

Computer Assisted Interview Testing Tool (CTT) - a review of new features and how the tool has improved the testing process.

Russell Stark and Rebecca Gatward – Survey Research Center, University of Michigan

1. Introduction

The CAI Testing Tool (CTT) is an application developed by the University of Michigan's Survey Research Center to manage and facilitate efficient testing of Blaise questionnaires.

CTT is a comprehensive tool designed to be used by all involved (programmers, project managers, and clients) in questionnaire testing and development from beginning to end. As Dascola et al. explain in their paper, presented at the 2007 International Blaise Users Conference, CTT was designed to improve the quality of CAI instruments through standardized testing procedures, reduce the cost of testing, and increase access to information about the CAI development process through preset and adhoc reports.

The tool has now been in use for around three years and a new version of CTT has recently been developed. This paper will outline recent updates to the testing tool, which include the capability to test Blaise IS questionnaires, will examine how the tool has changed the testing and development process, and will attempt to quantify how it has improved the quality of CAI instruments and reduced testing costs.

2. Background

Development of the CAI Testing Tool began 'in-house' in 2006 and the first production version was being used by project teams in 2007. The tool was developed with the aim of standardizing testing, increasing the overall quality of CAI through standardized testing procedures, reducing testing costs and increasing access to information about CAI development through standard and ad-hoc performance metric reports.

3. Key components of the CAI Testing Tool

The CTT consists of six main components, these are summarized below, a more detailed description of the tool is provided in Dascola et al (2007).

3.1 Automated testing tool

The CTT is a fully automated tool used to maintain a bug log for Blaise applications. This is a program that 'follows' the tester while they run through a Blaise application. The tool keeps a keystroke log, can capture screen shots, and automates the bug log and bug log item prioritization. Pre-loads (scenario information) can be specified to ensure all aspects of Blaise instruments are tested.

3.2 Pre-load builder

The CAI Testing Tool also includes a pre-load builder, which is used to enter or load any data required to test the instrument. The pre-load builder allows us to create a new or update a pre-load for a revised instrument, or delete an existing pre-load via an administrative data entry screen.

3.3 Random case generator

The random case generator feature tests questionnaire routing using test cases that have been created using random data. Using the output the programmer can determine if all the questions in the instrument have a response. The programmer can then investigate if the blank fields are due to a routing or programming error. The Random Case Generator includes some 'intelligence,' with preloaded

information guiding what variables are not randomly generated, to allow for focused generation of case data and testing of targeted paths. This feature helps ensure every piece of the questionnaire has been tested. The module produces an Excel file called frequency.csv, which lists the questions alongside the number of times each question has been answered.

3.4 Management

The ‘manage problems’ screen works as a functional dashboard to deal with issues. Its general purpose is to review and manage the bugs or enhancements reported during testing. The bottom half of the screen is primarily used by the Blaise programmer. Here they can view all information about the issue and update information about each problem for example, the problem status, open a screenshot or audit trail, and enter the date when a problem is fixed. This screen displays all problems, regardless of status. The addition of a capability to sort, move and filter columns is a great improvement in the newest version. Blaise and BlaiseIS problems are all listed on the ‘manage problems’ screen in the same way.

3.5 Reporting

CTT has a built-in reporting function providing the tester, coordinator and programmer the ability to generate specific reports. This helps each level of the testing staff to focus their efforts effectively. The ready to test report is designed primarily for testers. This report shows the tester which problems have been repaired, their locations and other relevant data. The tester can use this report as a guide to what to re-test and to confirm that problems have been solved. The ‘Not Fixed’ and ‘Not Fixed High Priority’ reports are designed for the Blaise programmer. This allows them to one-click a report of what they currently have to work on. We also display a number of summary reports that provide information on test performance metrics (e.g. Problems found, problems fixed problems remaining etc.). These are used by managers and testing coordinators to monitor testing progress.

3.6 Administration

The Administrative tasks tab in CTT contains a variety of upper-level functions. Managers or coordinators can add projects and set-up files for testing, assign testers to projects, attach specific priorities to different problem types, import master preloads and assign access rights. These administrative tasks are grouped in the Admin tab and then are shown as large icons on the ‘ribbon’.

