

Software Quality Assurance Plan

Highland Basic Order Tracking System

Prepared For: Highland Office Supply
Prepared By: John Steele
Elucidata

Document ID: 100-01-03
Version: 01c

Copyright (©) 2001 Elucidata
All Rights Reserved.

TABLE OF CONTENTS

INTRODUCTION	4
SCOPE	4
METHODOLOGY.....	4
DELIVERABLE CLASS STANDARDS	7
ASSESSMENTS AND REVIEWS	8
FORMAL ITERATION REVIEWS	9
IN-STAGE ASSESSMENTS	10
STAGE EXIT REVIEWS.....	10
VERIFICATION AND VALIDATION	11
VERIFICATION.....	11
VALIDATION	12
CONCLUSION	14
DELIVERABLE CLASS STANDARDS.....	14
TARGETED REVIEWS & ASSESSMENTS.....	14

INTRODUCTION

This document represents the Software Quality Assurance Plan for the Highland Basic Order Tracking System software development project.

Software Quality Assurance (SQA) is a formal process for evaluating and documenting the quality of the work products produced during each stage of the software development lifecycle. The primary objective of the SQA process is to ensure the production of high-quality work products according to stated requirements and established standards.

SCOPE

This SQA process is tailored to fit the current software development effort, and is related to the project planning and lifecycle description documents for this project. This project is classified as a small to medium database development effort; large software development efforts require more rigorous controls and more frequent reviews than are described here.

METHODOLOGY

The methodology presented here is based on the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) and the Institute for Electrical and Electronics Engineers (IEEE) standards for Information Management. This SQA process:

- Makes use of the principal project participants as defined in the Vision & Scope chapter of the project planning document for this software development project.
- Defines deliverable class standards to be applied during assessments and reviews.
- Identifies assessment and review points and how they will be conducted.
- Identifies the SQA work products produced as a result of the reviews and assessments.

The software development lifecycle for this project defines a series of stages; each project stage is defined as a separate operation with specific inputs and outputs. This SQA Plan implements assessments and reviews at specific points within each of these stages. The complete software development lifecycle for this project is described in a separate document, available at:

<http://www.elucidata.com/refs/sdlc.pdf>

Please refer to the SDLC document for a description of the structure, inputs to and outputs from each of the stages of the SDLC, as well as a comparison to other SDLC models. The terms and stage descriptions defined there are used extensively in this SQA plan.

Other terms common to the software development process are defined in a Glossary of Software Engineering Terms, available at:

<http://www.elucidata.com/refs/seglossary.pdf>

Please refer to this glossary for definitions of the terms used in this document and in the SDLC document.

FORMAL REVIEWS

For each project deliverable, as many as three types of formal reviews are conducted after the end users and development team have informally agreed that the deliverable content is accurate. The three review types are:

1. End-user review, conducted by at least one Subject Matter Expert (SME) who is familiar with the software product under development.
2. Technical review, conducted by at least one experienced software developer who is familiar with the product under development.
3. Quality Assurance review, conducted by an independent Quality Assurance Reviewer (QAR).

Each review is conducted in alignment with the reviewer's area of expertise, and in accordance with the review items described in the associated Deliverable Class Standard and review form. Refer to Chapter 2 for a discussion of Deliverable Class Standards. By tailoring the review focus to the expertise of the reviewer, this SQA plan prevents redundancy and inappropriate reviews.

PERSONNEL ROLES AND RESPONSIBILITIES

In a small database development effort, three principal roles are defined:

- Primary End-user Representative (PER)
- Primary Developer Representative (PDR)

- Quality Assurance Reviewer (QAR)

The PER acts as the primary point of contact and principal approver for the end-user community. The PER is responsible for ensuring that end-user reviews are conducted on time and by appropriate subject matter experts.

The PDR acts as the primary point of contact and principal approver for the developer community. The PDR is responsible for the conduct of technical reviews in a timely manner and by appropriate development team members.

The QAR acts as the independent quality assurance reviewer for the project. The QAR will work independently from the development team to ensure objective audits and reviews of the work products and processes of this software development project.

STANDARDS

The following standards were used as guides to develop this SQA process. The standards were reviewed and tailored to fit the specific requirements of small database projects using the referenced SDLC.

