Quality Assurance Plan

for

National Environmental Laboratory Accreditation Database System

TNI National Database Committee

December 2006

Table of Contents

Quality Assurance Plan for National Environmental Laboratory Accreditation Database System

1.0 INTRODUCTION

1.1 Background

The NELAC Institute (TNI) National Database Committee was originally formed in 2005, as an Institute for National Environmental Laboratory Accreditation (INELA) committee, and reformed in 2006 under TNI in order to provide the impetus to create a national central repository for information regarding the accreditation status of environmental laboratories in the United States. This database system (DBS) will be available to all accrediting and enforcement agencies, as well as the general public.

1.2 Purpose

This TNI National Database Quality Assurance Plan (QAP) describes the standards, processes and procedures used to ensure delivery of a high-quality, professional end product. The QAP establishes the authority of the TNI National Database Committee to dictate and impose the requirements set forth in this QAP for the development, deployment and maintenance of the DBS. The QAP will ensure Quality Assurance (QA) oversight of procedures, policies, monitoring activities, and evaluation processes to be used in order to determine acceptability. This QAP provides standards against which the quality of the product/service being provided and the progress toward completion can be measured. QA activities concentrate on the prevention of problems through the continuous process of validation and improvement.

This QAP serves as a guide to the QA activities and may be tailored with the approval of the TNI National Database Committee in order to respond appropriately to changes in project needs.

2.0 SCOPE

The scope of this plan covers QA measures for the DBS during all stages of planning, development, testing, training, maintenance and user support.

- Product standards
- Organizational structure
- Roles and responsibilities
- Quality indicators for software, hardware and data storage
- Planning requirements and procedures
- Testing procedures
- Training documentation requirements
- Quality assurance activities
- Problem reporting and corrective actions

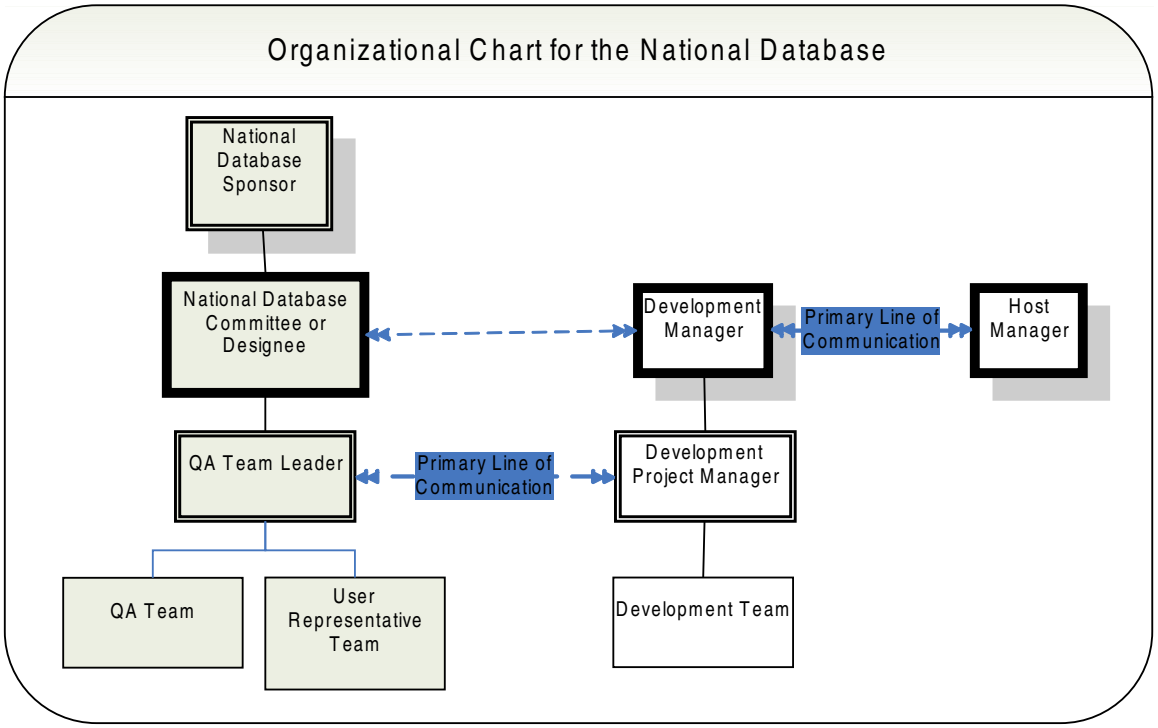
- Walkthrough procedures
- Audit procedures

