Remote Usability Evaluation Tool

by

Narayanan Kodiyalam

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Applications

Dr. H. Rex Hartson, Chair

Dr. Manuel A. Pérez-Quiñones

Dr. Scott McCrickard

May, 2003

Blacksburg, Virginia

Remote Usability Evaluation Tool

by

Narayanan Kodiyalam

(ABSTRACT)

Interactive system developers spend most of their time and resources on user interface evaluation in traditional usability laboratories. Since the network itself and the remote work setting have become parts of usage patterns, evaluators do not have unlimited access to representative users for user interface evaluation. Reproducing the user's work context in a laboratory setting is also difficult. These problems have led to the concept of Remote usability evaluation that takes interface evaluation of any application beyond the laboratory setting. The main aim of this thesis work is to develop a tool that can record problems faced by remote users in the form of text and video. The text report and video, which is a sequence of the user's actions while encountering the problem, would help evaluators in preparing usability problem descriptions. This thesis reports the development of the remote usability evaluation method and the process of usability evaluation performed in enhancing features offered by the tool.

Acknowledgements

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. *Problem Statement*

Interactive system developers spend most of their time and resources on user interface evaluation conducted in traditional usability laboratories. In these laboratories, a small number of selected users are directly observed by trained evaluators. With the advent of the Internet, the number of users at remote and distributed locations have increased manifold. Bringing these distributed users to the laboratory for direct observation in usability evaluation is neither feasible nor cost-effective. Reproducing the work context of a remote user is difficult in a laboratory setting. The existing lab-based formative evaluation technique, though effective, has its own limitations. Transporting the users or developers to a remote setting is also cost proibitive. These barriers to traditional usability evaluation led to the concept of remote usability evaluation. With the Internet as a bridge, remote usability evaluation takes interface evaluation to a broad range of users in their natural work setting. A tool that records problems encountered by a remote user is very important for usability evaluators.

1.2. *Research Goals and Contributions*

The overall goal of this thesis work was to develop a deployable tool based on cost-effective method, for remote users to report critical incidents and evaluate the tool for usability problems. With this tool, users performing tasks in their normal working environment report critical incidents that happen during their day-to-day task situation. Data are captured from the user reports of critical incidents. This may or may not be accompanied by a screen capture of user actions related to the critical incident. Because there is no necessity for a direct interaction between the user and evaluator during an evaluation session, this is a cost-effective way of capturing the remote user's data. The data captured are of high quality and can be converted easily into usability problems by usability evaluators.

Castillo [Hartson & Castillo, 1998] worked on a low-fidelity prototype (see also Related Work) of the remote evaluation tool. The research goals and steps that comprise the work completed for this thesis in developing a new remote usability evaluation tool includes:

1. Developing the original work done by Castillo and creating an advanced version of the tool
2. Send critical incident screen capture as a video sequence for evaluators and usability practitioners
3. Broaden the features and functionality of the tool
4. Tool based on a cost-effective method
5. High usability in the tool itself

The secondary things completed as part of the thesis include

1. Feasibility study on available technologies and the platform for the tool to be built.
2. Doing a formative usability study on the tool
3. New methodology of usability lab set-up
4. Usability evaluation of the tool that focuses on ease-of-use and ease-of-understanding for a typical user
5. Doing a cost-importance analysis on the problems found with the tool and incorporating ideas to improve the tool

The steps are explained in subsequent sections of this thesis report. The method used for conducting remote usability evaluation is called the user-reported critical incident method and was reported in Hartson and Castillo [1998].

The contributions of my thesis to the research world include:

1. Exploring the opportunity of a new usability evaluation method that involves users from any location across the world.
2. New laboratory setup for usability study of any application that assists the evaluators and developers of that particular application.

3. Improving the design of a tool by way of cost-importance analysis of usability problems.
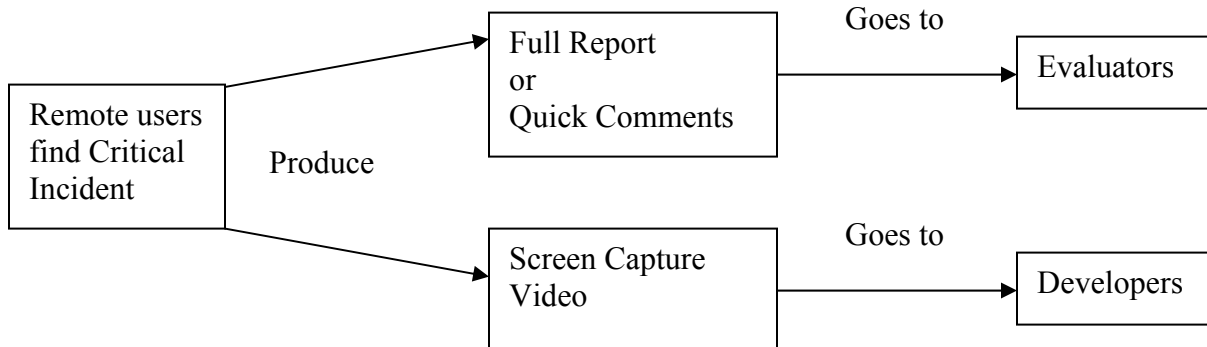
Figure 1-1: Overview of Remote Evaluation Method

An overview of the remote evaluation method is illustrated in Figure 1-1. Since the user identifies the critical incident and they have the most background knowledge of the problem, critical incident reports of very high quality are produced from the tool.

1.3. *Design Objectives*

The major design objectives of my thesis include:

- To give users maximum flexibility
- To follow a wizard-based model
- Ability to connect to web or non-web applications.

1.4. *Scope*

This thesis encompasses the following:

- Design, prototyping and implementation of the remote usability evaluation tool
- Iterative design walkthroughs and redesign
- Formative usability evaluation study on the tool with users (pilot user and real users) to find usability problems associated with the tool.
- Cost-importance analysis of usability problems found during the study
- Fixing usability problems found during the study

- Launching the tool in a real-world environment has just started though it is beyond the scope of this thesis.

This thesis does not include the following things:
- Study of tool performance
- Effectiveness of the tool

The effectiveness of this kind of tool is assumed from the study made by Castillo [Hartson & Castillo, 1998]. Taking this as an assertion, an advanced version of the tool has been developed as part of my thesis and a formative evaluation on the tool was performed.

1.5. *Overview of thesis*

Following this section, various methods of remote evaluation are discussed in section two. Section three discusses the design of remote evaluation tool, followed by a discussion of the Formative evaluation method conducted to find usability problems with the tool in section four. Section five discusses the results and lessons learned from the formative evaluation study, followed by the redesign ideas for improvement of the tool.