- ANSI/IEEE 730.1: Standard for Software Quality Assurance Plans
- ANSI/IEEE 1028: Standard for Software Reviews and Audits
- ANSI/IEEE 1012: Standard for Software Verification and Validation
- SEI/CMM: SQA Key Process Area

DELIVERABLE CLASS STANDARDS

One of the most important parts of maintaining high quality output is the establishment of realistic expectations in the minds of the project participants. Those personnel tasked with the production of project deliverables need to have a clear picture of what they are expected to produce. Conversely, when a deliverable is produced, those personnel tasked with reviewing the deliverable need to be working from the same picture. This picture is known as the Deliverable Class Standard (DCS).

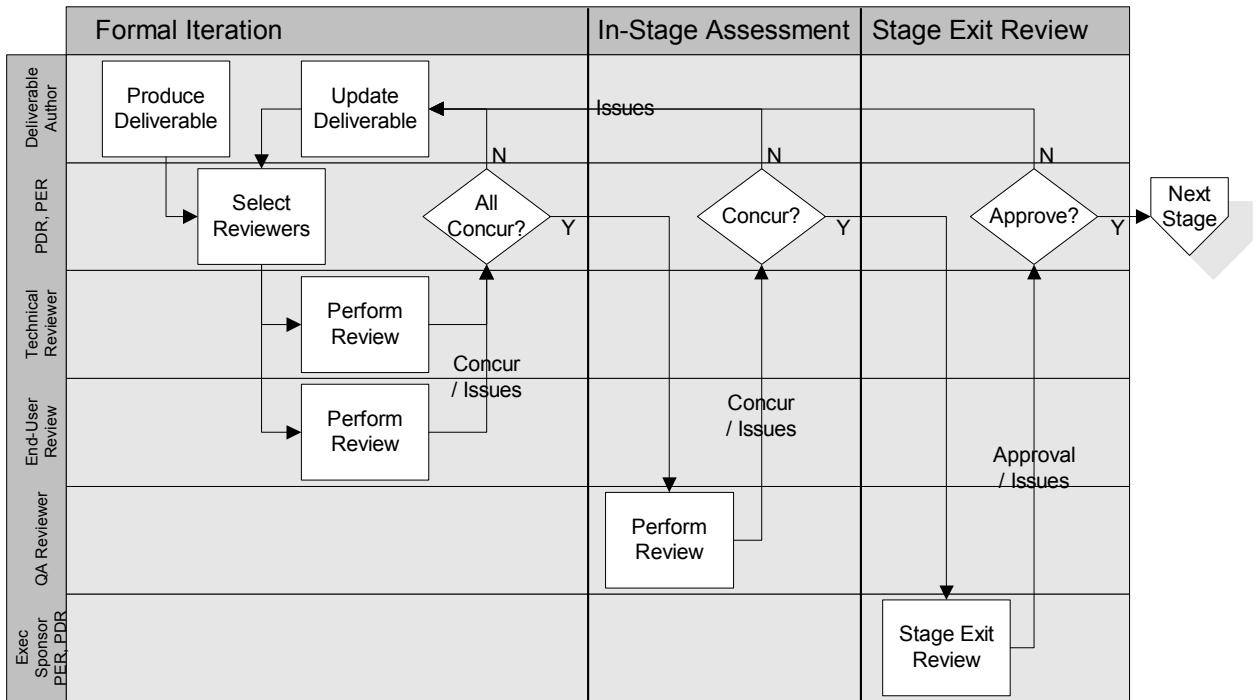
A DCS specifies the structure and content expected for a specific class of deliverables. A deliverable generally evolves from stage to stage in a software project. For example, the Requirements Traceability Matrix (RTM) is created during the Requirements Definition stage, but is very limited at that time. By the time the RTM has had the results of several successive stages added, it has evolved into a significant and fairly complex document. As the RTM evolves from stage to stage, the content requirements (and corresponding review checklists) need to evolve as well. All of the staged releases of the RTM, considered together, constitute the RTM "class" of deliverables.

Each DCS specifies the structure and content requirements for a specific class of deliverables. It describes the purpose of the class, the stages that produce deliverables of that class, and the expected structure and content of the deliverable for each stage. The following deliverable class standards are incorporated into Appendix A of this SQA Plan:

- Project Plan & Schedule DCS
- Requirements Traceability Matrix DCS
- Requirements Document DCS
- Design Document DCS
- Online Help DCS
- Implementation Map DCS
- Test Plan DCS
- Production Data Source DCS
- Production Initiation Plan DCS
- Acceptance Plan DCS

ASSESSMENTS AND REVIEWS

At the beginning of this project's SDLC description, a generic process is defined that applies to each stage. The generic process description includes references to reviews, in-stage assessments and stage exit reviews as quality assurance elements, with references to this document. These assessments and reviews occur during the formal iteration, in-stage assessment, and stage exit processes:



The intent of these assessments and reviews are to assure that the established system development and project management processes and procedures are being followed effectively, and that any risks to the current project plan and schedule are identified early and addressed quickly.