The following items are not within the scope of this QAP:

- Product specifications
- Development project plans
- Finished product

3.0 MANAGEMENT ROLES

3.1 Organizational Structure



3.2 Specific Roles

3.2.1 Sponsor

- Provides funding for the development of the DBS.
- Provides the authority to withdraw funding from the project if the TNI National Database Committee indicates that the development agency is not able to meet the specifications set forth in this QAP.

3.2.2 TNI National Database Committee

- Reserves the right either to perform assigned tasks or designate group tasks to an appropriate party.
- Develops the QAP.
- Provides oversight for the QA Team Manager.
- Communicates with the sponsor on an as-needed basis in order to keep the sponsor up to date on the project status or obtain support needed by the QA Team.

3.2.3 Quality Assurance Team Leader

- Provides oversight for the QA Team.
- Provides a means of communication to the TNI National Database Committee.
- Ensures that the independence of the contracted computer systems personnel is maintained.
- Ensures that the QA team has access to the data, SOPs, and other records pertaining to the operation and maintenance of the computer systems.
- Ensures that all deviations from the specifications set forth in this QAP have been corrected, approved and documented.

3.2.4 Quality Assurance Team

The Quality Assurance Team consists of a technical team of computer and IT specialists and appropriate stakeholders.

General

- Provides QA oversight for the creation and maintenance of the DBS.
- Adheres to the processes, procedures and standards defined in the QAP.
- Establishes processes and procedures that accurately verify and validate the adherence to the QAP and project plan.
- Maintains involvement throughout the entire project.

Communication

- Works to foster constructive communication, provide feedback to detect and prevent development problems, discuss alternative solutions, and ensure quality is built into the DBS and related services.
- Maintains an on-going dialogue with the development staff.

Auditing

- Conducts audits and reviews specified deliverables according to the QAP.
- Reports results of audits to the QA Team leader.
- Ensures that products adhere to the standards set forth in the QAP.

- Ensures that the expectations set forth in the TNI National Database QAP and Development plan are met.
- Ensures that all documentation containing requirements are understood by the development team.

Documentation

- Approves the project plan created by the developer.
- Identifies additional documentation that may be needed for specific tasks.
- Ensures that all issues arising, audits or user comments are addressed appropriately.

3.2.5 Customer Representatives

- Represent each major stakeholder.
- Provide feedback in the planning and testing stages of the project from the perspective of the average user.
- Reports any suggestions to the QA team leader.

3.2.6 Development Organization Manager

- Oversees all aspects of product development and maintenance or the maintenance agreement on a high level.
- Confirms the independence of the QA function.
- Ensures that staff and other resources are available as needed to support QA.
- Reviews QA audits and reports.
- Gives final approval for completed product and ongoing support.
- Ensures staffing is adequate to finish the DBS project within the timeline specified by the project plan.
- Ensures that staff has adequate education, training and experience to perform assigned computer systems functions.

3.2.7 Development Project Manager

The development project manager is a subject matter expert in IT project management.

General

- Ensures compliance with the QAP.

Communication

- Ensures that QA activities are understood by all staff and that time is allotted for QA activities.
- Works with developers and the TNI National Database Committee to define a timeline acceptable to both parties.

- Ensures that all parties involved are kept up to date by following the communication plan.

Management

- Provides oversight for all aspects of project development for the development organization.
- Ensures problems are resolved in accordance with the QAP.
- Ensures that the project plan is approved by the TNI National Database Committee.
- Ensures that the QA team has access to the project information in order to carry out the activities defined in this QAP.

Documentation

- Writes the project plan according to specifications provided by the TNI National Database Committee.
- Works with developers to define the work plan.
- Responds to deficiency reports from QA reviews and audits.
- Tracks the status of defects/errors until closed.