2. Related Work

Remote usability evaluation methods have been in use for development of many Internet based service and have been extremely popular to gather feedback about new products [Millen, 1999]. The work done by Jose Castillo and Dr. Rex Hartson on developing a user-reported critical incident method forms the basis for this thesis work. According to Hartson and Castillo, "the most significant impetus for remote usability evaluation methods is the need for a project team to continue formative evaluation downstream, after implementation and deployment" [Hartson & Castillo, 1998].

Through remote evaluation, Hartson and Castillo proposed using the network itself as a bridge to take interface evaluation to a broad range of users. They learned that users with no background in software engineering or human-computer interaction, and with the barest minimum training in these, can identify, report and rate the severity level of their own critical incidents. A feasibility study on user-reported critical incident method was conducted to determine two factors:

- if the method could provide approximately the same quantity and quality of qualitative data like that obtained from laboratory-based formative evaluation.
- if the method was cost-effective for both evaluator and user

Based on the insights gained from this feasibility study, a new method for conducting remote usability evaluation was developed called the user-reported critical incident method which was a low-fidelity prototype. The prototype could be used for web-based applications only. They derived a comparison of laboratory-based usability evaluation with remote evaluation and concluded that remote evaluation technique is as effective as the lab-based method. Some of the contextual factors associated with a critical incident have been discussed by them. Some of the possible applications of the critical incident method include – early formative evaluation, alpha, beta and other field usability evaluation, customer support and marketing strategies. A qualitative study was performed to gain insight and understanding about strengths and weaknesses of the method. Users who took part in their study were given brief training and a practice session in which they

gained experience identifying and reporting critical incidents via their web-based tool. The objectives of the qualitative study were to investigate feasibility and effectiveness of involving users with the user-reported critical incident method to identify and report critical incidents and to gain insight into the various parameters associated with the method.

The web-based tool consisted of a set of questions focusing on the user's experience with identifying the critical incident. The tool allowed user-subjects to report all the critical incidents they encountered in a structured way. The study was conducted with a set of three pilot users and twenty four user-subjects. All sessions of the user subjects performing tasks and simultaneously identifying critical incidents were videotaped. The experimenter developed a twenty-minute videotape for a consistent presentation about identifying critical incidents. Users had a practice session to get hands-on experience in reporting critical incidents using the web-based tool. The goal of their study was to evaluate the interface of the Internet Movie Database remotely using the web-based tool. Data entered by the user-subjects was analyzed and at the end of the study to determine the specific instants at which user-subjects identified the critical incidents, the quantity and severity ratings of critical incidents missed by user-subjects, the relevance of audio for the method and other useful information relevant to help identify critical incidents.

The critical incident data gathered were converted to usability problem descriptions with the help of volunteer subjects acting as evaluator-subjects. The different lists of usability problem descriptions created by these evaluator-subjects was analyzed to determine the feasibility of transforming remotely-reported critical incident data into usability problem descriptions, level of agreement about severity ratings between user-subjects and evaluator-subjects, quality of the critical incident report and the role of video, text and audio for this method. Case studies on these usability reports were conducted and the critical incident data were transformed to usability problem descriptions by usability experts. From the case study, they determined that "the user-reported critical incident method was a feasible method in the sense that it could provide approximately the same amount and value of qualitative data that can be obtained from laboratory-based

6

formative evaluation". A panel of three expert subjects viewed the tapes and tried to identify critical incidents missed by the user subjects. This led to another important result of their study - "expert subjects found very few critical incidents missed by user subjects, and those problems missed were of lower severity".

2.1. *Traditional Laboratory-Based Usability Evaluation*

Usability evaluation done in laboratories is the traditional way of evaluating any application or interface. Lab-based evaluation is usually considered as "Local Evaluation" as the user and the evaluator are in the same or adjacent rooms at the same time. Data collected in this case is both qualitative and quantitative [Nielsen, 1993]. Quantitative data are used to determine if usability specifications are met and when they were met so that iteration can halt while qualitative data serve to identify usability problems, their causes within the interface design, and potential redesign solutions [Hix & Hartson, 1993]. The sessions conducted by usability evaluators with the subjects are usually videotaped for backup and review.

2.2. *Critical Incident Reporting Tools*

Human-computer interaction researchers have developed software tools that help in identifying critical incidents and reporting them. Some of the existing critical incident reporting tools includes:

2.2.1. Online Critical Incident Tool

An online critical incident tool to collect critical incidents from user subjects was developed by del Galdo et. al. [Galdo, Williges & Wixon, 1986]. The study conducted by them investigates the use of critical incidents as a mechanism to collect end-user reactions to software documentation, the presentation of this information to systems designers, and the effectiveness of critical incidents as a tool in an iterative process of online and hardcopy documentation design. The principle purpose of this study was the development and validation of critical incidents as an effective tool for the incorporation of end-user feedback into the simultaneous design and evaluation of both online and hardcopy documentation. The online critical incident tool developed as part of their work consists of three components: 1) the format of questions for data collection, 2) the

procedure for data collection, and 3) the summarization of critical incident data into lists of interface problems and assets. The set of questions used for data collection included both open-ended and checklist style questions.

The three elements of essential information that were collected include – description of the critical incident, outcome of the incident classified as success or failure, and a severity rating of the incident (on a scale of 1 to 7). The evaluations performed led to a "decrease in the frequency of failure incidents and the alleviation of interface problems identified during previous iterations of tool use" [Galdo, Williges & Wixon, 1986].

This tool does not address points 2, 3 and 5 of the research goals described in section 1.2.

2.2.2. IDEAL

IDEAL is an interactive tool environment that comprises of several different tools and supports user interaction development activities, especially focusing on formative evaluation [Hartson & Hix, 1994]. It is a software and hardware tool for usability practitioners to use, in the front-line trenches of usability testing in the usability lab or in a remote location, for gathering and video tagging usability data, including:

- quantitative usability data (for assessing the level of usability achieved), mainly in the form of time-on-task, error rates, satisfaction ratings, etc.
- qualitative usability data (the most important kind of data for finding and fixing usability problems), mainly in the form of critical incident records and associated video clips

In 2001, the research group in Virginia Tech under Dr. Hartson began to develop a new generation IDEAL, mainly to:

- update the technology used for video capture, and
- integrate the data capture tool with the UAF-based data analysis and data management tools

Although the latest version of IDEAL is not yet complete, a well developed low-fidelity prototype is available that has undergone significant formative evaluation via walk-throughs and usability testing. The functionality of the IDEAL tool is organized around how and when it is used: before, during, and after data collection. The IDEAL tool links with the User Action Framework (UAF) tool which is a structured knowledge base organized as a search or classification space of usability concepts, issues, and situations.

This tool is not for remote users and so does not address points 1, 3 and 5 of the research goals described in section 1.2.

