Jefferson Lab

Software Quality Assurance (SQA) Procedure

Document Number:	ITD-SQA-1	**Approval Date:**	7/22/2013
Revision Number:	Rev 1	**Periodic Review Date:**	7/22/2014
Document Custodian:	Kari Heffner		

1.0 Overview

Purpose

In order to best deliver the mission of the Laboratory, Thomas Jefferson National Accelerator Facility (Jefferson Lab) implements software quality measures using a graded approach. Software quality is considered part of overall Lab operations, and design and implementation of SQA processes and procedure are performed by line management, in accordance with standard business practices.

An integral part of Jefferson Lab's overall laboratory operations includes computer software systems. Software system requirements are defined by a designated system owner, and the associated software design processes and procedures are defined by the group developing the software system.

Additionally, the Lab utilizes DOE O 414.1D with a graded approach. Its software quality assurance requirements are used as part of the basis for developing Jefferson Lab's Software Quality Assurance (SQA) procedure and are implemented through line management and standard work processes.

Scope

This SQA procedure applies to all Jefferson Lab software.

2.0 Responsibilities

System Owner

- Establish controls for software programs in accordance with determined risk level.
- Maintain software appropriately as determined by the risk level.
- Maintain required SQA documentation; and make available during software inventories and reviews.

Chief Information Officer (CIO)

- Initiate Lab-wide software inventory.
- Review all SQA documentation for software determined to be Risk Code $\geq$3.

3.0 Requirements

Control	Develop an impact assessment for the software system, using the Software Impact Assessment (based on Jefferson Lab ES&H Manual Chapter 3210 Work Planning, Control, and Authorization Process) to determine software's Risk Code Level. (Those evaluated to be a Risk Code ≥ 3 are required to have written documentation detailing SQA processes and controls, including, but not limited to, change management, separation of privilege, dependencies on other software systems and failure mitigation.)
Maintenance	Software system maintenance is performed in accordance with group-defined procedures. Adequate procedures include, but are not limited to inventory, modification and review. (Reviews include Jefferson Lab's annual cyber security risk assessment, internal and external audits and safety reviews).
Inventory	Lab-wide software inventories are performed to evaluate software risk and dependencies. They include an evaluation of department SQA procedures for software determined to have a Risk Code ≥ 3.

4.0 Revision History

Revision #	Revision or Update:	Effective
1	This is new content written to reflect current laboratory operations	**7/22/2013**

5.0 Approvals

Approved by	Signature	Date
Document Custodian	Kari Heffner	**7/22/2013**
CIO and Information Technology Division AD	Roy Whitney	**7/22/2013**

Click to Certify

6.0 Appendix A – Risk Analysis Matrix

This appendix provides direction for determining the risk levels of software system.

Determine Consequence Level

Using Table 1 – Software Impact Assessment, determine the greatest impact that failure of the software could have.

Table 1 – Software Impact Assessment

Consequence Level	Severity
High (4)	Software failure could result in serious personnel injury or death, extended loss of facility operation, or significant impact on environment. Potential consequence(s) requires a rigorous series of actions. Activities at this level fall into a formal review and approval process, e.g.: the Accelerator Readiness Review.
Medium (3)	Software failure could cause temporary loss of facility operations, compromise PII, or have impact of >$100,000. Potential impact(s) justifies a disciplined set of actions.
Low (2)	Software failure would have limited impact on facility operations. Potential impact(s) justifies a higher degree of proficiency or supervisor consideration.
Extremely Low (1)	Software failure would have very limited impact, with very limited scope, on facility operations and would be readily detected. Potential impact(s) justifies limited supervisor oversight.
Negligible (0)	No potential impact, or software will produce scientific results that will be reviewed via peer review process (ANSI/ASQ Z1.13-1999).