MEASURING QUALITY IN THE PRODUCTION OF WEB-BASED TRAINING: INSTRUCTIONAL DESIGN, PROCESS CONTROL, AND USER SATISFACTION

Dr. Susan Codone
Raytheon Technical Services Company
Pensacola, Florida

ABSTRACT

The field of software engineering has extensive literature regarding quality engineering and management. Procedures for process control, problem identification, defect analysis, and revision control are specified and standardized across the industry. Within the training realm, and specifically within production of web-based training, however, standardized quality engineering techniques are not commonly specified or followed. Although the production of WBT software follows many principles of software engineering, quality engineering for WBT production often does not follow the same guidelines. For the training industry, quality engineering and management is an area that needs more attention.

This paper will present an introduction to quality engineering principles relating directly to four attributes of web-based training production. First, quality measurement within the instructional design underlying web-based training will be explored – how to control and measure reliance on a standard design model and how to verify the integrity of WBT components, such as design strategies, objectives, instruction, assessment, and delivery mechanisms. Next, the paper will describe techniques to control quality during WBT production. In this section of the paper, the process of adapting software engineering quality control mechanisms to WBT production will be described and will include formative and summative courseware testing procedures, quality standard definitions, quality documentation and reporting, production quality management techniques, and courseware quality ratings. Following that, the costs of quality management will be explained as they relate to the costs of quality personnel and production time dedicated to quality engineering. Finally, the paper will conclude with a look at potential barriers to effective quality management, including staff acceptance, funding, the availability of a quality management process, and availability of quality management experience. Resources for quality management education and information will be presented, as will techniques for quality insertion into WBT design and production.

Again, although formal quality engineering principles for software engineering do not translate completely to the production of web-based training, there is potential for procedural commonality. It is hoped that this paper will contribute to the shared knowledge base of quality engineering procedures within the training industry.

ABOUT THE AUTHOR

Dr. Susan Codone is a project leader and instructional designer at Raytheon Interactive in Pensacola, Florida. Susan manages daily production of courseware for the Training Support Agency Requirements contract and serves as the company liaison with the Navy. Susan has a B.S. from the University of Montevallo, an M.B.A from the University of West Florida and Ph.D. in Instructional Design from the University of South Alabama.

MEASURING QUALITY IN THE PRODUCTION OF WEB-BASED TRAINING: INSTRUCTIONAL DESIGN, PROCESS CONTROL, AND USER SATISFACTION

Dr. Susan Codone
Raytheon Technical Services Company
Pensacola, Florida

INTRODUCTION

Rapid development in multimedia technology has given rise to new forms of online instruction via the Internet, organizational intranets, and other delivery mechanisms. The variation in instructional programming causes growing uncertainty about how to measure quality and how to know what comprises an effective multimedia training program. For both developers and users, there is an emerging need to define the attributes of quality and create a reliable set of evaluation standards (Gillis, 2000.)

This paper will present an introduction to quality engineering principles relating directly to four attributes of online, or web-based, training production. The paper will begin with an exploration of quality terminology and conceptual specifications found in the literature. Then, quality measurement within the instructional design underlying web-based training will be explored – how to control and measure reliance on a standard design model and how to verify the integrity of WBT components such as design strategies, objectives, instruction, assessment, and delivery mechanisms. Next, the paper will describe techniques to control quality during WBT production. In this section of the paper, the process of adapting software engineering quality control mechanisms to WBT production will be described and will include formative and summative courseware testing procedures, quality standard definitions, quality documentation and reporting, production quality management techniques, and courseware quality ratings. Following that, the costs of quality management will be explained as they relate to the costs of quality personnel and production time dedicated to quality engineering. Finally, the paper will conclude with a look at potential barriers to effective quality management, including staff acceptance, funding, and the availability of a quality management process and quality management experience.

QUALITY IN THE SOFTWARE ENGINEERING AND INSTRUCTIONAL DESIGN LITERATURE

Recent writings have given rise to more complete definitions of quality in the software engineering industry. Quality, which often hinges on acceptability, is defined in different ways. Vincent, Waters, & Sinclair (1988) distinguish between quality and acceptability, which may differ, they state, upon the user's understanding and the context of the situation. These authors cite Phillip Crosby's often-used definition of quality: *Quality must be defined as conformance to requirements, not as goodness.*

Further, they state that Crosby's definition emphasizes quality assurance functions in two major ways. First, developers must ensure that the requirements established for software correspond to user needs, and second, to ensure that the software product adheres to requirements, both in general functionality and specific quality characteristics. In other words, this definition of quality can be stated as fitness for use per customer specifications. Schulmeyer and McManus (1987) cite J.M. Juran's definition of quality assurance as the activity of providing to all concerned the evidence needed to establish confidence that the quality function is being performed adequately.