2.2.3. UCDCam

UCDCam, a tool developed by researchers in IBM, Toronto [http://www.lotus.com/products/screencam.nsf], is used for capturing digitized video of screen sequences during task performance, as part of critical incident reports. While users work with their normal tasks, UCDCam runs as a "background" process. When a user first activates UCDCam, the application opens a "Session Info" window to store user name, name of users' organization, name of the product being evaluated, and the disk where information would be stored. UCDCam also counts the number of reports sent by each user, and can automatically transfer clips back to developer [Hartson & Castillo, 1998]. The UCDCam software has been renamed to ScreenCam and can create movies seamlessly, quickly and easily. Some of the features offered by ScreenCam include:

- Link up to twenty shorter segments into seamless ScreenCam movies. There is no need to re-record an entire movie to improve a small section.
- ScreenCam movies can be stored on a Web page. They can easily be downloaded and viewed by other Windows users-even those who don't have ScreenCam.
- ScreenCam offers full Windows 95 integration including 32-bit support for OLE, Notes/FX, SendMail, dialog and more.
- Thumbnails are automatically generated for PC movies created with any version of ScreenCam, making it easy to determine their content

The ScreenCam software was not used for my thesis as IBM has clearly stated on its website that there are no plans for development to support Windows 2000 or Windows XP.

This tool addresses only point 2 of the research goals described in section 1.2.

2.2.4. NetCharts Server

NetCharts Server is a reporting tool provided by Visual Mining Inc [http://www.visualmining.com]. According to the company release, it is a perfect reporting tool with good flexibility and ease of use when on-line data visualization is a strong requirement. Some of the advantages offered by NetChart server include:

- Wizards for data access and chart, graph, and table development
- APIs for integration into your operating environments
- Powerful utilities for project development and integration

This tool is not for remote users and so does not address points 1, 3 and 5 of the research goals described in section 1.2.

My thesis is an evolution of Castillo's work on remote usability evaluation and details development of the remote evaluation tool and the usability evaluation study conducted on it to improve its features.

3. Remote Evaluation

Remote evaluation is defined to be "usability evaluation wherein the evaluator, performing observation and analysis, is separated in space and /or time from the user" [Hartson, Castillo, Kelso, Neale & Kamler, 1996].

3.1. *Remote Usability Evaluation*

Developers of an application often have limited access to representative users for usability testing of their applications in the laboratory. Moreover, the user's work context is difficult or impossible to reproduce in a laboratory setting. These barriers to traditional laboratory-based usability evaluation have led to extending the concept of formative evaluation beyond the usability laboratories to methods for remote usability evaluation. Remote evaluation typically uses the network as a bridge and takes interface evaluation to a broad range of users. These users can be located at remote locations in the globe and just have to be connected to the Internet. This mode of communication between the remote users and the Usability evaluator is illustrated by Figure 3-1.
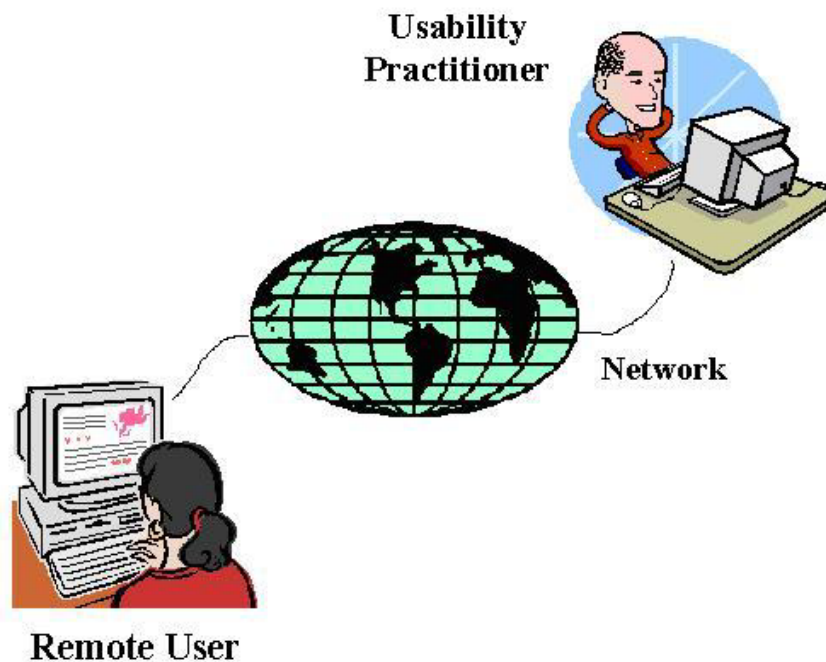
Figure 3-1: Remote User collaborating with Usability Evaluator

If a project team wants to continue formative evaluation downstream, after implementation and deployment, remote evaluation is a significant step in that direction. Remote capture method is very important because, post deployment usage data is not available to be captured locally in the usability lab.

3.2. *Types of Remote Usability Evaluation methods*

Empirical Formative Usability evaluation is defined as evaluation used for improving user interaction designs, especially early in the system life cycle [Hartson, Castillo, Kelso, Neale & Kamler, 1996]. The different kinds remote evaluation methods are briefly described below.

3.2.1. Portable Evaluation

In this method, the laboratory equipment is taken to the users in their normal work environments. It's often been termed as portable usability evaluation unit [Hartson, Castillo, Kelso, Neale & Kamler, 1996].

3.2.2. User Reported Remote Evaluation

The evaluation is local to the evaluators and remote from the developers, with results being returned via the network. The remote evaluation tool follows this type of remote evaluation.

3.2.3. Remote Laboratory Testing

Usability testing with representative users and tasks is done by consulting groups in a formal usability laboratory.

3.2.4. Remote Inspection

In this process, "developers send their designs to remote contractors who perform local evaluation using ad hoc, intuitive interface inspection, drawing on design guidelines, user profiles, and software standards" [Hartson, Castillo, Kelso, Neale & Kamler, 1996].

3.2.5. Remote Questionnaire/Survey

The software application displays a user questionnaire to gather subjective user preference data about the application and its interface.

3.2.6. Remote-Control Evaluation

The remote-control method employs control of a local computer from another computer at a remote site. Virtual Network Computing (VNC) is a tool that can be used for this purpose. More on VNC is available in section 5.2.1.

3.2.7. Video Conferencing as an Extension Of The Usability Laboratory

The user located remotely can be connected to evaluators using the network with video conferencing software. Available network bandwidth is a big obstacle to this approach.

3.2.8. Instrumented Remote Evaluation

An application and its interface can be instrumented with embedded metering code. Without interfering with the user's work activities, a log of data occurring as a natural result of usage in the users' normal working environment is collected and returned to the evaluators.

3.2.9. Semi-Instrumented Remote evaluation

This method uses selective data collection by users performing tasks in their normal work context.