Development Team/Technical Staff

- Implements task level quality control based on QA standards, policies, and procedures.
- Participates in reviews and audits.
- Performs corrective actions or processes improvements in response to QA findings.
- Manages and controls defects/errors and corrections.

4.0 PLANNING AND DESIGN

All required documents for the TNI National Database project will follow the appropriate standards concerning content and format, as outlined in the development plan. Required documentation includes the following:

4.1 The Database System Development Plan

The database system development plan must include:

- Scope of the project
- Constraints and assumptions
- Functional requirements
- System specifications
- Software design requirements

- Risks and risk management

4.1.1 Needs Assessment and High-Level Requirements Definition

The needs of the users must be assessed during the development process. This can be accomplished through literature searches, interviews with users and other expert advice. A high-level requirements document (referred to as the Functional Requirements document) must be created. This document is an overview of all the functions the system must perform. It is the foundation of the system specification.

4.1.2 Detailed Requirements Analysis

Documentation needs to be generated to concisely describe how the features in the functional requirements will be implemented. This specification will also include any configuration information that might be necessary to properly implement the feature(s) requested. This set of information is the blueprint from which the software is built. Typically requirements include the following:

- All inputs that the software system will receive
- All outputs that the software system will produce
- All functions that the software system will perform
- All performance requirements that the software must meet, e.g., data throughput, reliability, timing, etc
- The definition of all user interfaces
- What constitutes an error and how errors should be handled
- The intended operating environment for the software, e.g., hardware platform, operating system, etc
- All ranges, limits, defaults and specific values that the software will accept

This phase of the development process will result in a detailed requirements document (also referred to as the System Specification). This document will define all critical functions that the completed system must support. Performance goals must also be included in the requirements. The System Specification document should be reviewed by the end-user, if possible, to ensure that critical functions and other requirements have not been overlooked.

4.1.3 Software Design

In the design phase, software requirements are translated into a logical and physical representation of the software to be implemented. To enable persons with varying levels of technical responsibilities to clearly understand design information, it may need to be presented both as a high level summary of the design, as well as a detailed design specification.

The completed software design specification should constrain the programmer/coder to stay within the intent of the agreed upon requirements and design. The software design specification should be complete enough that the programmer is not required to make ad hoc design decisions.

The software design specification describes the software's logical structure, parameters to be measured or recorded, information flow, logical processing steps, control logic, data structures, error and alarm messages, security measures, and predetermined criteria for acceptance. It also describes any supporting software (e.g., operating systems, drivers, web browser, other applications), special hardware that will be

needed, communication links among internal modules of the software, links with the supporting software, links with the hardware, and any other constraints not previously identified.

The software design specification may include:

- Data flow diagrams
- Program structure diagrams
- Control flow diagrams
- Pseudo code of the modules
- Context diagrams
- Interface/program diagrams
- Data and control element definitions
- Module definitions
- Module interaction diagrams

4.2 The System Configuration Management Plan

Changes need to be defined and a tracking process must be established. Procedures for completing a change should be written and should define the type of change and reasons for the revision. This plan needs to be applicable to both hardware and software processes. Since the failure mechanisms are different for the two processes, the changes must be defined and classified for the specific type of process. The plan must apply to tracking both production and development issues. Change orders will be the primary form of communication and will require approval by the designated QA liaison. The system configuration management plan must include:

- Change management and control process
- System baseline definition
- Quality indicators
- Implementation Plan
- Testing, Verification, and Validation Plan
- Defect reporting system
- Communication Plan

4.2.1 Implementation Plan

Implementation is the activity where detailed design specifications are implemented as source code. Programming usually involves the use of a high-level programming language and may also entail the use of lower-level languages for time-critical operations. The source code may be either compiled or interpreted for use on a target hardware platform. Uniform standards for code during this phase will significantly improve the ability to maintain the software. Coding standards should include conventions for clarity, style, complexity management, and commenting. Code comments should provide useful and descriptive information for modules, including expected inputs and outputs, variables referenced, expected data types, and operations to be performed.

