Reason**	**Page 1 of 11**
DOC TYPE: ☒ **PROCEDURE** ☐ **POLICY**	**APPROVED BY/DATE:**	
PROC TYPE: ☒ **OPERATING PROCEDURE** ☐ **FACILITY SPECIFIC PROCEDURE** **FACILITY:** _____	**R. Jeffrey Hoyles, 4/22/08** **(Signature on file in DCC)**	
TITLE: **Software Quality Assurance**		
USQD ☒ **UCD** ☐ **CAT EX** ☐	**EFFECTIVE DATE: 05/26/2008**	
USQD/UCD No: UCD-PH-SITE-0208; USQD-PH-SITE-0129	**REQUIRED REVIEW DATE: 05/26/2011**	
Mandatory Subcontractor Pro Forma Procedure? ☒	**If an interim Procedure, Expiration Date:**	

REVISION LOG		
Revision Number	Description of Changes	Pages Affected
0	Initial Release. This document replaces blue-sheeted procedures “BJC-IT-6007 Software QA Program” and “BJC-IT-6008 Application Life Cycle Mgmt”	All

TABLE OF CONTENTS

1.0 PURPOSE

This performance document flows down and provides implementation guidance for software quality assurance related to planning, development, acquisition, operation and maintenance. It is the intent of this document to describe those accepted methodologies that result in software implementation consistency, efficiency and reliability for those PRS work activities that the software application is used.

2.0 SCOPE

This document applies to software applications developed, purchased commercially, or acquired from DOE or other government agencies or their subcontractors used in Project work activities by software owner organizations, teaming partners, and/or subcontractors.

This procedure applies to PRS, PRS partners and PRS subcontractors doing work on PRS projects.

NOTE: For PRS safety software applications the requirements of PRS-BFM-0079, *Safety Software Quality Assurance Requirements*, also apply.

3.0 TRAINING

Software Owner — Ensure that all Software Users are adequately trained in the use and operation of software applications by developing a training document or manual, or the use of formal training materials or classes provided by the software vendor or other appropriate third party vendor.

4.0 PROCEDURE

NOTE 1 Software owners have the responsibility to coordinate with the PRS Information Technology (IT) manager concerning software applications they plan to procure, develop internally, or have developed by a third-party.

NOTE 2 Sections 4.2 through 4.5 requirements will not apply to commercial off-the-shelf (COTS) non-safety and general business application software.

4.1 Software Acquisition

Software Owner — **4.1.1** Software acquisitions/procurements shall be coordinated with the PRS Information Technology (IT) Manager.

IT — **4.1.2** As required, review software purchase requisition with software owner to ensure software is applicable for stated requirements.

Software Owner

4.1.3 COTS, contractor developed, DOE, or other government agency software used for PRS Project work activities shall be acquired using the PRS procurement process and requisite approvals obtained prior to submitting purchase requisition to Purchasing.

IT

4.1.4 Add new, acquired software to PRS software listing.

4.2 General Requirements for Non-COTS or DOE Acquired Software Applications

NOTE: The requirements discussed in Sections 4.2 through 4.6 are applied using a graded approach based on the type of software, i.e., DOE contractor provided, custom developed, internal developed utility application, etc., and its intended use.

Software Owner

4.2.1 Planning, development, modification, and maintenance shall be performed in a planned, traceable, and systematic manner using an accepted software life-cycle methodology to include as applicable:

- Requirements
- Design
- Implementation
- Testing
- Verification and validation (V&V)
- Operations and maintenance
- Retirement

4.3 Software Planning:

Software Owner

4.3.1 A plan that addresses software quality assurance (SQA) shall be developed for each covered application.

4.3.2 Using a graded approach, the SQA plan should address the following:

- Description of the nature and purpose of the software or application
- The software product or application to which it applies
- Organization responsible for performing the work and achieving software quality
- Standards, convention, techniques, or methodologies that shall guide the software activity
- Required reviews for testing and V&V
- Methods for error reporting and corrective action

4.4 Software Life Cycle Requirements

Software Owner

4.4.1 Requirement Phase

4.4.1.1 Specify in a requirements document or specification those requirements that address functionality, performance, design, constraints, attributes, and external interfaces.