3.3. *User Reported Critical Incident Method*

A critical incident is an event observed within task performance that is a significant indicator of some factor defining the objective of the study [Andersson & Nilsson, 1964]. In the formative evaluation context, a critical incident is an occurrence during task performance that indicates some positive or negative feeling about usability [Hartson & Castillo, 2000].

An informal study was conducted by Hartson and Castillo [1998] on the user-reported critical incident method. The conclusion made from the study was that users with no background in software engineering or human-computer interaction, and with the barest minimum of training in critical incident identification, can identify, report and rate the severity level of their own critical incidents. The result is significant because success of the user-reported critical incident method depends on the ability of users to recognize and report critical incidents effectively. Users are trained to identify critical incidents that may occur in a normal work environment.

With this hypothesis that users can effectively report critical incidents, the remote evaluation tool aims to extract that from the remote users by way of a Full report or Quick comments.

4. Design of the Remote Evaluation Tool

Usability engineering process involves task analysis, usage scenarios, iterative usability design, prototyping and usability evaluation. To produce a product with high usability, this process must be an integral part of the overall software development life cycle [Hartson, Shivakumar & Perez, 2003]. The usability engineering process was an integral part of the overall development of the remote evaluation tool. Since any user can identify and report critical incidents, they need a software tool or application that helps them in reporting the problems by just clicking an icon that launches the remote evaluation tool. A software tool residing on the user's computer would help in the collection of critical incident reports from users about problems they encounter while performing a task. Users are trained to use this tool to report critical incidents that happen during their work.

Users can send a full-report or give quick comments on the critical incident encountered. This can be accompanied by a video reenactment of the critical incident as the tool enables capturing the actions leading to the critical incident as a video sequence. This video sequence can be played back and seen by evaluators and developers later. This enables developers of the application in fixing the critical incident and evaluators in creating a list of usability problem descriptions. The video sequence is done with the help of a screen capturing software from Techsmith (http://www.techsmith.com) called Camtasia Screen Recorder SDK.

4.1. *Overview of the Remote Evaluation Tool*

The remote evaluation tool is a Visual Basic application that can run as an executable program (RemoteEvaluation.exe) on any machine. Microsoft's Visual Basic, a powerful tool for developing graphical user interfaces, is used as the front-end. The backend, which was earlier supported by Microsoft Access, is now supported by a MySQL server. The migration to the MySQL server was instrumental in making the database centralized. Figure 4-1 shows the overall flow of alternatives within the remote evaluation tool. Designing the flow of events for the tool was one of the first steps in the design stage.
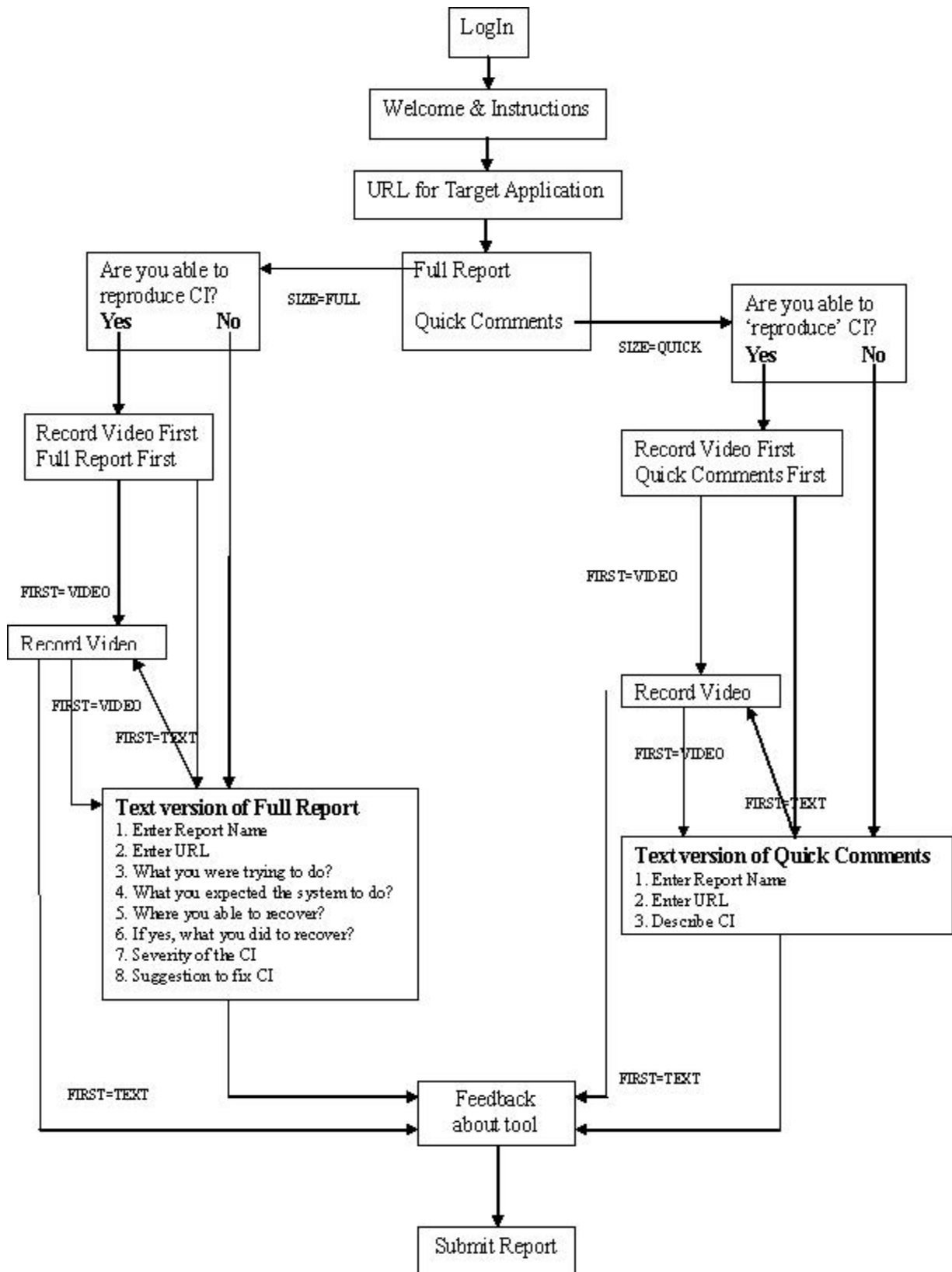
Figure 4-1: Flow of Remote Evaluation Tool

Users can follow four paths for reporting a critical incident. This offers a lot of choices in functionality of the tool. The different paths include:

1. Full Report with Video Capture. This path is followed if users feel that they can re-enact a critical incident.
2. Full Report without Video Capture. This path is followed if users feel that they cannot re-enact a critical incident.
3. Quick Comments with Video Capture
4. Quick Comments without Video Capture