Routine design/implementation team meetings are essential during this stage to ensure that the design specifications are being met. Also, configuration control procedures must be established to ensure that changes in the specifications are documented and the implementation schedule adjusted accordingly.

4.2.2 Testing, Verification, and Validation Plan

The newly developed system needs to be tested. The purpose of testing is not simply to detect errors but also to verify that the completed software meets user requirements. In designing any test, the "correct" or "acceptable" outputs should be known in advance, if possible. Testing should be planned in an orderly, structured way and be documented.

Test plans should be created during software development phases. They should identify the test schedules, environments, resources [personnel, tools, etc.], methodologies, cases [inputs, procedures, outputs, expected results], documentation and reporting criteria. Individual test cases should be definitively associated with particular specification elements and each test case should include a predetermined, explicit, and measurable expected result, derived from the specification documents, in order to identify objective success/failure criteria.

Source code should be evaluated to verify its compliance with specified coding standards. It should also be evaluated to verify its compliance with the corresponding detailed design specifications. Source code evaluations are implemented as code inspections and code walkthroughs. Appropriate documentation of the performance of source code evaluations should be maintained as part of the validation information.

Installation testing should be performed through either actual or simulated use of the software being tested within the environment in which it is intended to function. This testing should follow a pre-defined plan with a formal summary of testing and a record of formal acceptance. There should be retention of documented evidence of all testing procedures, test input data and test results. There should be evidence that hardware and software are installed and configured as specified. Measures should ensure that all system components are exercised during the testing. The testing methods should encourage use through the full range of operating conditions and should continue for a sufficient time to allow the system to encounter a wide spectrum of conditions and events in an effort to detect any latent faults, which are not apparent during more normal activities.

During design and development planning, a software validation plan needs to be created to identify necessary validation tasks, procedures for anomaly reporting and resolution, validation resources needed, and management review requirements including formal design reviews. A software life cycle model and associated activities should be identified, as well as validation tasks necessary for each software life cycle activity.

Typically each validation task requires personnel as well as physical resources. The validation plan should identify the personnel, facility and equipment resources for each validation task. A configuration management plan should be developed that will guide and control multiple parallel development activities and assure proper communication and documentation. Controls should be implemented to assure positive and correct correspondence between all approved versions of the specifications documents, source code, object code and test suites which comprise the software system, accurate identification of the current approved versions, and appropriate access to the current approved versions during development activities.

Procedures should be created for reporting and resolving all software and validation anomalies. Management should identify the validation reports, and specify the contents, format, and responsible organizational elements for each report. Procedures should also be created for the review and approval of validation results, including the responsible organizational elements for such reviews and approvals.

A validation report should be created to complete the validation process. The summarized data and evaluations should support the conclusions that the system has been validated. The report must be reviewed for accuracy before the approval process. Review and approval personnel will be able to defend their reasons for approving the validation report. The validation report becomes a part of the facility's documentation, with well-defined storage and retrieval procedures. The testing, verification, and validation plan must include:

The specific validation tasks for each life cycle activity.

- Methods and procedures for each validation task.
- Criteria for initiation and completion (acceptance) of each validation task.
- Inputs for each validation task.
- Outputs from each validation task.
- Criteria for defining and documenting outputs in terms that will allow evaluation of their conformance to input requirements.
- Roles, resources and responsibilities for each validation task.
- Risks and assumptions.

4.3 Database System Hosting Provider

A complete description and identification of the installed hardware should be created. The manufacturer's name, the model numbers, and the configurations must be clearly documented. A block diagram of the architecture with component parts will serve as a focal point in identifying the hardware and understanding the system's operation.

System maintenance responsibilities will be clearly defined, and all maintenance of hardware will be documented. Environmental requirements need to be addressed relevant to the installation and must be documented.

Backup procedures and contingency plans for when the system fails will be required. Since the possibility of a system failure always exists, contingency plans will need to be put in place to address what happens in the event of power failures, hardware crashes, etc. It will be required to define how redundant systems take over if the primary system fails and how system processes will be affected. Issues such as the following will need to be addressed:

- What effects on quality could occur if hardware fails
- How will reactions to system failures be documented
- Time allotment window to perform system recovery and restore online services after a system failure

5.0 QUALITY INDICATORS

5.1 Product - Hardware

- Adequately designed to ensure reliability and maintainability and of adequate capacity to accommodate all records and function in a timely manner.
- Installed and operated in accordance with manufacturer’s recommendations.
- Tested to ensure conformance to pre-determined acceptance criteria prior to use.