4.4.1.2 Specify only requirements that can be verified and validated.

4.4.1.3 Ensure requirements are documented and traceable through the life-cycle phases.

Software Owner and Developer

4.4.2 Design Phase

4.4.2.1 Software or application design shall be developed, documented, and reviewed based on the requirements document.

4.4.2.2 Design considerations may include:

- Software or application operating environment
- Mitigation of potential problems or abnormal conditions

4.4.2.3 Describe in the design documentation the following, as applicable:

- Major components of the software design
- Technical descriptions of mathematical models, control flow, data flow, control logic, data structure, etc.
- Allowable or defined ranges for inputs and outputs
- Translation of design basis to code
- Requirements-based test plans that outline acceptance criteria and verification of results
- Computer programs used in design activities or operational control and means that demonstrate these programs produce desired results or required performance over range of operation

Software Developer

4.4.3 Implementation Phase

4.4.3.1 Translate design into source code and executable commands to perform intended functions per design specifications.

4.4.3.2 Establish any necessary configuration management (CM) controls.

4.4.3.3 Develop user information documentation on how to use the software or application.

4.4.3.4 Ensure CM items are under appropriate configuration change control measures prior to testing & V&V.

4.4.4 Testing Phase

Software Owner **4.4.4.1** Test and V&V activities are planned, performed and documented.

Software Validator **4.4.4.2** Testing and V&V activities are performed by person(s) who were not involved in the software design and development activities, but are technically qualified to perform and evaluate the adequacy of software to perform its intended function.

4.4.4.3 Testing methods should demonstrate that the software application:

- Properly handles abnormal events and conditions as well as failures
- Does not perform unintended functions
- Does not unexpectedly degrade the system either by itself or in combination with other functions or configured items
- Provides for or yields the anticipated solution/result.

Software Validator **4.4.4.4** Evaluate the technical adequacy of software test cases by use of alternative methods such as:

- Hand calculations
- Other validated computer programs
- Standard problems with known solutions
- Experiments and tests
- Comparisons to confirmed published data correlations

Software Owner **4.4.4.5** Software validation documentation describes and includes as applicable:

- Hardware and software configuration
- Traceability of software requirements and design criteria
- Contain the results of V&V and testing activities

4.4.4.6 Document software test and V&V failures and determine if modifications to requirements, design, coding, or test plan are required.

Software Validator **4.4.4.7** Modification or revisions to existing approved software applications are subject to selective testing to detect unintended adverse effects that may have been introduced during the modification, to verify that the modifications have not caused unintended adverse effects, and to verify that the modified software or application still meets specified requirements.

Software Owner

4.4.5 Operations and Maintenance Phase

4.4.5.1 Once validated and approved for use, install in operating environment and implement user controls with approved procedures or instructions.

4.4.5.2 Verify software installation installs properly and satisfies the requirements for its intended use on host computer or system.

4.4.5.3 Document successful installation and ready for use.

4.4.5.4 Maintain software for proper operation and performance of intended function.

4.4.5.5 Document any changes or modifications to include testing and V&V activities performed.

4.4.6 Retirement Phase

IT and Software User

4.4.6.1 Terminate support or use for any software product or application that is retired.

4.5 Software Configuration Control

Software Owner or Developer

4.5.1 Establish software configuration control measures to include, as applicable, configuration identification, change control, and status control.

4.5.2 Place software under configuration control as each baseline element is approved. Controlled configured items can include:

- Documentation – plans, requirements, designs, user manuals, test reports, etc.
- Computer programs – source, backup files, media, etc.
- Support software

Software owner or Developer

4.5.3 Assign configuration identification number or identifiers, e.g. unique identification for each software item placed under software CM.

4.5.4 Establish configuration change controls to include, as applicable:

- Release and control process for baseline elements
- Changes to baseline elements
- Evaluation of baseline element change(s)
- Transmission of approved changes to affected organizations
- Verification and validation performed, as required, for change(s)

4.5.5 Maintain a process of configuration status accounting to include

as applicable:

- Listing of approved baseline elements and unique identifiers
- Status of proposed, in-process, or approved changes to the baseline elements
- History of changes to software items

4.6 Problem Reporting and Resolution

Software Developer or Owner

4.6.1 Report problems, errors, failures to affected users/owners and implement corrective measures. Issues affecting software quality or impact Project quality assurance should be reported through the issues management system per PRS-QAP-1210.

4.7 Software Use Controls

Software Owner

4.7.1 Control and document the use of released software to authorized users.

4.7.2 Implement necessary controls to limit unauthorized access to the software program or application.

4.8 Program Assessment

IT

4.8.1 An annual management assessment of SQA program element(s) shall be performed as scheduled in accordance with PRS-QAP-1420, *Conduct of Assessment*

QA

4.8.2 Independent assessment of SQA program compliance and effectiveness shall be performed at least once every two (2) years in accordance with PRS-QAP-1420.