QUALITY WITHIN INSTRUCTIONAL SOFTWARE

Meanwhile, quality within instructional software is a much deeper construct that is more difficult to define. Is quality instruction simply instructional integrity, marked by the reliance of content on prespecified objectives? Does it consist of sound instructional strategies? Is it courseware that is pleasing to the eye, well designed aesthetically? Is quality determined in how well the instructional product satisfies requirements of a certain training medium? In many cases, quality evaluation of instructional software rests on the inclusion of certain elements that meet prespecified quality standards.

Brandon Hall, a noted e-learning researcher and consultant, hosts annual online training awards in which the following criteria are utilized:

Online Training Award Criteria	
Evaluation Criteria	**Explanation**
Content	Does the program include the right amount and quality of information?
Instructional Design	Is the course designed in such a way that users will actually learn?
Interactivity	Is the user engaged through the opportunity for input?
Navigation	Can users determine their own way through the program? Is there an exit option available? Is there a course map accessible? Is there an appropriate use of icons and/or clear labels so that users don't have to read excessive documentation to determine program options?
Motivational Components	Does the program engage the user through novelty, humor, game elements, testing, adventure, unique content, surprise elements, etc.?
Use of Media	Does the program appropriately and effectively employ graphics, animation, music, sound, video, etc.? Is the gratuitous use of these media avoided? Is the soundtrack really annoying?
Evaluation	Is there some type of evaluation, such as completion of a simulation or mastery of each section's content before proceeding to later sections, section quizzes, and final exam?
Aesthetics	Is the program attractive and appealing to the eye and ear? Does the structure of the screen add to the program?
Record-keeping	Are student performance data recorded, such as time to complete, question analyses, and final scores? Is the data forwarded to the course manager automatically?
Tone	Is the program designed for the audience? Does it avoid being condescending, trite, pedantic, etc.?

Table 1.0 Online Training Award Criteria http://www.brandon-hall.com

Building on Brandon-Hall's evaluation criteria, Gillis (2000) created Quality Standards for Evaluating Multimedia and Online Training. In this publication, she presents a four-stage quality evaluation process that includes the following steps:

1. Match courses to organizational needs
2. Conduct a content review
3. Conduct a usability review
4. Conduct an instructional design review

Stage One, Organizational Needs, reviews five areas: content, performance objectives, learners, course management needs, and technology. Stage Two, Content Review, examines the accuracy, breadth, depth of the content, and measures whether the presentation is clear, learning skills required match those of users, and the cultural/gender/racial appropriateness of the content. Stage Three, Usability Review, looks at the ease of use of installation, plug-ins, courseware speed, directions, interface design, appropriateness and timing of cueing and feedback, and the functionality of the interface elements (such as navigation and menus.) Stage Four, Instructional Design, examines the course objectives, modular structure of the content, engaging nature of the course, media, simulations, and the ability of the courseware to involve the user in higher-level thinking and interaction activities.

MEASURING QUALITY OF INSTRUCTIONAL DESIGN WITHIN WEB-BASED TRAINING

Building on the work of Gillis (2000), quality within the underlying instructional design in instructional software can be traced directly to the instructional integrity of a course. An examination of the course objectives, content scope and structure, instructional strategies, complementary and supporting media, practice and assessment, and review and conclusions typically yield usable information about whether the course is instructionally sound. David Merrill (1996) states that there are known instructional outcomes, and that instructionally sound courses will match those outcomes with appropriate instructional strategies. Without that match, Merrill states that instructional products do not teach and are thus not instructionally sound.

Good instructional design should be transparent to the user. In other words, an online course based on sound design principles should be built with instructional components seamlessly woven together to engage the user in learning while transferring intended content via prescribed instructional strategies. Because of this transparency, the responsibility to evaluate the strength of the instructional design inherent in any online multimedia course falls on design, production, and quality assurance staff rather than the user.