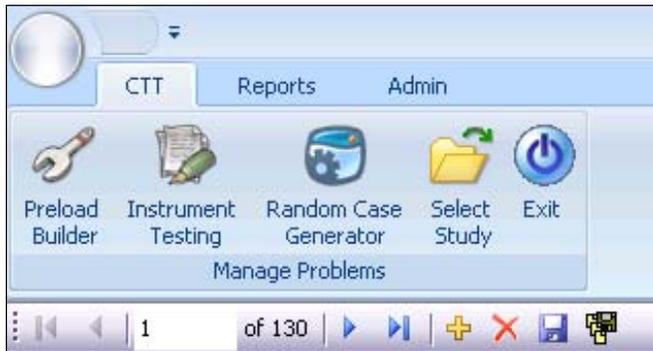
4. Recent developments to the CAI Testing Tool

Since 2007, we have continued to develop the CTT. The enhancements made are in response to feedback from users and the need to keep up to date with new developments in Blaise. The key developments are described below.

4.1 Updated user interface

Since the first version of the CTT was developed, we have updated the overall look of CTT using the now familiar ‘Ribbon’ interface. It allows for a smoother user interface, using large icons for navigation (figure 1) instead of dropdown menus from the title bar. We have also introduced tabs, allowing similar tasks to be grouped and separated. Users have reported that they find the new user interface and its updated look easy to navigate.

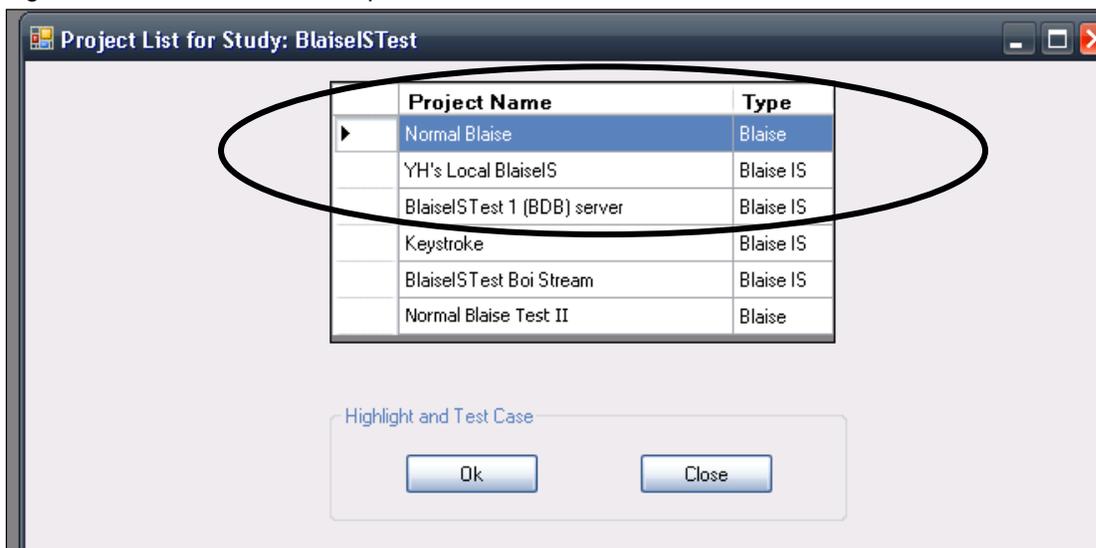
Figure 1. Large icons for navigation



4.2 Blaise IS

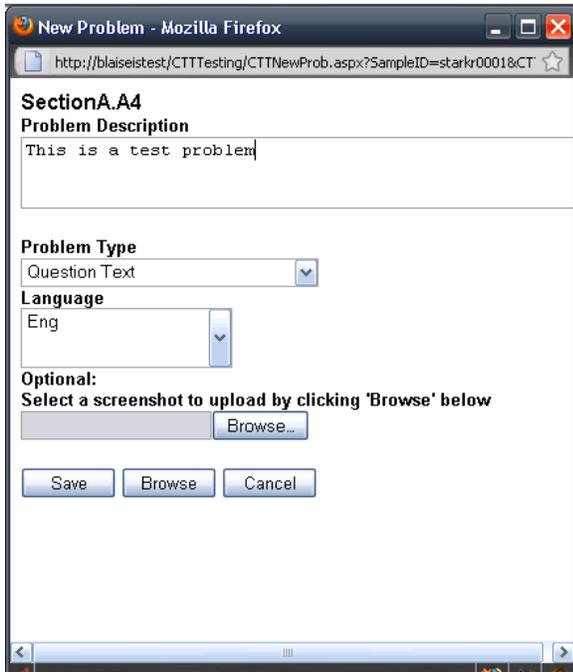
A recent addition to the functionality of CTT, is the ability to test both BlaiseIS web-based instruments and 'standard' Blaise instruments simultaneously (figure 2). The Blaise IS component has been designed to follow the functionality of the general Blaise testing tool as close as possible.