The very first step is for the user to log in. The tool in a real-world scenario expects the user to specify the partition in their local machine where Camtasia SDK is installed. The screenshot of this question is shown in Figure 4-2. Camtasia has to be installed in the Program Files\TechSmith folder of any partition in the local machine. Once the user specifies the correct partition, the tool reads the license key of the Camtasia Screen Recorder SDK software, for the screen capture to work properly. If the user specifies the wrong partition, the tool will indicate an appropriate error message to the user. An alternative way to avoid asking the user for the partition is being left for future work.
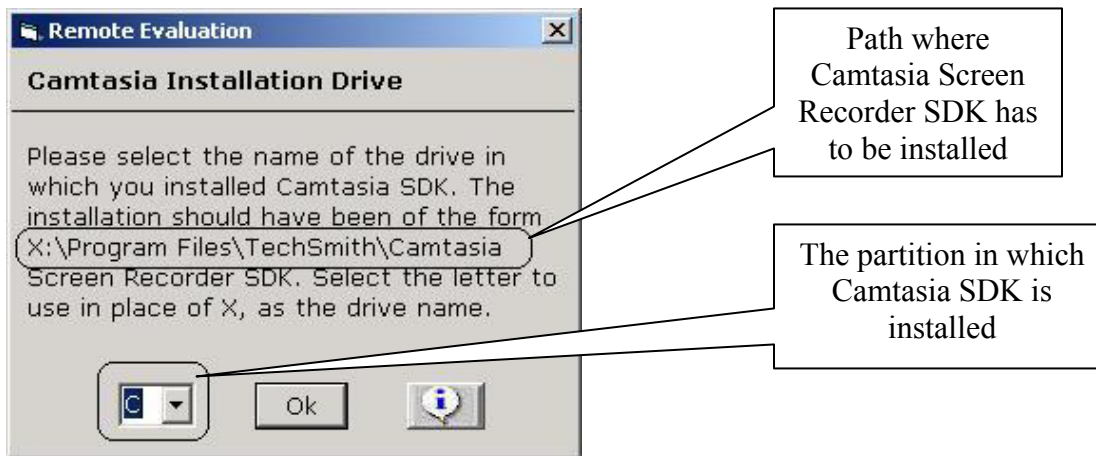
Figure 4-2: Camtasia Installation Drive Specification

The user reads a set of instructions shown in Figure 4-3, on how to report critical incidents. Users not familiar with identifying critical incidents are advised to take the

critical incident training by clicking the button just above the middle in Figure 4-3. The user moves on toward using an application and reporting critical incidents by clicking on the button labeled "Prepare to bring up target application with Report Incident button", shown at the bottom of Figure 4-3.
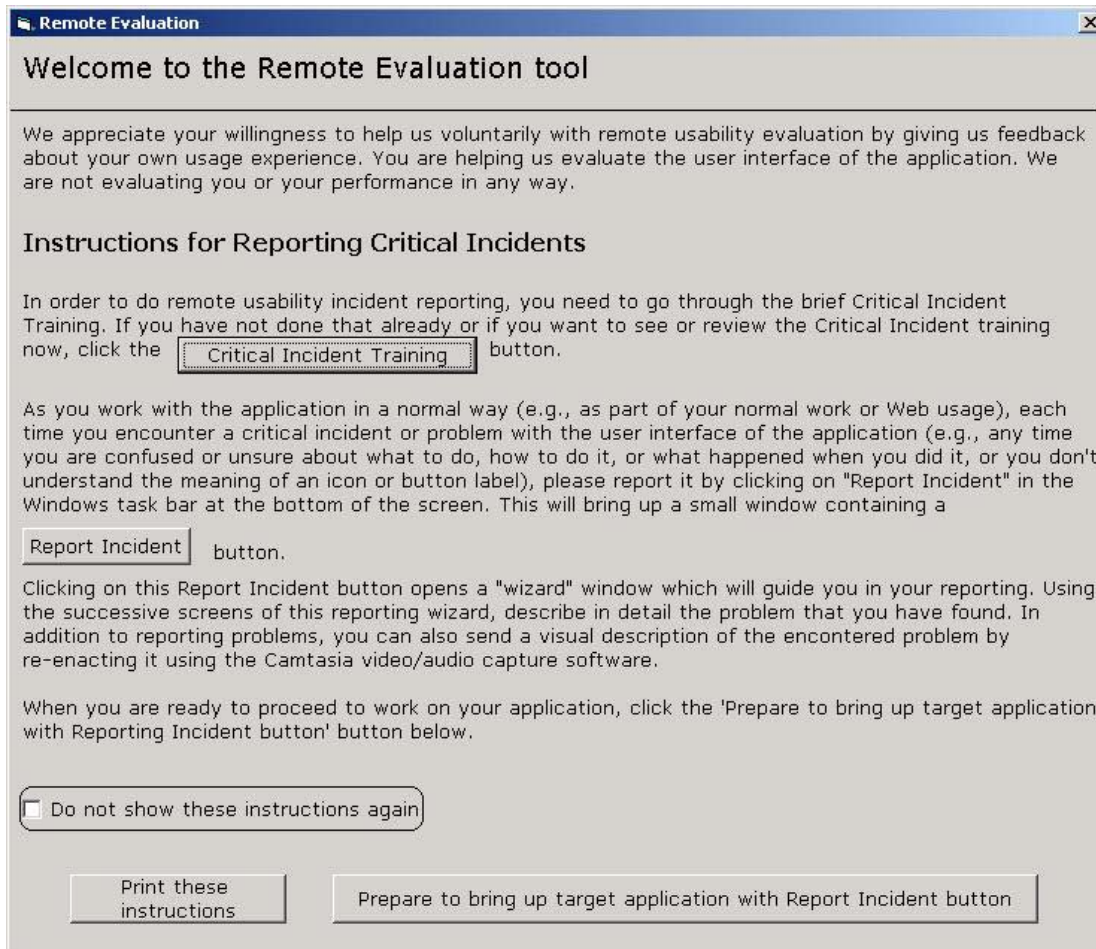
Figure 4-3: Instructions Screen

After this step, users can bring up their target web-based application by entering the URL of the application in the window shown in Figure 4-4.