MEASURING QUALITY DURING WBT PRODUCTION

In contrast to evaluating the quality of instructional design, the procedures for evaluating process or procedural quality during production typify what is commonly known as Quality Assurance or Quality Control. More inspection than evaluation, these techniques for measuring quality during production vary widely within the instructional software industry. Within software engineering, however, SQA, or Software Quality Assurance, is a well-defined facet of the industry. There are set procedures for SQA followed by most major producers of software from which the instructional software industry can learn a great deal.

Schulmeyer & McManus (1988) cite J.M. Juran as defining quality assurance as the "…activity of providing to all concerned the evidence needed to establish confidence that the quality function is being performed adequately." (p. 5.) Meanwhile, IEEE defines quality assurance as the planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements. Furthermore, Juran defines quality control as the regulatory process through which actual quality performance is measured, compared with standards, and differences acted upon. Schulmeyer & McManus define quality assurance simply as independent evaluation to assure fitness for use of the total software product.

Chin-Kuie Cho, in his book Quality Programming: Developing and Testing Software with Statistical Quality Control (1987) states that statistical quality control is widely used in the manufacturing industry but not always utilized in the software industry. Cho states that quality control is defined as "…the act of directing, influencing, verifying, and correcting to ensure the conformance of a specific product to a design or specification." (p. 11.)

In preparation for this paper, production personnel at three companies were interviewed to determine the nature of their quality assurance activities. Questions were asked regarding their QA staff, methods, documentation, reporting, and organizational placement. Table 2.0 illustrates the interview findings and further validates the stated gap between production and quality management.

Comparison of QA Procedures at Three Online Training Vendors			
Software Quality Assurance Steps	**Company A:** **Intermediate-Size** **Producer of Customized Web-Based Training**	**Company B:** **Large Size** **Government Contractor; Producer of Customized Web-Based Training**	**Company C:** **Intermediate Size** **Producer of Customized Web-Based Training and Development Software**
QA Management Plan in place?	No	No	Yes
QA Staff in place?	No	Yes	Yes
Technical standards developed in advance?	No	Informally-developed standards in use	Formal standards in use
Configuration Management and version control utilized?	Yes	No	Yes
Standard QA review phases in place?	No	Yes	Yes
Problem Reporting & Corrective Action in place?	No	Yes	Yes
Place of QA staff in organization?	N/A	Independent of production	Independent of production

Table 2.0 Comparison of QA Procedures at Three Online Training Vendors

These differences among companies clearly indicate that the instructional software industry is in need of standardized QA procedures that guide production, reduce errors, and contribute to the development of quality software. The IEEE Standard for Software Quality Assurance lists several essential elements for quality assurance that can be considered basic best practices for software producers. Many software producers use this as an essential guide for quality management. Taken at face value, it can also be used to guide quality control activities for production of instructional software.

IEEE Standard for Software Quality Assurance

The IEEE Standard prescribes the comprehensive development of a software quality assurance plan that includes the following elements:

IEEE Standard for Software Quality Assurance	
Management	A plan should be in place that describes the quality organization, tasks, and responsibilities.
Documentation	Documentation should include: Software Requirements Specification Software Design Description Software Verification and Validation Plan Software Verification and Validation Report
Standards, practices, and conventions	Identify development standards and state how compliance will be measured.
Reviews and audits	Reviews include: Software Requirements Review Preliminary Design Review Critical Design Review Software Verification Review Functional Audit Physical Audit In-process Audit Management Review
Configuration management	Document and explain methods for identifying software product items, controlling and implementing changes, and recording change implementation status.
Problem Reporting and Corrective Action	Describe procedures used for reporting, tracking, and resolving software problems.
Tools, Techniques, and Methodologies	Identify tools that support QA activities.
Code, Media, and Supplier Control	Identify methods used to store controlled versions of identified software.
Media Control	Identify methods used to protect physical media from degradation.
Supplier Control	State the provisions for providing that vendor and subcontractor developed software meets requirements.
Records Collection, Maintenance, and Retention	Identify QA documentation to be retained, state methods and facilities to safeguard this documentation, and designate the retention period.