Figure 2. Blaise and BlaiseIS questionnaires are accessible from the CTT



The report problems screen, which users see most often, is nearly identical in both versions, as is the short-cut key (F8) used to call this screen (figure 3). This reduces the need for any retraining for users who were trained on the original system or the need for separate trainings for general Blaise and BlaiseIS testing.

Figure 4. New problem screen in BlaiseIS testing



The BlaiseIS testing tool has been designed to work in all major web browsers (IE, Mozilla Firefox, Chrome, and Safari). After a user has selected an instrument to test, they are prompted to select whichever browser they prefer (and have installed) (figure 4).

Figure 4. Choosing a Web browser for BlaiseIS testing

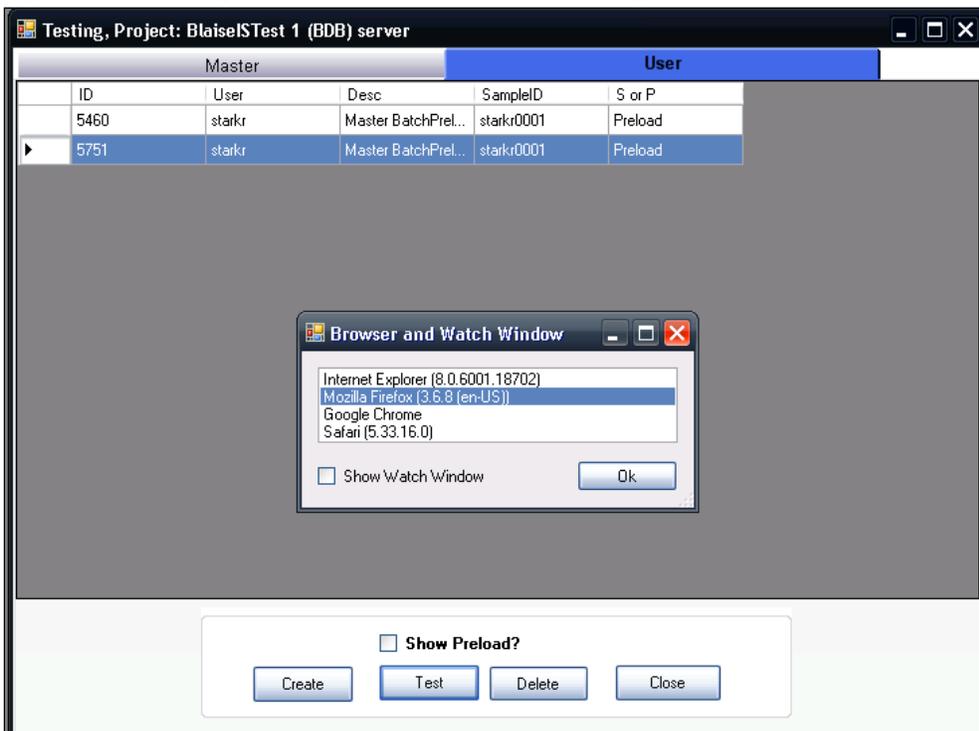
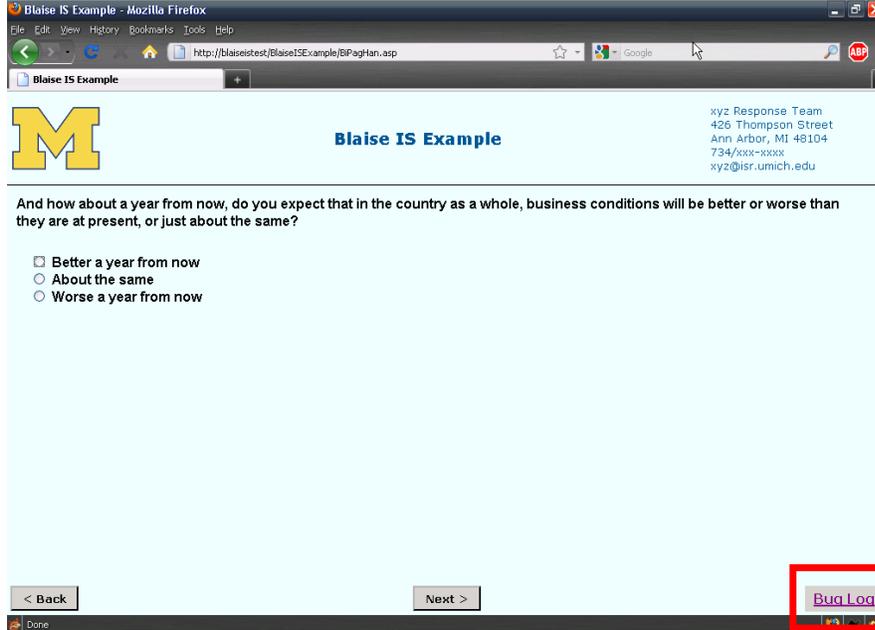