5.0 RECORDS

The following records are generated per this procedure and maintained in accordance with PRS-DOC-1009, *Documents and Records*, as applicable.

- SQA plan
- Software requirements documentation
- Software design documentation
- User information manual/documentation
- V&V test results
- Installation verification documentation

6.0 SOURCE DOCUMENTS

- 10 CFR 830 Subpart A, *Quality Assurance Requirements*
- ASME NQA-1-2004, *Quality Assurance Program for Nuclear Facilities,* American Society of Mechanical Engineers
- DOE O 414.1C, *Quality Assurance*

- PRS-CDL-0058, *Quality Assurance Program Plan*
- DOE G 414.1-4, *Safety Software Guide for Use with 10 CFR 830, Subpart A, Quality Assurance Requirements, and DOE 414.1C, Quality Assurance*

Attachments A
DEFINITIONS/ACRONYMS
Page 1 of 3

DEFINITIONS

ACCEPTANCE TESTING – The process of exercising or evaluating a system or system component by manual or automated means to ensure it satisfies the specified requirements and to identify differences between expected and actual results in the operating environment.

ASSESSMENT – A review, evaluation, inspection, test, check, surveillance, or audit, to determine and document whether items, processes, systems, or services meet specified requirements and perform effectively.

COMMERICAL-OFF-THE-SHELF – Software that is available commercially and is defined by DOE software guidelines as acquired software. This type of software may include operating systems, compliers, database management, and spreadsheets programs.

CONFIGURATION MANAGEMENT – The process of identifying and defining the configuration of items in a system (software or hardware), controlling the release and change of these items through out the systems life cycle and recording and reporting the status of configuration items and change requests.

CONFIGURATION MANAGEMENT SOFTWARE ITEM – A work product that is managed under specified configuration management controls. Examples are software code and related documentation.

DESIGN PHASE – A phase in the software development life-cycle at which point the defined requirements are translated into design specifications.

DEVELOPMENT – The creation of new software applications.

FIRMWARE – The combination of hardware device and computer instructions and data that reside as read-only software on the device.

LIFE-CYCLE – The series of stage through which a software application passes during its lifetime.

MAINTENANCE – The modification and day to day support of existing software applications. Includes processes of supporting the application operational status, to correct faults, improve performance, or adapt to a changed operating environment.

OPERATION – The day-to-day use of existing software applications.

PHASE – A portion of a software development project that reduces it into manageable pieces and represents a measurable set of tasks that are performed to obtain specific work products.

PLANNING PHASE – A phase in the development life-cycle of a software application, during which the project is planned, approved and resources allocated.

PROCURED SOFTWARE APPLICATONS – Software that has been purchased or acquired

from an outside service provider, vendor, supplier or distributor, such as commercial-off-the shelf packages or government off-the-shelf packages.

QUALITY – The condition achieved when an item, service or process meets or exceeds the user's requirements and expectations.

QUALITY ASSURANCE – All of those actions that provide confidence that quality is achieved.

QUALITY ASSURANCE PROGRAM – The overall program or management system established to assign responsibilities and authorities, define policies and requirements, and provide for the performance and assessment of work.

REQUIREMENTS DOCUMENT – A document that specifies the manual and automated requirements for a software application; describes in detail what will be delivered in the software application.

RETIREMENT – Permanent removal of a software application from use and from its operational environment.

SOFTWARE – Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system.

SOFTWARE APPLICATION – A set of computer instructions intended to perform a specific task. Software applications may be spreadsheets, databases, process-control devices, or any other program used in the processing, gathering or generation of information.

SOFTWARE CODING – The point of software development where the design is translated into code.

SOFTWARE OWNER – As applied in this procedure is the person or organization who is responsible for the end use and "ownership" control of the software program or application for PRS work activities.

SOFTWARE QUALITY ASSURANCE – A process for systematic development, testing, documentation, maintenance and execution of software applications.

TEST PLAN - A document specifying the scope, approach, resources and steps of intended testing activities to verify and validate compliance with specified requirements.

USER – An individual who uses a software application.

VERIFICATION AND VALIDATION – The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements.

ACRONYMS

DOE – United States Department of Energy
COTS – Commercial Off-The-Shelf
CM – Configuration Management
IT – Information Technology
PRS – Paducah Remediation Services, LLC
QA – Quality Assurance
QAP – Quality Assurance Program
QAPP – Quality Assurance Program Plan
SQA – Software Quality Assurance
V&V – Verification and Validation