Table 3.0 IEEE Standard for Software Quality Assurance

While software engineering certainly has a design and configuration component, the necessity of performing full instructional design activities for the production of instructional software adds complexity not found within typical software engineering. In addition to the software requirements and design description, instructional software producers are often tasked with performing needs, task, and audience analysis; writing the instructional content; creating assessments, and interweaving instructional strategies and tactics into the course design. This adds additional steps to the IEEE software quality assurance standard. Additionally, instructional software is primarily a product of intensely creative screen design and unique multimedia delivery strategies, since instruction is carried and delivered by various screen elements of text, graphics, animation, video, narration, and the graphical user interface.

Many instructional designers and developers feel that standards and controls cannot be applied to something as creative and imaginative as developing instructional software. Without these controls, however, wide variations in product quality have resulted – gaps noticeable today by looking at the many different demos of instructional products available today on vendor websites. IEEE recommends that software production companies follow a set standard for conducting software quality assurance. The nature of instructional software requires a quality assurance standard not available in the industry today or included in the publications of various standards organizations responsible for overlaying standards onto instructional software production. In response, this paper presents a recommended standard for an instructional software quality assurance plan, presented below.

INSTRUCTIONAL SOFTWARE PRODUCTION QUALITY ASSURANCE PLAN: A SIX-STEP GUIDE TO PERFORMING QUALITY ANALYSIS OF INSTRUCTIONAL SOFTWARE

Step One: Estimate and Create an Instructional Software Quality Assurance (ISQA) Organizational Strategy

Schulmeyer and McManus (1988) state that the software quality program is the overall approach implemented to influence the level of quality actually achieved in a software product. The plan includes the:

- Establishment of requirements for the product quality
- Establishment and enforcement of procedures to develop and maintain the software
- Establishment and implementation of procedures to evaluate quality of the product and its associated documentation, processes, and activities.

The foundation of an ISQA plan is the organizational definition of product quality, including both functional and instructional criteria. Often this can be driven by customer expectations, but since most customers purchasing instructional software are not instructional designers and do not have a comprehensive understanding of instructional programming, the vendor can often guide these expectations by prescribing achievable instructional standards within the development project. The establishment of such an instructional software quality plan will be of immediate benefit to the vendor and customer. Vincent, Waters, and Sinclair (1988) state that an intelligent, properly conducted SQA program will significantly reduce the number of errors introduced into the software product and also ensure early detection of errors.

The ISQA plan should be written as an organizational document and considered as a procedural guideline by all involved in production. The plan should delineate the following:

1. A detailed ISQA organizational chart, listing responsible personnel and a communication chain
2. Clear assignment of responsibilities for quality management across the organization
3. Independence of the QA personnel from the core production team
4. Review schedules and reporting mechanisms
5. Specifications/criteria for reviews
6. Management structure

Schulmeyer & McManus state that such a software quality program covers not only technical aspects but also managerial. Procedural enforcement – a key component of ISQA – is a management activity, while actual development and enactment of these procedures is a technical activity. Development of the ISQA plan should occur at the beginning of a comprehensive instructional software development project and the plan itself should be published and disseminated to all development personnel.

Step Two: Create ISQA Documentation Plans

An ISQA plan should include a list and examples of relevant documentation for quality management activities. Relevant documents can include:

- **Requirements Specification** – describes each of the essential requirements (functions, design constraints, and course attributes) of the instructional software.
- **Design Description** – a design document detailing the design plan (i.e. the design blueprint)
- **Review and Audit Schedule –** a list and schedule of review phases covering the entire courseware production cycle, including both internal (production team) and external (customer) reviews.
- **Defect Reports and Revision Requests –** standard methods for reporting defects and requesting revisions, including report templates for use during QA activities within production.

- **Revision Verification and Validation Plan and Reporting Mechanisms –** a plan and process for reporting revision requests, verifying corrections, and reporting version maintenance.

In the documentation of online courseware, the use of online QA documentation in an intranet setting can be very efficient for an internal QA staff.

Step Three: Create Technical Standards and Conventions – a Design "Spec"

An ISQA plan should include specifications for design to serve as the basis for later quality control. In this way, quality standards are defined in the beginning, thereby narrowing the scope for later quality management activities. Quality assurance can easily experience scope creep when the definition of quality is not established in advance of development. When this occurs, quality analysis continues to creep upward toward a zero defect goal, when in reality, zero defects may not be the goal of either the customer or the software producer. Just as software engineers develop design specs for software in production, instructional designers and project managers should develop specifications for instructional software that serve as quality management anchors in the production process.