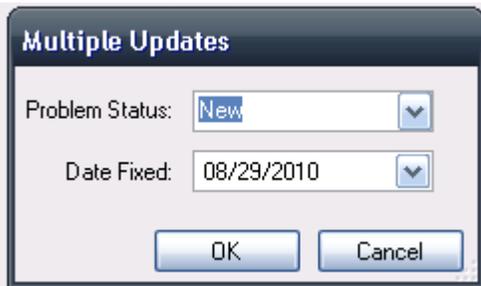
Figure 5. Example of BlaiseIS instrument, with CTT



4.3 Manage problems

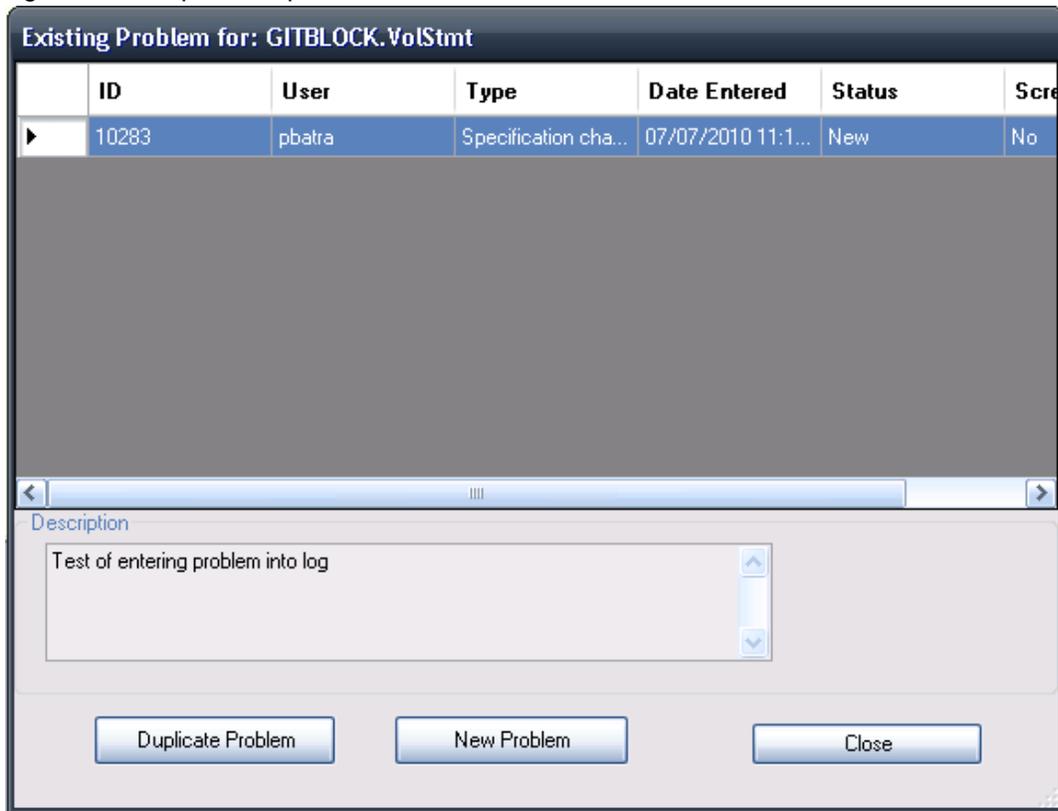
Many updates have been made to the Manage Problems screen. A wish list item from the first version of CTT was the ability to sort and filter information based on user requirements. This option is now functional, even allowing the user to resort each column to their needs. This main screen also now features a functional screenshot button. This permits the user to open a screen capture of the problem in question. Feedbacks from CAI programmers lead to the development of the Multiple Updates button (figure 6). This allows the programmer to update the statuses of multiple issues without having to save each time they updated a line.

