SANDIA NATIONAL LABORATORIES

Electronic Unreviewed Safety Question (eUSQ) System Lessons Learned

Stephen A Coffing, Sandia National Laboratories; sacoffi@sandia.gov
Jeffrey W Marr, HukariAscendent Inc.; jwmarr@sandia.gov

Electronic Unreviewed Safety Question (eUSQ) System Lessons Learned

Introduction

The Electronic Unreviewed Safety Question (eUSQ) System is an electronic tool that was developed to simplify the existing USQ paper-based system and reduce the potential for errors by allowing personnel to electronically follow the Sandia Unreviewed Safety Question (USQ) Screening Checklist and Unreviewed Safety Question Determination (USQD) Worksheet processes and generate the required forms. This system was designed to automate workflow control while providing the proper access controls (i.e., access only to qualified preparers, reviewers, approvers, and specially designated personnel such as the local site office safety basis counterparts). The system was also designed to allow users to track all USQ screens and USQDs in process in one place improving configuration management of the process. It provides for creation, editing, revision, copying, and cancellation USQ documents. For completed documents, eUSQ automatically transmits a PDF record for 75-year document retention.

As with the conversion of any paper-based system to an electronic system, several unforeseen problems were encountered that had to be solved in order to meet the requirements established for this system. This paper highlights the most important issues and provides points of consideration for those who are planning to automate their USQ or similar processes. The areas discussed in this paper are software documentation, developing an automated process, software platform, consideration for administrative features, effect of code changes on software processes, and software usability.

Software Documentation

The initial documentation prepared for development of eUSQ consisted of a requirements document. After research to learn what had been performed on larger software projects and corporate guidance for implementing a graded approach to software quality, several other documents were added. A Software Design Document was created to sufficiently describe the eUSQ system such that another application developer with the proper skills would be able to support the maintenance and retirement phases of the software product lifecycle. A test plan was prepared to identify the tests necessary to verify that the requirements from the requirements documents were achieved. A software verification and validation plan was created to document the level of verification and validation required for eUSQ based on application of the graded approach to software quality. A practice level and safety software determination was performed and documented to identify the magnitude of risk associated with a potential failure of the eUSQ software. A risk mitigation plan was developed to identify potential problems before they occur such that risk-handling activities could be planned and invoked as needed (e.g., how to temporarily use a paper-based system in the event of a serious software or server failure). Process diagrams were created to show how the software implements the paper-based process for creating, reviewing, and approving USQ documentation. In addition, an application screen and navigation diagram was created to document flow from a software perspective, which was useful to ensure software consistency (e.g., similar application screens had same navigation options) and enhance usability. Finally, a software lifecycle process was documented that provides information on the implementation and tools used in the software product lifecycle and version updates for the eUSQ

system. The format and content for a majority of this documentation was taken from IEEE software standards. Collectively, these documents are expected to result in cost savings for maintaining the software and to minimize the potential for human error and software failure.

Developing an Automated Process

The paper-based process for performing USQ evaluations was originally believed to be very simple to translate into an electronic process, but was learned during development to be much more complicated. A case-in-point was the use of two forms for the paper-based system. The first form, a screening checklist, was completed to determine the applicability of the USQ process. The second form, the USQ Determination (USQD) Worksheet, documents a detailed description, relevant information used for the evaluation, and the results of the evaluation (i.e., the answers to the seven questions). Two separate database records are generated; one for the screening checklist, one for the USQD, which are listed separately on the listing page. When performing searches, some of the search parameters entered are relevant to the screening checklist, some relevant to the USQD, and some relevant to both. When coupled with multiple Boolean conditions (AND/OR), search routines became quite complicated as information from both records need to be searched and tied together. This also resulted in an occasional listing of erroneous information (e.g., accompanying screening checklist is listed when the search was performed on a USQD approval).

Another complication of using two forms arose based on how steps had been performed in the paper-based process. While the steps in preparing and reviewing a screening checklist and in preparing, reviewing, and approving a USQD might be perceived to only be performed in serial fashion, facility preparers would often submit the screening checklist and USQD together for review (see Figure 1). Rules had to be carefully defined for eUSQ. Should a USQD database record always be created, even if not used? And if not, what should be the point in the screening checklist process when the USQD record is created? At what point should the USQD process be halted until the screening checklist is completed?

The dynamics of the paper-based decision process had to be limited in the electronic process. For Sandia, the best solution was to create the USQD record when the screening checklist is signed by the preparer (it could be deleted afterward if the preparer alters the input to result in the conclusion that a USQD was not required), but not allow the USQD to be signed by the reviewer until the screening checklist review was completed.