Consider technical specifications governing the following design components:

- Screen elements (text, graphics, animation, video)
- Interface elements (navigation, instructions, other resources like glossaries and maps)
- Graphic sizing
- Animation timing and other effects
- Standard instructional or directional text, especially on menus and assessment pages
- Button/navigational functionality
- Page/screen numbering
- Course communication with browsers, learning management systems, etc

The Software Productivity Consortium (1995) lists the following eleven software quality factors. These factors can be adapted to instructional software as well as part of a design specification, or list of expected standards:

Software Quality Factors		
Software Quality Factor	**Definition**	**Application to Instructional Software Production**
Correctness	Extent to which a program satisfies its expectations and fulfills the user's mission objectives	Content accuracy; ability of course content to convey information represented by instructional objectives
Efficiency	Amount of computing resources and code required by a program to support a function	Amount of code required to create instructional software program and enable functionality of all components
Flexibility	Effort required to modify an operational program	Ability to modify underlying code (i.e. HTML)
Integrity	Extent to which access to software or data by unauthorized persons can be controlled	Extent to which course content is kept sacrosanct from unauthorized editing by users
Interoperability	Effort required to couple one system to another	Ability of course to be used with other courses or learning management systems
Maintainability	Effort required to locate and fix a defect within the operational program	Same
Portability	Effort required to transfer a program from one hardware configuration to another	Same; especially true regarding performance of instructional software program from one browser to another
Reliability	Extent to which a program can perform its intended function with required precision	Extent of program to perform to expected levels from a functionality and instructional perspective; ability of course to teach content it purports to teach
Reusability	Extent to which a program can be used in other applications	Extent to which portions of the instructional software can be extracted and reused as content objects
Verifiability	Effort required to test a program to ensure that it performs its intended function	Same; but expand testing range to include functionality testing and instructional evaluation
Usability	Effort required for one to learn, operate, prepare input for, and interpret the output of a program	Same

Table 4.0 Software Productivity Consortium Quality Factors

Additionally, other specifications for instructional courseware can govern the design process, thus infusing standards throughout development and reducing errors at production. Table 5.0 lists standard evaluation criteria currently in use at a large vendor of instructional software production. These criteria relate to course functionality, instructional design, and screen design, quality of assessment items, and spelling and grammar.

Production Specifications
Functionality
Each navigation button functions as designed.
The topic is lockstepped at appropriate points in the topic.
Menu items change colors at appropriate points in the topic.
The pointer changes to a hand on all active paths or buttons.

Production Specifications
The items on the Main Menu "check off" upon completion of the module and not before.
The test scores properly and a score displays on the summary page.
Practice questions score appropriately.
Drag and drop items function appropriately.
Mouseovers function appropriately throughout the topic.
Instructional Design
The topic is presented in a logical sequence.
The lessons within a topic are consistent with one another in design, organization, and presentation.
The objectives are presented as provided in the government-furnished materials (GFM).
The topic supports the objectives presented.
Instructional frames match objectives.
Assessment items match objectives.
Content is chunked appropriately.
Screen Design
The text is well spaced and formatted for readability.
The frame is arranged in a balanced, eye-pleasing manner.
The graphics are crisp and legible.
There are no stray marks (a.k.a. pixel poop) on the graphics.
The call-outs, highlighting, etc., augment the presentation.
Text does not shift position from frame-to-frame.
Graphics do not shift position from frame-to-frame.
Assessment Items
Items are not so simple or obvious that they can be answered on the basis of common sense.
Instructions for answering questions are clear and simple.
Proper responses appear after selecting an answer to a question.
Fill in the blank practice questions: Ensure all blank lines are equal and distracters are lower case unless proper terms.
Multiple choice practice questions: Distracters - first letter only is capped, unless proper term.
Spelling & Grammar
Narrative and visual texts are presented concisely.
Topic is free of spelling and grammatical errors.
Narrative and visual texts are grammatically correct.
Topic flows and transitions smoothly between sentences and concepts.
New or unusual vocabulary and acronyms are explained when first introduced and are included in the Reference section of the product.