Checking this option would not ask the user next time he/she logs in
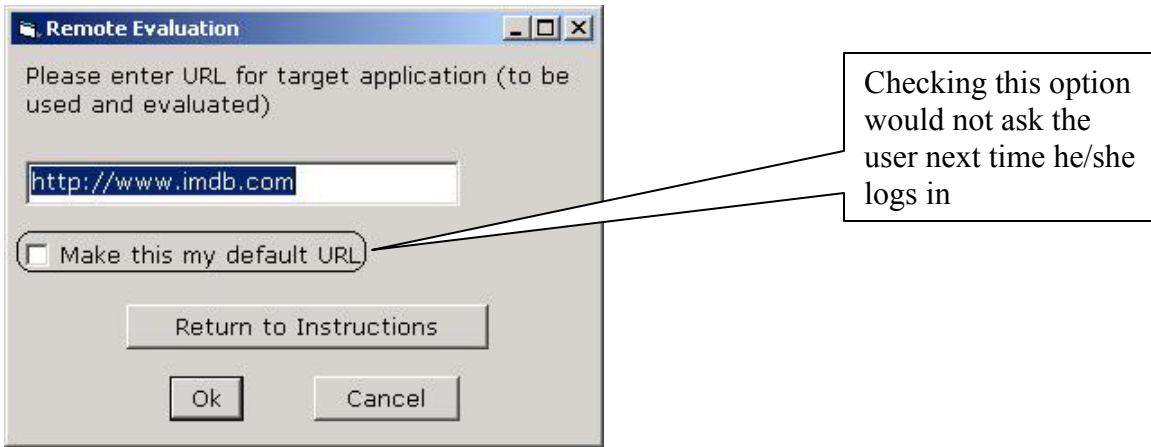
Figure 4-4: Target URL Screen

The Report Incident button will be available in the toolbar. If the user encounters a critical incident and wants to report the problem, clicking the 'Report Incident' icon in the Windows task tray would throw up the screen with the "Report Incident" button as shown in Figure 4-5. Closing this screen closes the remote evaluation tool. Clicking the 'Report Incident' button would enable the user to give a detailed report (called Full Report) or a short explanation (called Quick Comments). The wizard-based model starts at this point and guides the user until he is finished with submitting the report.

To report a problem encountered
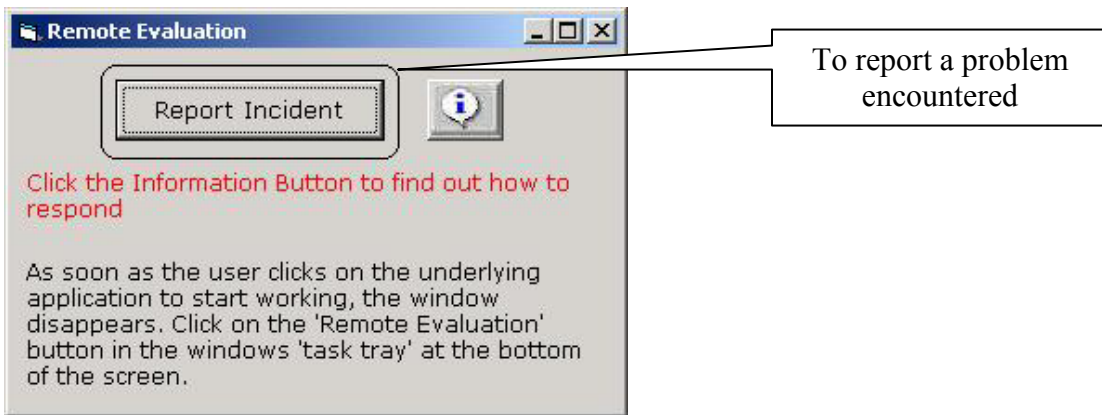
Figure 4-5: Report Incident Button Screen

4.2. *Components of the Remote Evaluation Tool*

4.2.1. Full Report Or Quick Comments

As shown at the top middle of the flowchart in Figure 4-1, users can give a full report or just enter quick comments describing the critical incident they encounter. The full report is comprised of nearly eight questions while the Quick comment contains only three questions. Usability evaluators will typically gain more from the descriptive answers given by users through a full report, but sometimes the user does not have time to make a full report.

Some of the questions which the full report covers compared to quick comments include:
1. What the user was trying to do when the critical incident occurred.
2. What the user was expecting the system to do just before the critical incident occurred.
3. Was the user able to recover from the critical incident? If yes, what the user did to recover from the critical incident.
4. Severity of the critical incident
5. Suggestions from the user to fix the critical incident.

These questions were collected from the study done by Castillo and Hartson [Hartson & Castillo, 1998].

4.2.2. Critical Incident Training

The instructions page of Figure 4-3 has a button that leads to the critical incident training page. First-time users who are not comfortable in identifying critical incidents might find the training very helpful. The training is a single webpage that gives the users inputs on identifying critical incidents. It also contains links to four movies that are in the form of .avi files. Each movie file has Dr. Hartson explaining the process of spotting a critical incident with the Internet Movie Database available at http://www.imdb.com. Two movies discuss the idea from a normal user's perspective while the other two discuss from an expert's perspective. The training is also available online at the link /HEMLOCK/CITraining.

4.2.3. Wizard-Based Model

The remote evaluation tool uses a wizard-based approach that steps the user through a set of questions. The wizard-like windows of the tool use a gray background with foreground text in black font, which is a common feature of any wizard-based tool. The user can navigate back and forth using the 'back' and 'next' buttons, discussed in detail in Section 4-3. The wizard-based approach helps the user in modifying answers to any question at any point of time. This is because, the user would not lose any text typed earlier for a question.

4.2.4. Critical Incident Screen Capture

Users can reenact critical incidents encountered and the reenactment can be captured as screen sequences with audio input. The audio input would contain the user "thinking aloud", while describing what he expected and how the critical incident happened. Camtasia Screen Recorder SDK, a product from Techsmith Corporation is used for this purpose. Camtasia Screen Recorder SDK is a toolkit designed to allow users to easily add screen recording from their Windows application. With this toolkit, screen activity, mouse movement and audio can be recorded to create movies of the desktop experience.