Figure 6. Multiple Updates screen



User's access levels now determine what each level of user sees on the Manage Problems screen. Access level 1 (tester) cannot see the problems listed by default. This was put in to place to help keep the interface clean and uncluttered for novice testers. Access levels 2 (Testing Coordinator) and 3 (Programmer) allow users to perform Administrative tasks as well as view all issues listed.

Figure 7. Example of Duplicate Problem screen



5. How the CAI Testing Tool has changed the testing process

The CAI Testing Tool has been used extensively within the Survey Research Center since the first production version was completed in 2007. In order to understand how the CTT has changed the testing process, it would be useful to briefly describe how testing was carried out before CTT was developed. Pre-CTT, comments were recorded on one or multiple Excel spreadsheets. This meant that a lot of time was spent reconciling logs and trying to determine what the tester was trying to convey. Too much time was spent working on 'bug logs' and not enough on testing or developing the instrument. CTT integrates all of the functions for Blaise testing we need into one software package, and automatically gathers the testing information and outputs it to formats which can be used by all.

In this section of the paper we describe how CTT has improved the testing experience from each of the three main users perspective: tester, programmer and manager.

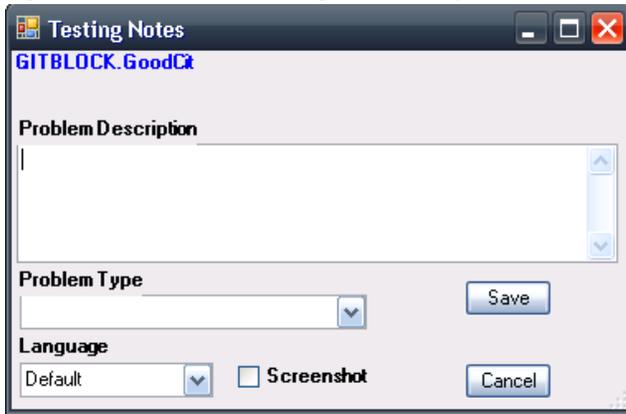
5.1 Tester

5.1.1 Direct access to the CTT

The main advantage of using the CTT for the tester is the ability to access the tool and record comments directly from the Blaise questionnaire (figure 8). Using a short-cut key, they can record comments with no need to jump between applications. The testers comments are saved directly into one master 'bug' log ,

the tool ‘stamps’ the comment with the field name/path, records the testers username and attaches any additional data – for example, a screen shot to the comment.

Figure 8. CTT Blaise testing notes entry screen



5.1.2 Existing problem

The ‘existing problem’ feature, eliminates duplication of effort – the tester is notified if a comment has already been attached to a field – they can then check if the existing check is the same as the one they have (figure 9a and 9b). When appropriate testers can then click a ‘duplicate problem’ button while testing inside Blaise, affirming they have recognized the problem also, but do not need to make separate note for it.

Figure 9a. Duplicate problem screen (Blaise)