FORMAL ITERATION REVIEWS

As described in the SDLC, the formal iteration process develops the "final" versions of the current stage deliverables, based on the information and documents developed during the previous informal iteration process. End-user and technical reviews are performed to close out the formal iteration process.

END-USER REVIEWS

Each class of deliverable must be reviewed by at least one Subject Matter Expert (SME) who is familiar with the software product under development. The SME will examine the deliverable for the presence of attributes specific to each class of deliverable, as described in the DCS. The intent of the end-user review is to insure that each deliverable is examined from the point of view of the ultimate users of the system, by someone who is knowledgeable about the process being automated.

TECHNICAL REVIEWS

Each class of deliverable must be reviewed by at least one development team member who is familiar with the product under development. This review will be conducted from a technical point of view, with the reviewer examining the deliverable for the presence of attributes specific to each class of deliverable, as described in the DCS. The intent of the technical review is to insure that each deliverable is examined for technical accuracy by someone who is familiar with the processes and development tools for the project. In other development methodologies, technical reviews may be known as "peer reviews," or "code walk-throughs," depending on the lifecycle stage of the project.

CONCURRENCE

Each reviewer conducts a review of the current stage deliverables for structure and content in compliance with the DCS, using the appropriate review form for each class of stage deliverable. When all reviewers have indicated substantial or unconditional concurrence on the review forms for each deliverable, the PDR notifies all team members that the formal iteration process is closed, pending successful completion of the in-stage assessment.

In the event a technical or end-user reviewer indicates non-concurrence during a review, an issue is raised and submitted to the Configuration Review Board (CRB). The composition and functionality of the CRB is fully described in the Configuration Management Plan for this project. The CRB works with the reviewers and document authors to resolve the issue(s) raised, at which point the appropriate reviews are repeated. This cycle continues until concurrence is achieved.

IN-STAGE ASSESSMENTS

The Quality Assurance Reviewer (QAR), at the request of the Primary Developer Representative (PDR), conducts in-stage assessments. For projects of this size, the review occurs after technical and end-user reviews concur on the content of the deliverables in the Formal Iteration process of a stage. The QAR reviews stage deliverables for structure and content in accordance with the review form for each class of stage deliverable. The review form for each class of stage deliverable is incorporated into the relevant DCS, as shown in Appendix A.

If the QAR indicates substantial or unconditional concurrence on the appropriate review forms, the in-stage assessment process is considered to be successful. Final document versions are placed under configuration control, and posted at the document distribution site. At this point, the PDR notifies the project team that the formal iteration process is closed, and the in-stage assessment was successful via the In-Stage Assessment Report.

If the QAR is unable to concur with one or more deliverable review form statements, the CRB works with the QAR and appropriate project personnel to revise the deliverable in accordance with the issues raised. The formal iteration process is then reopened for the affected deliverables, and a new set of end-user and technical reviews is performed. The project iterates between the formal iteration process and the in-stage assessment process until all reviewers indicate substantial or unconditional concurrence on their review forms.

STAGE EXIT REVIEWS

The Primary End-user Representative and the Primary Developer Representative, with the project Executive Sponsor in attendance if possible, conduct stage exit reviews. The purpose of a stage exit is to allow all personnel involved with the project to review the current project plan and stage deliverables, provide a forum to raise issues and concerns, and to ensure an acceptable action plan exists for all open issues. Refer to the SDLC for a more complete description of the stage exit process.

If any substantial issues are raised during the stage exit review, the PDR and PER, together with appropriate project team members, work to resolve the issues and modify any effected deliverables. At this point, the project has "backed up" to the formal iteration process. The project then continues forward as previously described.

If no substantial issues are raised, the minutes of the stage exit review are entered as a formal document of record, with either physical or digital signatures of the project executive sponsor, the PER, and the PDR. At this point, the current stage is considered to be successfully concluded.