Table 5.0 Production Specifications for Online Training

Finally, in the Handbook of Multimedia Programming, Claypool and Riedl (1999) explain additional multimedia quality specifications. They write that multimedia applications must meet the performance needs of the users they support. In particular, they state multimedia performance is dependent upon delay (latency), jitter, and data loss. They tie these three attributes into a perceptual quality metric to quantitatively evaluate application performance from the user perspective. Claypool and Riedl state that application performance should strive to meet user-level requirements in addition to system-level requirements. Users want defined image resolutions and audio clarity, a smooth playout of audio and video, and an upper boundary on response time for interactive applications. Three quality components important to user perceptions include:

- Latency: the time it takes information to move from the server to the client to the user. Latency, also known as delay, decreases the effectiveness of applications by making them less like real-life interaction.
- Jitter: Variance in latency. Jitter can cause gaps in playout of a stream of narration or choppy appearance of animation or video. This often results from networks using packet switching and user workstations running multiple processes.
- Data Loss**:** Any data less than the amount determined by user requirements; may take many forms such as reduced bits of color, pixel groups, smaller images, dropped frames. Data loss may be done voluntarily by either the client or server to reduce load, jitter, or latency.

The importance of establishing technical standards in advance of development cannot be overestimated. Doing so provides QA personnel boundaries in which to work and shapes development of quality end products.

Step Four: Formalize and Document Review/Audit, Revision, and Reporting Procedures

An ISQA Plan ideally includes pre-specified courseware review phases in which comprehensive audits of the instructional software product are completed, revisions are noted, and full reports are made to production personnel. A distinction must be made between review and audit activities and review and audit phases. Review and audit activities are the mechanisms for quality control, which occur in larger review and audit phases, as explained below.

Review and audit activities include the following:

- Defect Identification
- Revision Prioritization
- Defect Trend Analysis
- Root Cause Analysis
- Correction and Recurrence Control

Meanwhile, actual review and audit phases are milestone development points at which comprehensive reviews are conducted, and the above activities take place as part of those reviews. Reviews and audits should occur at set stages in the production process, including the following:

1. **Design Plan Review** – evaluation of design plan, for instructional planning and course integrity

At the Design Plan review, typical evaluation factors include verification of the instructional plan and a decision regarding the economic and technical feasibility of the project itself. Documentation regarding process and production is updated and reviewed, and initial product requirements are verified and documented. At this phase, the technical and instructional specifications are presented in the design plan as factors to be tracked for adherence to the ISQA plan.

2. **Prototype/Storyboard Review** – for design and instructional quality

At the Prototype Storyboard Review, the in-process product is evaluated against immediate customer requirements for both instructional and functional performance. Additionally, basic quality factors such as text and other screen components are employed, such as grammar, spelling, presentation, functionality, and aesthetic appeal. At this stage, an internal prototype review is conducted immediately prior to an external, or customer, review.

3. **Comprehensive Functional Review** – full functionality review to ensure that course works correctly before release.

At this review, the ISQA staff combs the entire course to ensure the functionality of the review. They ensure that major instructional modules are properly linked and tested, ensure that all bugs and defects identified in the prototype review have been remedied, and ensure that the course meets the made at this point. Any revisions not made, either partially or completely, can be identified for repair before final delivery.

Illustration One identifies how review activities and design review phases are enmeshed to create a total