4.3. *Wizard-based model*

Our wizard behavior is based on the model of wizards in the Windows Operating System. The 'Add printer' option from Start->Settings->Printers, shown in Figure 4-6 is a good example of this wizard-based approach.
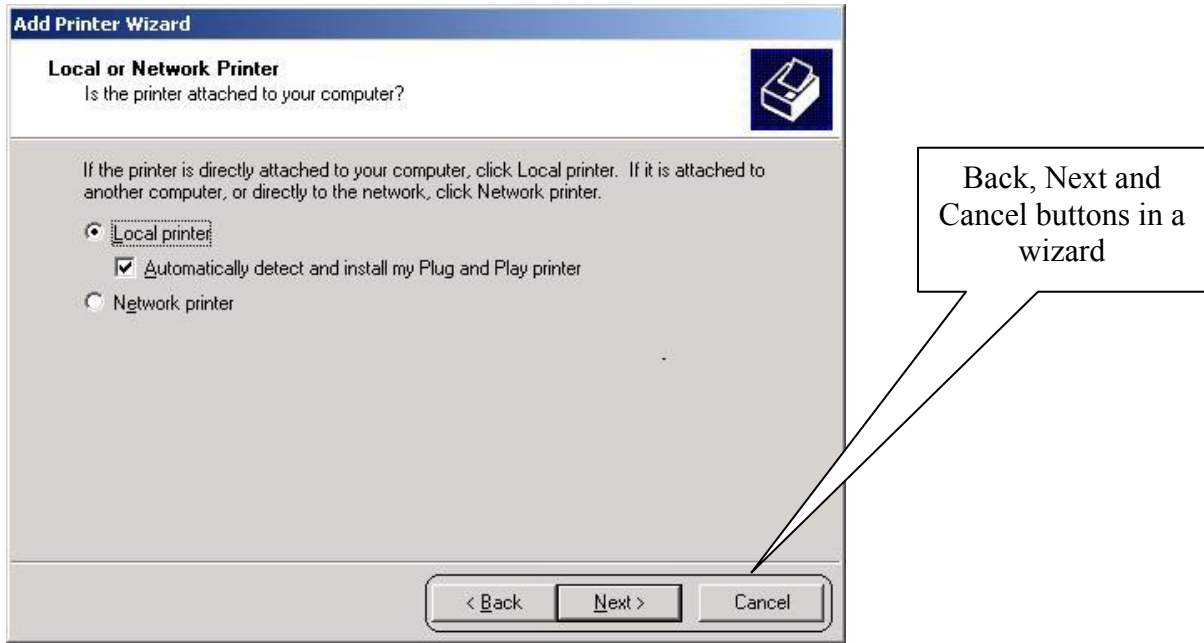
Figure 4-6: Add Printer Wizard in Windows Operating System

The Remote evaluation tool has features that closely follow any other wizard. By clicking the 'Do not show these instructions again' checkbox in Figure 4-3, the user would not be troubled the next time he/she logs into the system. This is an interesting feature as the user does not have to go through the problem of reading the instructions every time he/she logs in.

The most important feature is that users can navigate easily within the tool. The back and next buttons are handy when it comes to navigating within the tool. This is indicated by the screenshot shown in Figure 4-7. As the name suggests, the '< Back' button takes the user to the previous screen and the 'Next >' button takes the user to the next screen. The 'Cancel' button just closes the wizard without giving any warning to the user. The 'Report Incident' button in Figure 4-3 has to be clicked to start the wizard again.

The information button is yet another interesting feature provided by the tool. Users confused on how to answer a specific question or what a feature actually represents would find the information button very helpful. This is also shown by the screenshot in Figure 4-7.
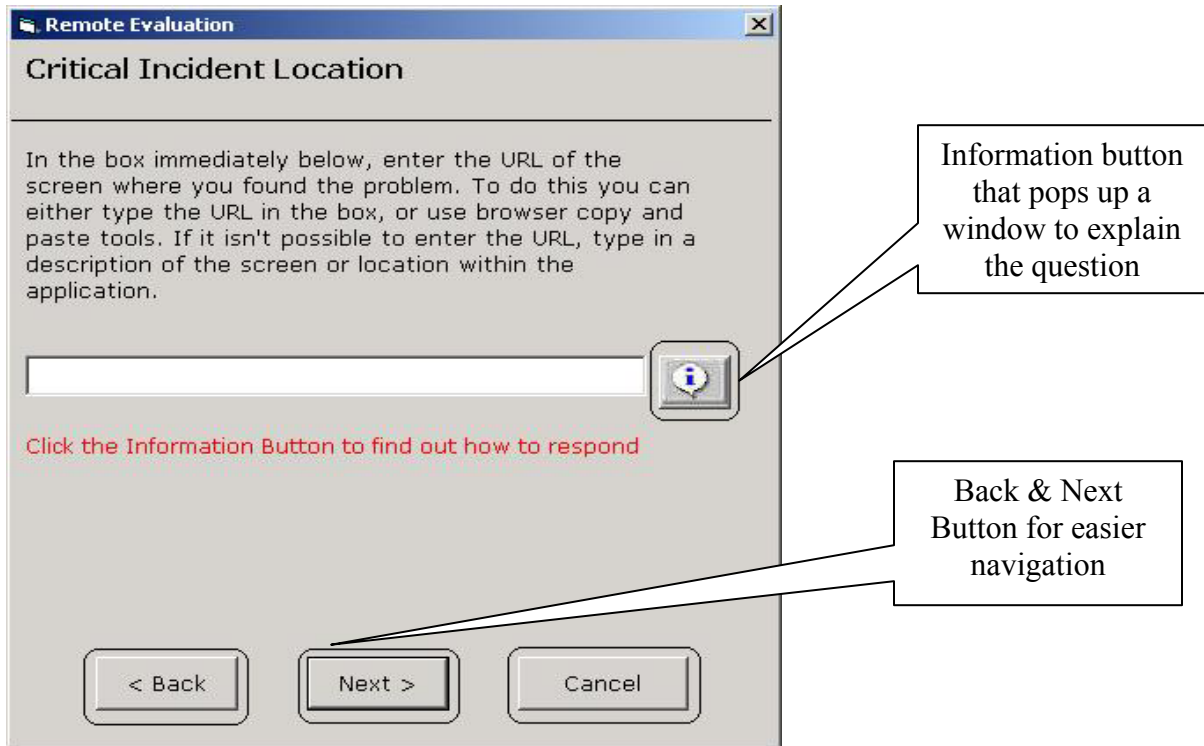
Figure 4-7: Wizard with Back, Next and Cancel button

4.4. *Critical Incident Screen Capture*

Users can report problems encountered and can even capture a video sequence of how and where the problem occurred. Instructions are given to the user on making a video clip of where the problem encountered. Figure 4-8 shows the screenshot of this particular scenario. The user has to click the 'Record' button. This would connect to Camtasia Screen Recorder SDK's DLL and further action sequence on the screen gets recorded. The user can start to reenact the problem encountered from this point onwards. Once the user is done with reenacting the critical incident, the 'Stop' button will have to be clicked. This creates a unique .avi file with a certain naming convention followed, as discussed in section 4.4.1.