VERIFICATION AND VALIDATION

The following IEEE definitions apply to this SQA plan:

Verification: The process of determining whether or not the products of a given stage of the software development life cycle fulfill the requirements established during the previous stage.

Validation: The process of evaluating software at the end of the software development process to ensure compliance with software requirements.

VERIFICATION

Best practices in SQA identify the following activities as part of requirements verification:

- Evaluate requirements and relationships for correctness, consistency, completeness, accuracy, readability and testability.
- Assess how well the requirements document satisfies the high level requirements described in the project plan.
- Assess the criticality of requirements to identify key performance or critical areas of software.
- Produce a traceability matrix tracing all requirements back to high-level requirements in the project plan and forward to software design elements.

The first two activities are handled by the review cycles and DCS documents described in the previous chapter. The last two activities are handled by the development of a Requirements Traceability Matrix, as described in the following section.

REQUIREMENTS TRACEABILITY

The linkages between test cases, their parent design elements, and their parent requirements are maintained through a technique termed Requirements Traceability. In essence, it is necessary to be able to trace the linkage from a test case all the way through to a grandparent goal.

Requirements traceability is managed through the establishment and maintenance of a Requirements Traceability Matrix (RTM). The RTM is first established during the Requirements stage, and is one of the formal deliverables for that stage. The RTM is then updated during each of the subsequent stages, and reviewed to ensure that all elements have appropriate linkages (there are no "orphans").

An RTM can best be envisioned as an outline. As with any outline, parent elements are on the left and child elements move successively to the right. Each requirement has an associated criticality value; the values are used to determine which areas of the application are to be addressed first, and identify candidates for removal in the event of a reduction in the project budget.

- G1 – High-level requirement 1
 - R1(G1) – Requirement 1
 - D1(R1) - Design element 1
 - T1a(D1) - Test Case item 1a
 - T1b(D1) - Test Case item 1b
 - R3(G1) – Requirement 3
 - D4(R3) - Design element 4
 - T6c(D4) - Test Case item 6c
 - D7(R3) - Design element 7
 - T5c(R3) - Test Case item 5c
 - T6a(R3) - Test Case item 6a
- G2 – High-level requirement 2
 - R2(G2) – Requirement 2
 - D2(R2) - Design element 2...

The requirements traceability matrix is a formal deliverable for many stages of the SDLC. Refer to the RTM DCS for this project for details on required structure, content and review elements.

VALIDATION

In validation, the software is tested in a formal, structured manner intended to provide complete coverage of all requirements. This is accomplished through the execution of a series of test cases, each of which is traceable to parent requirements as described above. In essence, the test cases used to validate the software are in themselves verified.

The test cases are created during the development stage and incorporated into the project test plan, which is one of the deliverables of the development stage. The test cases include criteria for compliance with all requirements, performance at boundaries, and under stress conditions. Test cases are run against the software during the integration & test stage, where they may be modified before being incorporated into the final acceptance plan.

The acceptance plan is one of the deliverables of the integration & test stage, and is used as part of the formal acceptance process. Successful execution of the test

cases in the acceptance plan provides a formal basis for acceptance of the software by the customer.

CONCLUSION

The processes and techniques described in this SQA plan are specifically intended to insure the production of high-quality deliverables while minimizing the overhead generated by quality assurance activities. To accomplish this, this SQA plan relies heavily on two primary concepts:

- Deliverable Class Standards
- Targeted Reviews & Assessments

DELIVERABLE CLASS STANDARDS

Each deliverable review is based on a Deliverable Class Standard, which defines the scope, structure and content of the deliverable, as well as the items upon which each class of reviewer is to focus. The structure imposed by the DCS prevents reviewers from assessing topics outside their area of expertise. This is a critical concept; one of the most common problems found when implementing SQA is the generation of spurious issues by reviewers working outside their domain of expertise.

TARGETED REVIEWS & ASSESSMENTS

Project deliverables are reviewed at two levels: End-user (customer) and Technical (peer). Each reviewer reviews deliverables based on their area of expertise; this targeted approach allows reviewers to work with low effort and high quality, since they're evaluating topics that are directly related to their domains of expertise.

Quality assurance assessments are likewise defined under the DCS umbrella. This relieves the quality assurance reviewer from the responsibilities of determining the acceptability of deliverable content from an end-user or technical point of view. Instead, the quality assurance reviewer is restricted to verifying the presence or absence of specified items within, or associated with, the deliverable.