- Adequately tested, inspected and maintained.
- Housed in environmental conditions that are regulated to protect against data loss.
- Provided adequate storage capability for retention of data storage.
- Made readily available all documentation and records pertaining to the computer system.

5.2 Data

- Retention of data, documentation and records pertaining to the computer system complied with EPA contract, statute or regulation.

5.3 Software

- Tested to ensure that the software accurately performs its intended functions.
- Adequately documented problems that arise from upgrading or replacing software adequately documented.
- Applied version control methods to document the revision currently in use.
- Physically secured to prevent tampering or other adverse actions.
- Periodically changed system passwords
- Controlled authorization to change the system password.
- Restricted access to software and data to authorized personnel.
- Required system users to periodically required change their passwords.
- Established access categories for various levels.
- Maintained and document a list of authorized personnel and their level of access.

5.4 Testing

- Includes testing of each element and each choice or combination of choices that could influence functionality on a system that is identical to the software that the user will experience.
- Includes load testing with a load that will approximate the highest estimated simultaneous use scenario.
- Includes user testing by the customer representatives on a system that is identical to the proposed end product.
- Tests are documented in enough detail to ensure that the results of the test are complete and easily understood.

5.5 Maintenance

The maintenance agreement will be approved by the TNI National Database Committee and will include the following:

- Terms of Agreement
- Length of Agreement

- Hosting, hardware and software maintenance
- Backup, archiving and disaster recovery plan

5.6 Training

The training deliverables will include the following in either printed or electronic format. The QA team may use an independent expert to evaluate the completeness of the technical training materials.

- User manual including basic instructions on how to use the software.
- Help desk manual in the form of a standard operating procedure.
- Database maintenance manual in the form of a standard operating procedure describing all technical aspects of the system in a format that an expert in the chosen software can easily understand including common errors and solutions and system maps.

5.7 Security

There are many different types of threats to data security and communications. Since there are many potential security vulnerabilities, planners should identify as many of these as possible and state explicitly how they will be prevented. Specific tests should be conducted that address the security features of the system. Some specific methods for addressing security vulnerabilities include the following:

5.7.1 General

System must provide separate passwords for user log on, for remote dial-in, and for access to sensitive portions of a database.

5.7.2 Privacy

The system may contain records of proprietary business data, these records must be kept private. Encrypting the identifying records and restricting use of the application to only personnel with special password-protected access will ensure customer privacy.

5.7.3 Data Defensibility and Traceability

When many different users have read/write access to a common data set, assurance of data integrity will be a concern. If absolute traceability of each data item is required, an Audit Trail, which records each transaction and includes the date, time, and person responsible for the change, will be required.

6.0 QUALITY ASSURANCE ACTIVITIES

6.1 Problem Reporting and Corrective Action

- The problem reporting and corrective action process of this project will be governed by the configuration management plan. TNI National Database corrective actions either during development or after implementation must be itemized, documented, tracked to closure, and reported by the QA team.
- Problem resolution will include documentation of root cause analysis and corrective action.
- Problems are resolved with the developer, host, or the appropriate task leader when possible. Problems that cannot be resolved with the technical team or task leader are elevated to the project sponsor.

- Problems that have been referred to the project sponsor are reviewed weekly until they are resolved.

6.2 Walkthrough Procedure

- Walkthroughs will be used to evaluate plans, documentation and other deliverables and will serve to orient staff members to new developments in the project.
- Walkthroughs shall be conducted on an as-needed basis.
- Records of these walkthroughs shall be maintained by the developer, along with issues that were identified and resulting action to be taken.
- The QA team shall determine if issues should be accepted as is or addressed.