Given the complexity of this approach, the next revision of the USQ procedure will likely involve one combined form. This will allow for a single database record (instead of two connected records) and one review process (instead of two intertwined review processes). This is expected to both streamline the process and the eUSQ code.

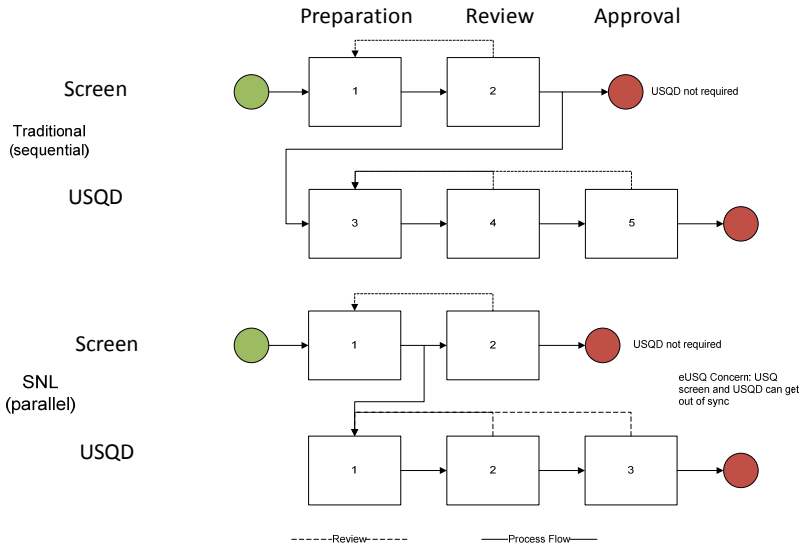

Figure 1. Traditional Versus Actual Process for Processing USQ Evaluations

Software Platform: Web Server Application vs. PC Client Application

The eUSQ system was developed as a corporate web server application using ASPX and VB.NET code that is accessed via an internet browser. This system uses a SQL database stored on a separate server. The rich-formatted text that users input into the browser application pages is translated into HTML and stored in the SQL database.

This approach resulted in several difficulties. HTML has limitations on how text is formatted making creation of items (e.g., tables) that would be easy to create in a PC application such as Microsoft Word a more challenging task. HTML uses tags to mark formatting such as italics and bold, and when tags get nested or incorrectly ordered, the resulting text may not match the format the user desires. The two biggest problems occasionally encountered were that selecting text and formatting with the menu controls did not format the text as expected and the format of the draft PDF did not match the appearance on the computer screen (e.g., table and text would be truncated at the margin).

Another difficulty is the level of control the application developers have over the environment in which the software is running. Application developers typically have limited control over a corporate web server environment compared to stand-alone PCs. Corporate servers are designed to allow a variety of software to perform and have a higher level of security applied to server resources. An example of this impact occurred in trying to determine how to configure the eUSQ code and a PDF conversion utility running anonymously on the server such that when the utility is invoked by the code, it has access to the USQ data to create a PDF file and then have the proper access to store the PDF file in the desired location on the server. This issue became further complicated when trying to migrate the code to a new

64-bit server, as changes to the server configuration and protocols effected the way in which server processes have access to server resources.

The lesson learned for this situation would be to identify the available platforms prior to software development and attempt to estimate cost and resources for development on each platform. For eUSQ, the initial decision was primarily based on the skill sets of the application developers (as is often done). The application developers were very familiar with the programming languages used for web-based code and less familiar with code such as JAVA, that would be used for the development on a stand-alone PC client. However, many other factors should be considered including: 1) software usability (e.g., control over content such as text formatting), 2) control or lack thereof over the application environment, 3) time and resources for development and maintenance throughout the software lifecycle, and 4) ease of installation of use.

This consideration may involve a decision regarding the level to which usability is compromised to reach a reasonable level of cost and resources, similar to a cost/benefit analysis. Since eUSQ has already been developed on a corporate server, if the above analysis were to result in a higher level of usability to convert from a corporate server application to a PC client application, the cost might not justify the gains. A better use of resources in this case might be to develop techniques to improve the user's ability to achieve their goals (e.g., use MS Word to create the table then copy/paste into eUSQ).

Ability to Access Administrative Features

The eUSQ system was originally designed to only allow the application developers the ability to modify information stored in the SQL database tables via external software such as the SQL Server Management Software. Data specific to a USQ evaluation would not be modified, however other data such as the list of qualified users and the list of safety basis documentation for each nuclear facility would require updates on an as-demanded basis. While initial development of the eUSQ system required the application developers full time, maintenance of eUSQ may not require this level of effort. Therefore the potential exists where an update to this information is required to occur when the application developers are unavailable. The solution to this potential issue was to develop an administrative module to eUSQ to allow specially-designated eUSQ administrators access to update this information. Administrators will be able to add or delete users and update their qualifications (i.e., user roles within eUSQ).