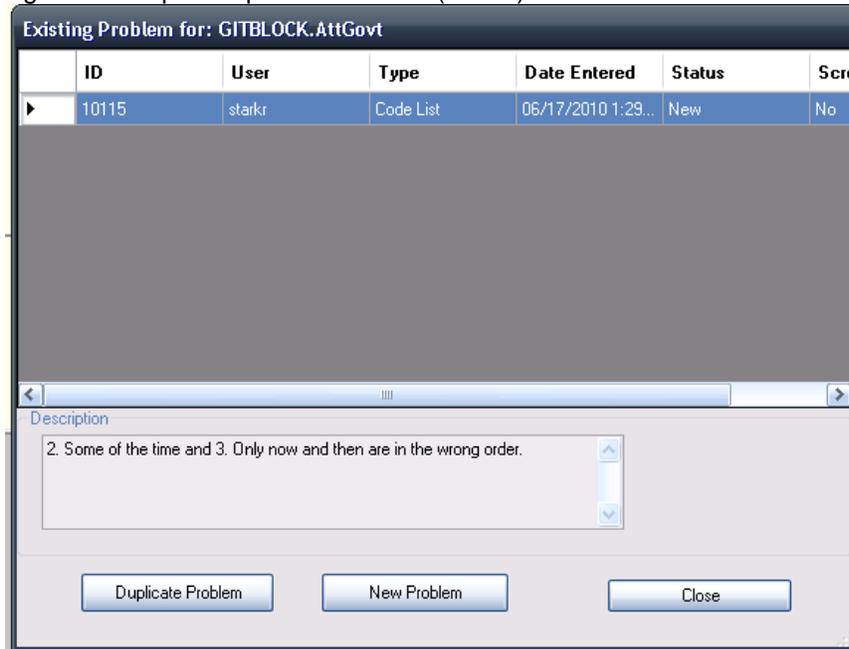
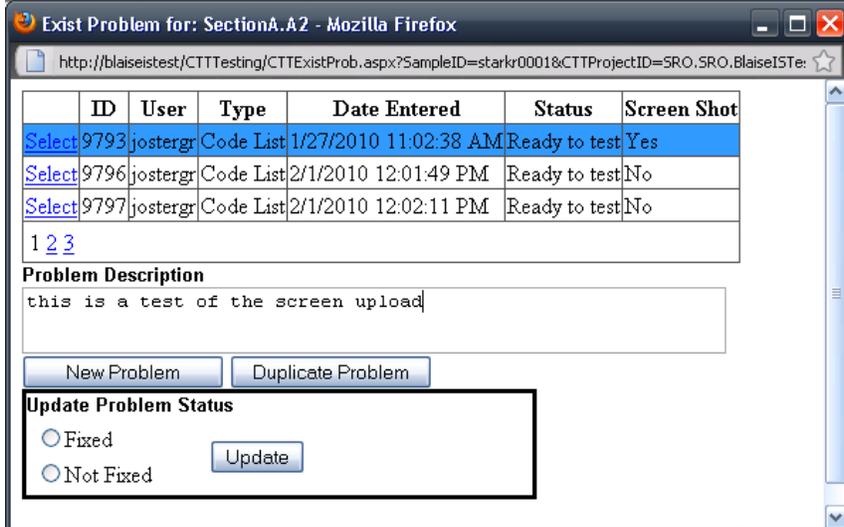


Figure 9b. Duplicate problems screen (Blaise IS)



5.1.3 Screen shots

Testers have found the screen shot option is particularly useful when commenting on screen layouts because it reduces the amount of text they are required to enter to describe the problem, in addition the programmer is able to see how the screen layout appears without entering the Data Entry Program (DEP) themselves.

5.1.4 Central creation of testing cases

The ability to create ‘master scenario’ testing cases has also saved time. Those with particular access levels can create ‘master’ cases and make them available to more than one tester. The tester can then select and save the cases to their own ‘testing area’. This feature reduces time spent specifying scenarios and for testers recreating them, it also helps ensure all testers are using the correct testing parameters.

In further benefit of using a shared pool of master cases is that the cases can be prefilled to particular points in the questionnaire, this allows the tester to concentrate on elements of the questionnaire they have been assigned to test rather than wasting time getting to the point of testing. This is especially useful when testing effort has been divided between sections of the questionnaire or when clients wish to concentrate their testing on the questions they are funding.

5.1.5 Remote testing

One of the aims of the CTT was to allow remote testing. This facility has proved to be useful with not only off-site testers, such as field interviewers, but also with clients.