6.3 Audit Procedures

- Audits are used to assess compliance with this QAP and associated documents. Quality factors include but are not limited to:
- Correctness: The extent to which the software and documentation satisfies the specification document and the quality assurance plan.
- Timeliness: The software and documentation are provided when needed to the QA team.
- Reliability: The extent to which the software performs on a consistent basis.
- The QA team will plan an audit schedule appropriate to the timeline agreed upon for completion of development.
- Audits will seek to verify that documentation is complete, correct and in compliance with the criteria set forth by the TNI National Database Committee.
- The QA team will summarize audit findings for the software developer, QA Team leader and other stakeholders.

7.0 REFERENCES

1. IEEE Standard for Software Quality Assurance Plans (Std 730-1998).

2. EPA Office of Technology Operations and Planning “Interim Agency System Life Cycle Management Procedures”, 3/20/2004.

APPENDIX A

Quality Assurance Checklist

ITEM		Y/N	COMMENT
MANAGEMENT– When data are collected, analyzed, processed, and maintained:			
1.	Does the computer support staff clearly understand the function(s) they are to perform with the computer system?		
2.	Is there a Quality Systems Officer to oversee system operations?		
3.	Is there an Information Management Manager to oversee system operations?		
4.	Are there adequate personnel, resources, and facilities to complete the work as scheduled?		
5.	Are internal QA audits of the system and data performed, and are corrective actions taken in response to any deficiencies?		
6.	Are approved standard operation procedures for computer applications in place?		
PERSONNEL – Does management ensure that all computer support staff:			
7.	Have adequate education, training and experience to perform assigned computer systems functions?		
8.	Have a current summary of their training, experience and job description, including their knowledge relevant to computer application design and operation maintained at the facility?		
9.	Are of sufficient number for timely and proper operation of the system?		
QUALITY ASSURANCE – Does a Quality Assurance Officer:			
10.	Maintain independence of the contracted computer systems personnel and report back directly to management?		
11.	Have access to the data, SOPs and other records pertaining to the operation and maintenance of the computer systems?		
12.	Conduct periodic inspections, report problems that may		

ITEM		Y/N	COMMENT
	affect data integrity, recommend actions to be taken, and schedule dates for re-inspection?		
13.	Periodically audit the data to ensure integrity?		
14.	Ensure that his/her records are documented and appropriately indexed?		
15.	Ensure that the storage media on which data resides are identified and documented?		
16.	Ensure that the individual(s) responsible for entering data is (are) uniquely identified when the data are recorded and the time and date documented?		
17.	Ensure that the computer system transmitting data is uniquely identified when the data are recorded and the time and date documented?		
18.	Ensure that the procedures and practices to verify accuracy of data are documented and managed as described in the SOPs?		
19.	Ensure that the procedures and practices for making changes to data are documented and provide evidence of change, preserve the original recorded documentation are dated, indicate the reason for the change, and identify the person who made the change?		
20.	Maintain independence of the contracted computer systems personnel and report back directly to management?		
SOFTWARE –			
21.	Are testing and quality assurance methods implemented to ensure that the software accurately performs its intended functions?		
22.	Are problems that arise from upgrading or replacing software documented?		
23.	Are version control methods used that document the software revision currently used?		
24.	Is documentation established and maintained to demonstrate the validity of the software?		
SECURITY –			
25.	Is the system physically secure?		
26.	Is the system password changed periodically?		

ITEM		Y/N	COMMENT
27.	Is authorization to change the system password tightly controlled?		
28.	Are log-ons, passwords, etc. used to restrict access to authorized personnel?		
29.	Are system users periodically required to change their passwords?		
30.	Are access categories established for various levels?		
31.	Is there a list of authorized personnel and their level of access?		
HARDWARE – Are computer hardware and communications equipment:			
32.	Of adequate design and capacity?		
33.	Installed and operated in accordance with manufacturer's recommendations, and at installation undergone acceptance testing that conforms to predetermined acceptance criteria?		
34.	Adequately tested, inspected and maintained?		
FACILITIES – Do facilities personnel ensure that:			
35.	The environmental conditions of the facility housing the computer systems are regulated to protect against data loss?		
36.	The facility has environmentally adequate storage capability for retention of data storage media, documentation and records pertaining to the computer system are provided?		
37.	The retention of data, documentation and records pertaining to the computer system complies with EPA contract, statute or regulation?		