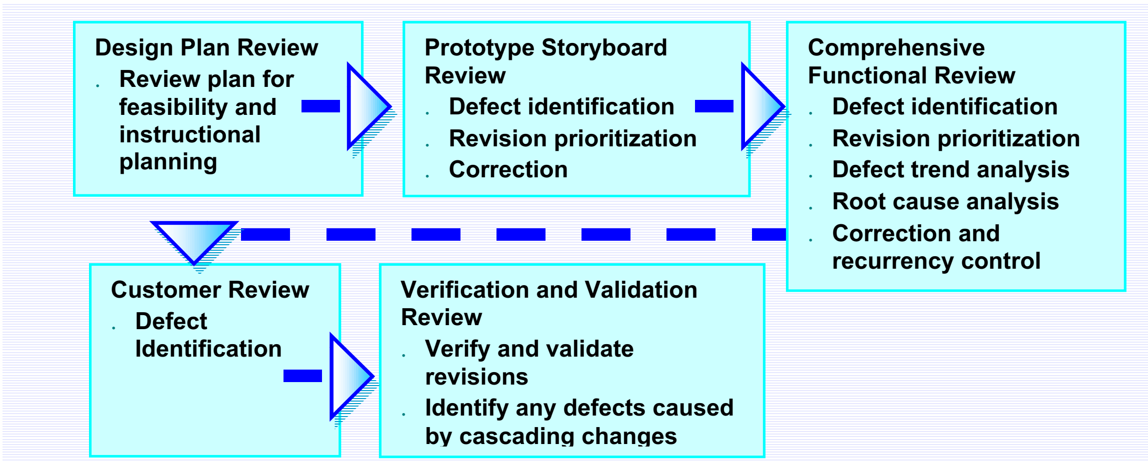

Illustration One: Design Review Phases

prespecified design standards (based upon customer requirements.)

4. **Customer Review** – of total product or prototype, depending on the production process in use.

At the Customer Review, typical expectations involve the identification of bugs or defect in the courseware and an overall evaluation of whether the in-process instructional product meets customer requirements. Typically, a defect report is compiled by the customer and sent back to the production team for action.

5. **Verification and Validation Review** – this review is restricted to verification of customer change requests. At this point, the instruction and content should be clean from error after the design, prototype, and customer reviews.

At the Verification and Validation Review, ISQA personnel should simply audit the course to ensure that requested revisions have been made. In some cases, the act of revisions creates new errors; these should be noted and forwarded to the design and production team for a decision regarding whether a change will be made. Revision-produced errors are sometimes known as "cascading changes", or changes that cause a cascading error effect, causing additional errors. A full course review should not be quality assurance function.

Vincent, Waters, and Sinclair (1988) differentiate between reviews and audits. Reviews, they state, are designed to check the development of software and make sure that the production process is employed effectively. Audits, in turn, are designed to check the state of the product in light of preestablished production criteria; they aim at identifying defects and problems in the courseware.

At each review phase, the review activities should occur in light of the production phase. Ideally, defect identification should occur at all phases, but trend and root cause analysis activities may taper off as the production process comes to a close. The idea behind trend and root cause analysis is to identify defect patterns that may reoccur based on faulty or defective production techniques; once the cause is identified through root cause analysis, the problem can be eliminated from further production through correction and recurrence control activities. The key in root cause analysis is to correct errors early in the process and then to change the process to prohibit error redundancies.

Step Five: Implement Process Control

Instructional Software Quality Assurance should not only consider product quality, but process quality as well. Quality assurance personnel should be integrated into the production process at an independent level so standard procedural steps can be implemented, monitored, and enforced. Ideally, process steps within production and the order of execution will be specified in advance. Project QA staff should be the process enforcers, ensuring that all production personnel follow the process without skipping steps or cutting corners (and thus opening the product to quality vulnerability.) Additionally, project QA staff should be involved in continuous process improvement based on data gathered through defect identification and trend/root cause analysis. Schulmeyer and McManus (1988) state the following:

> A product-driven process…. demonstrates that product quality can never be presumed and is a summation of every step that precedes the current state of the product. Just as a product cannot be separated from the development activities that bring it about, neither can the quality of the product be separated from the management goals, objectives, tools, techniques, and directives employed to achieve the quality goal. (p. 61.)

A typical instructional software production process includes the steps of design, development, programming, evaluation, and implementation. At times, the evaluation phase may be attached to each development step, and implementation may be followed by lifecycle management. An effective ISQA plan will afford QA personnel the authority to oversee process conformance by production personnel, thus ensuring that production steps are not skipped, thus inadvertently exposing the product to quality vulnerabilities.

Step Six: Control Records and Data Collection Activities

An effective quality assurance program will produce meaningful documentation either on paper or online regarding course reviews, problem identification, and revision requests. An organizational ISQA will specify how this documentation should be retained and managed. It is recommended that a configuration management system for QA files be developed and followed, and that all reports made available to project leadership as well as production personnel so that trend and root cause analysis activities may occur on a regular basis.

Summary

An ISQA plan clearly benefits the vendor organization, the customer, and the end-user by virtue of systematically planning for quality and reducing errors during production.

COSTS OF QUALITY ASSURANCE

Quality costs are typically measured in the actual cost of performing quality assurance activities rather than the very real cost of *not doing* quality assurance; i.e., in rework resulting from persistent errors and product defects. Schulmeyer & McManus (1987) state the following:

If the (QA) task costs one amount, but the result of performing that task saves another amount, the real cost is the *difference* between the two, not the initial cost of the activity. Schulmeyer and McManus further provide the following steps to estimate quality costs.