In addition to the ability to maintain this information, two report features are in development. The first is to report the number of USQDs prepared, reviewed, or approved over a specified period by user for purposes of USQ evaluator requalification. The second report is a metrics report on quantity, times, and results of USQ evaluations over a specified time.

Effect of Code Changes

As mentioned earlier, two forms are used in the Sandia USQ process; a screening checklist form and a USQD form. The screening checklist consists of three web application pages, each representing a group of questions geared toward different conclusions: 1) the changes are screened out and do not require a USQD; 2) a decision has already been made to submit the changes to DOE for approval, and 3) changes which trigger an entry condition and require a USQD. A question answered 'Yes' in either of the first two sections completes the USQ process at that point. A question answered 'Yes' in the last group of questions continues the process into the USQD.

The software flow was originally designed to be sequential assuming that no information had been entered beyond that point. For example, when navigating from the first group of questions to the second group, the code assumed that only the first group of questions had been answered, and the act of navigating to the second group of questions results in the code evaluating the results of the screening checklist based only on the answers to the first group of questions. This assumption was consistent with the directions provided on the paper form.

At a later point, features to copy or revise a USQ evaluation were added. The eUSQ system would copy the information to the new/revised evaluation and the user would update it based on the information relevant to their new evaluation. Addition of these new features led to a requirement clarification that the draft screening checklist identifies results of the screening checklist based on the questions answered. The code was modified to provide this information using the same assumption.

An end user discovered cases where the screening checklist results on the draft PDF were not consistent with how the questions were answered. In one particular case, no information had been changed from the copied screening checklist, yet the results were different. In some cases the draft results were correct, while the results were incorrect for other cases. For the final screening checklist, the results were always correct.

Tests were performed, re-creating the conditions the end user provided. The problem was discovered to be that the software flow changed from what was originally assumed. When the user updates the information in the copied or revised evaluation, there already may be information in subsequent groups of questions as was the case in the above example. When the user navigated from the first group to the second group, the screening checklist results were updated ignoring a 'Yes' answer in the third group of questions.

The lesson learned is a recommendation to document software flow and assumptions that are made during development. When changes are made, the software flow and assumptions can be reviewed to determine if the changes affect the flow or assumptions. Then appropriate changes to the code can be made. This would be particularly useful if the application developer maintaining the code had no involvement in the initial development.

Software Usability

Feedback and improvement have always been encouraged for the eUSQ System to improve usability and enhance the USQ process. Four improvements worth discussion are currently in development: 1) export feature for facility or search result listing of USQ evaluations to an Excel spreadsheet; 2) batch download feature to download evaluations from a facility or search results listing, 3) provide ability to identify whether an evaluation would result in a DSA page change to support the annual safety basis update process (a data item which was not documented on the official USQ evaluation forms), and 4) provide ability to access the comment field at a later date.

The first two features are useful in creating documentation associated with the annual update of safety basis documentation and for performing assessments of the USQ process. Currently the user has to take screen snapshots or manually write down the list; with the new feature, the user will click on an Excel icon next to the listing table and the eUSQ system will prompt the user to open the spreadsheet on his/her PC, where it can then be manipulated on the user's PC. The batch download feature will save the user a significant amount of time by downloading a group of selected files to a folder on the user's PC.

Items 3 and 4 are in regard to access of information after the USQ evaluation has been completed. The initial requirements for eUSQ were to only retain PDF documentation of the completed record and to maintain the data in the database for use when copying or revising the USQ evaluation. Beyond this, little thought was originally provided to define additional information (that beyond the information directly used in the USQ process or required by procedure) that would be useful. Upon further discussion with the end users, the scope of the third item was broadened to include the ability to change the answer whether a DSA page change was required to support subsequent review by nuclear facility safety basis personnel for evaluations prepared by qualified operations personnel who don't have the same level of familiarity with the safety basis documents as the safety basis personnel. The lessons learned is to consider, during development, what features beyond those required by procedure that would be useful and to solicit feedback from end users on features they believe would improve productivity or usability.

Conclusions

Several lessons learned were discovered during development and initial use of the eUSQ system. These lessons illustrate both the difficulty to preclude problems that the eUSQ user will encounter, but also illustrate the system that is in place to identify and improve the system to enhance usability. Also, the lessons learned will provide additional thought for those who will be implementing their USQ or similar process in the future.