Software Quality Assurance Implementation Plan

Implementation Overview

American Nuclear Society Meeting
November 17, 2004

Chip Lagdon/Subir Sen

Office of Quality Assurance Programs
Office of Environment, Safety and Health
U.S. Department of Energy

Presentation Outline

- What the SQA IP has achieved
- What tasks remain to be completed
- What lessons have we learned

SQA IP Achievements

- Central Registry established
- Gap Analysis completed for toolbox codes
- Code Guidance Reports issued
- SQA Knowledge Portal & List Server in use
- Design Code Survey conducted
- SQA Assessments conducted

SQA IP is a Success

- Issues identified and corrected by the field
  - Eberline Meter
  - AMWTP Software Issues
- Assessments have increased awareness by DOE & contractors
- Guidance reports & Gap Analyses are being used by safety analysts
- Continue to train and communicate SQA
- Policy & guidance will create uniformity in approach

Remaining Tasks

- Continue to support training and qualification of SQA personnel
- Begin upgrading toolbox codes
- Complete comment resolution and issue SQA Order and Guide
- Institutionalize SQA under existing QA programs

Lessons Learned

IP Development

- IP commitments must be well defined with a clear understanding of the deliverables
- Personnel executing the IP should be actively involved in developing the IP
- The sequence of IP activities requires careful consideration
  - Train and Qualify personnel before Assessments
    - ASQ SQE course plus assessment experience
    - Provide better guidance to field
  - Code Guidance Reports after Gap Analysis
  - Central Registry established early

Lessons Learned (cont.)

Roles and Responsibilities

- DOE organization responsible for IP must have the authority and budget to implement the plan
- SME Panel was to be used to supplement experience based but was not properly utilized
- Coordination and communication between organizations was not always effective

Lessons Learned (cont.)

Schedule & Expectations

- The IP schedule must be realistic and resource loaded
    - Changing IP schedule is difficult
- Original Central Registry concept in IP was not practical
    - Toolbox code ownership issues made Central Registry and upgrades difficult
- DOE Order development is time consuming without a technical authority

Assessments – Lessons Learned

- Software Requirement Specification (SRS) and Software Design Document (SDD) are essential for developing quality software and life cycle maintenance.
    - Majority of software projects did not have SRSs and SDDs
    - Sites using the SRSs and SDDs have clear understanding of what was needed to develop and maintain software quality.
    - The sites without SRSs and SDDs appeared to be relying heavily on the available experts to ensure software is developed or procured to meet the project needs.

Assessments – Lessons Learned (cont.)

- Software procurement specifications should specify details of software requirements, not just catalog data.
  - Sites procuring PLC’s for process systems only specified the vendors’ catalog model information as procurement specifications
  - Supporting documentation for the suitability and applicability of the technical requirements not included

Assessments – Lessons Learned (cont.)

- Formal procedures for software problem reporting and corrective actions for software errors and failures need to be maintained and rigorously implemented.
  - Many sites resolve software errors and corrective actions at the project level and maintain informal coordination with vendors or other effected entities.
- Software quality assurance program and procedures should be rigorously implemented.
  - Assessments revealed inconsistencies in the requirements contained in the SQA program and procedures and their implementation.
  - Many sites rely on individual expertise and their personal effort and put less importance on corporate program.

Assessments – Lessons Learned (cont.)

- Appropriate qualifications and training on software use is essential for proper use of safety software.
  - Very sophisticated and complex software are being used without appropriate training in their use.
- Appropriate software control and configuration management are essential for safe use of the software.
  - Lack of proper control has resulted in multiple versions being available at the same time and even some with known errors.
  - Deficiencies have been noted with configuration control in terms of software version and documentation.
  - Inconsistencies exist in the requirements contained in the SQA program and procedures and their implementation.