1) List all major QA activities.
2) For each activity, list inputs and outputs.
3) For each activity, determine customer and supplier requirements and project parameters.
4) For each activity, analyze parameters, determine level of effort, estimate cost of effort, and divide cost into meaningful categories.

A good rule-of-thumb is to estimate a certain amount of QA hours for each hour of online courseware, depending on its complexity and level of instruction. Then, break those hours down into the individual review phases to create a QA labor budget. If actual QA expenditures by hours and dollars are tracked, and if QA actuals exceed hours budgeted, then process control, defect and trend analysis, and root cause analysis can be employed to systematically reduce QA time and improve product quality.

BARRIERS TO EFFECTIVE QUALITY ASSURANCE

There are three primary barriers to effective quality assurance:

1. An organizational structure in which QA is not an independent entity
2. Resistance of development and production staff to QA activities
3. Lack of funding for QA activities

In all three cases, effective planning for QA at the advent of a development project can alleviate these barriers. When a project is estimated, QA time must be estimated as part of the level of effort required for successful project completion. Likewise, project staff must be educated about the value and proper place of QA, and the organizational structure must support QA's role as an independent auditor of courseware.

An effective ISQA plan will promote QA's independence, plan for developer buy-in of QA activities, and allow for proper funding support for QA. For production of quality products, however, the organization has to maintain a commitment to quality throughout the production process, even when time is short, deadlines are near, and funding is decreasing at a rapid rate.

SUMMARY

Just as production management and efficiency is important in the design and development of online training, quality management is crucial to ensuring both instructional and functional integrity of the final product. This paper has explored four areas in which quality can be infused into the design of online courseware.

The instructional design underlying online training must be evaluated for quality – especially controlling and measuring reliance on a standard design model and verifying the integrity of WBT components, such as design strategies, objectives, instruction, assessment, and delivery mechanisms. Measuring quality during production is also essential. This paper presented a six-step standard for controlling quality during WBT production. In this section of the paper, the process of adapting software engineering quality control mechanisms to WBT production were described and included courseware test and review procedures, quality standard definitions, quality documentation and reporting, production quality management techniques, and courseware quality ratings.

Finally, the costs of quality management were explained as the actual difference between the cost of quality and the cost of not performing quality assurance activities (i.e., rework). Three barriers to effective quality management were also explored, including staff acceptance via an organizational structure conducive to quality, funding for quality activities, and the availability of a quality management process and experience.

REFERENCES

Cho, C.K. (1987). Quality programming: development and testing software with statistical quality control. New York: John Wiley & Sons.

Claypool, M., & Riedl, J. (1999). End-to-end quality in multimedia applications. In B. Furht (Ed.) Handbook of multimedia computing. Boca Raton: CRC Press.

Driscoll, M. (1998). Web-based training : using technology to design adult learning experiences. San Francisco : Jossey-Bass/Pfeiffer

Furht, B. (Ed.) (1999). Handbook of multimedia computing. Boca Raton: CRC Press.

Gagne, R.M., Briggs, L.J., & Wager, W.W. (1992). Principles of instructional design. Fort Worth, TX: Harcourt Brace Jovanovich.

Gillis, L. (2000). Quality standards for evaluating multimedia and online training: everything you need to know to rate online courseware. Toronto: McGraw-Hill Ryerson.

Institute for Higher Education Policy. (2000). Quality on the line: benchmarks for success in internet-based distance education. Washington, DC.

Lee, W.W., & Owens, D.L. (2000). Multimedia-based instructional design : computer-based training, Web-based training, distance broadcast training. San Francisco : Jossey-Bass/Pfeiffer

Merrill, M.D. (1996.) Instructional strategies that teach. CBT Solutions, Nov/Dec, 1-11.

Schulmeyer, G.G., & McManus, I. (Ed.) (1987). Handbook of Software Quality Assurance. New York: Van Nostrand Reinhold.

Software Productivity Consortium. (1995). The software measurement guidebook. Boston: International Thomson Computer Press.

Vincent, J., Waters, A., & Sinclair, J. (1988). Software quality assurance: practice and implementation. Englewood Cliffs, NJ: Prentice-Hall.