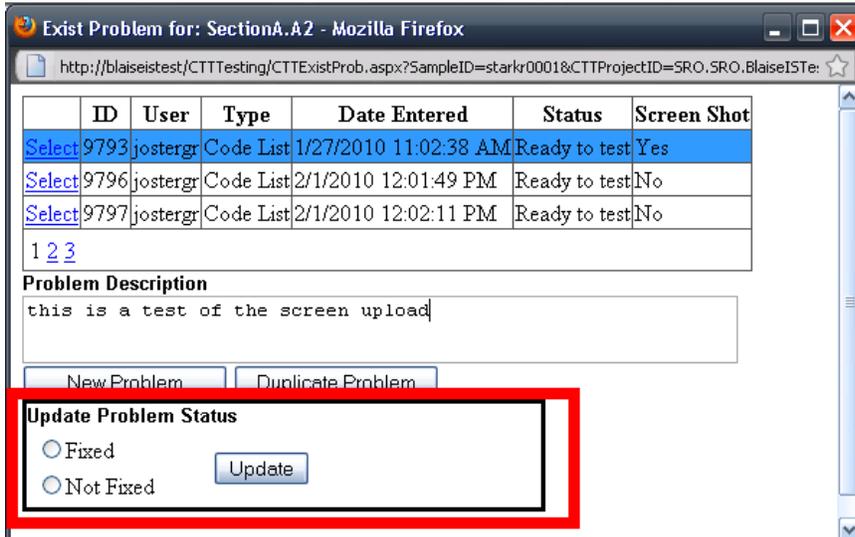
5.1.6 Sharing test cases

A further feature that has saved testers’ time is the ability to share cases or scenarios with others. In the CTT, testers can save cases they have created whilst testing and those with a higher access level can create and save ‘master’ cases for others to use when testing. This eliminates the need for others to spend time recreating a reported.

5.1.7 Targeted retesting

When retesting the questionnaire the tester is able to access the ‘Ready to Test’ report. This report, in Adobe .pdf format, outputs a form displaying problems that the programmer has changed the problem status from ‘New’ to ‘Ready to test’ after they have repaired the problem. This lets the tester focus on retesting the specific problems in the instrument. After the tester has determined that the problem has been fixed, the tester will enter that information via the Update Problem status check box (figure 10).

Figure 10. Update Problem status screen



5.2 The programmer

5.2.1

For the programmer, the key advantage is that all comments are stored in a consolidated list, in a standardized format with other details, such as field name and tester username attached. The programmer no longer has to decipher emails, Excel spreadsheets, scraps of paper, or remember to act on comments provided face to face. Having comments all in one place ensures programmers have all the information in one place and streamlines their editing time.

5.2.2. Specifying reports

A further key feature is the reporting function in the CTT. The programmer can either view or print standard reports or specify them. For example, programmers can select just the problems they have been assigned to fix or a general list of unfixed problems, and then work from a printed copy.

5.2.3 Identifying problems

Programmers have found that simply being able to view and manipulate the list of problems helps identify similarities in problems and can lead to further common problems being found. CTT gives programmers the more confidence in an instrument because they know it has been tested thoroughly and can release it with more confidence.

5.3 The Manager

5.3.1 Control of the testing process

The CAI Testing Tool enables the manager to have control over the following aspects of the testing process.

- Access to survey questionnaires or questionnaire versions is controlled by username. This ensures that the intended version of the questionnaire is used for testing and testers do not waste time working on an incorrect version.
- The manager can also control which aspects of the questionnaire are tested or the scenarios used by creating master cases.
- If necessary, the manager can control which problems are fixed first by prioritizing cases.

5.3.2 Setting up scenarios for testing

Setting up 'master' testing cases once has also streamlined the testing process for the manager because the scenarios do not need to be specified to the testers and only need to be specified once.

5.3.3 Managing problems

Storing all the testing information in one database is also beneficial to the manager partly, again, because of the efficiency of just needing to consult one list, but mainly because problems can be prioritized, sorted, and allocated to programmers to fix within the database.

5.3.4 Information about the testing process

Reports generated from the CTT allow the manager to access information on progress at any point throughout the testing process.

6. Conclusion

In summary, the CAI testing tool has made the testing process more efficient – saving time and thus money and improved the quality of CAI questionnaires.

The tool has fulfilled all the objectives set out to justify its development. Using the CTT ensures that all projects are following a standardized approach to testing and teams are not wasting time developing their own databases in which to collate comments. The tool has saved time for all those involved in the testing process. The CTT also stores and allows access to documentation about the entire testing process, providing a source of data about one project and the ability to run reports easily. The tool also acts as a repository for information about future enhancements and could provide information that could be used to influence development of future projects or the questionnaire development process.

From our experience CTT has facilitated a smoother (and more harmonious) testing process with all those involved working in the same environment accurate and appropriate information can be shared easily among the programmer, tester, and manager.

We have found the CAI Testing Tool to be a useful tool that reduces the burden of testing for all involved in the process. Although the quality of a questionnaire can only be as good as the time spent testing it-- this tool allows the tester to concentrate their efforts on releasing an instrument with minimal errors.

References

Dascola, M., Pattullo, G. and Smith, J. (2007): CTT: CAI Testing Tool Proceedings of the 11th International Blaise Users Conference, 2007, Annapolis.

Pattullo, G., Stark, R. and Batra, P. (2010) Testing Tool Manual Survey Research Center, University of Michigan.