Records the screen activity from that point

Stops recording the screen activity
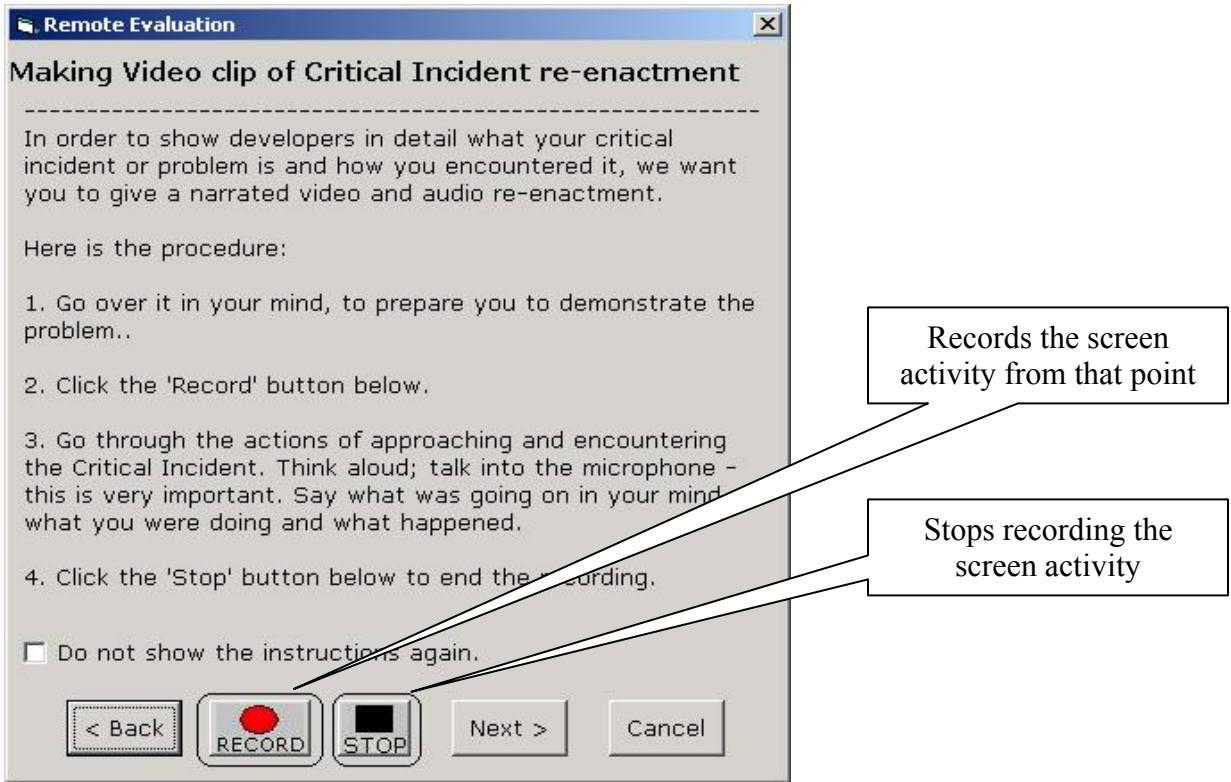
Figure 4-8: Record Video Instructions Screen

Clicking the 'Stop' button pops up a small window as shown in Figure 4-9, presenting the user with two options: keeping the just-recorded video clip or doing it over. Clicking 'Keep This File' button transfers the video clip file over the network to a remote server, by way of the file transfer utility of the Windows shell. Clicking the 'Do It Over' button would delete the video clip from the local file system. The user will have to reenact the problem again and record it by clicking the 'Record' button.

Transfers the screen capture file to the remote server

Deletes the screen capture file
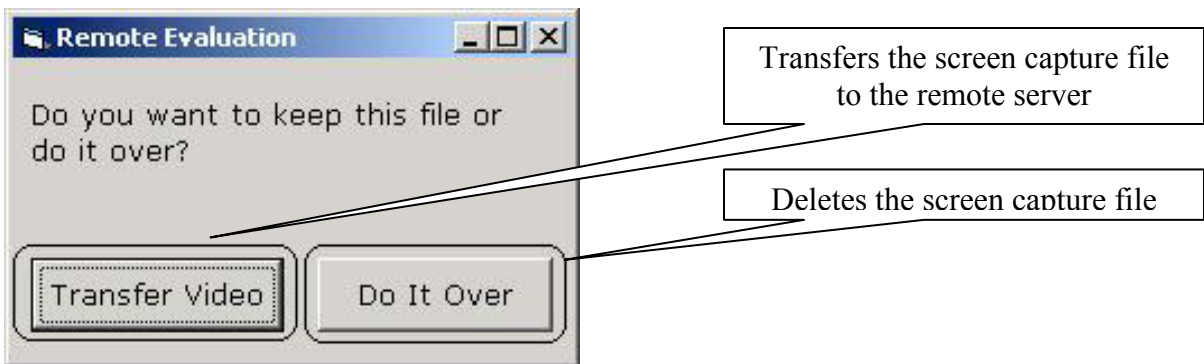
Figure 4-9: Pop-up Screen After Video Recording is done

4.4.1. Unique Screen Capture File Name

The sequence of events done by the user while reenacting the critical incident is recorded as an .avi file. The naming convention followed for the file is: CI which stands for critical incident is appended with the login name of the user. This is then appended with the date and time with an 'underscore' between them.

The filename would look like CIloginname_Date_Time.avi. The presence of the date and time creates a unique file name at any point of time.

4.5. *Technical difficulties*

Technical difficulties are inherent in the development of any application or tool. Some of the technical difficulties involved while developing the tool are discussed in the subsequent sections.

4.5.1. Migrating From ASP To VB

Majority of the modern computer users are connected to the Internet. Hence, to reach maximum users, it would be advantageous if an application is web-enabled. The original version of the remote evaluation tool was developed using Active Server Pages (ASP). An Internet Information Server (IIS) will be running and remote users can connect to the tool from any part of the globe just by typing a link in the browser. The tool couldn't connect to Camtasia Screen Recorder SDK's Dynamic Link Library (DLL). The DLL only takes structures as parameters to the methods and VBScript (server side script in ASP) did not support this. The technical support people from Techsmith were contacted about the problem. They responded that from the present version of Screen Recorder SDK, connection to its DLL can be made only from C++ or Visual Basic (VB) platform. Since there was no plan to support ASP atleast for the time being, the decision of migrating to VB was taken. Proper feasibility study was not done before going ahead with developing the application and this proved to be an important cost factor in terms of time involved. The concept of "Waffle Wisdom" applied perfectly to the building of this tool as the first prototype done in ASP had to be thrown away [Hartson and Hix, 1993]. The migration process showed that Feasibility Study is a very important step in designing

a software application and the technology needed to build it. Though the early prototypes were developed in ASP, it helped in fine-tuning the tool.

4.5.2. Centralized Database

The remote evaluation tool initially connected to an Access database. Validating user login, storing reports generated by the users and other validations were done by connecting to the Access database available in the local system. If the executable version of the remote evaluation tool (.exe) is available over the Internet, remote users can't use the tool as the database has to be centralized.

The problem was solved later by using the MySQL database server for centralization. MySQL database server is the world's most popular open source database. Its architecture makes it extremely fast and easy to customize. The functionality-rich MySQL is unmatched in speed, compactness, stability and ease of deployment. All the tables were recreated in the MySQL database server and connection was made from the tool to this centralized database server.

4.5.3. Must-Be Camtasia Installed

The remote evaluation tool requires that Camtasia Screen Recorder SDK is installed in the local machine for the tool to connect to its DLL successfully. This is because; the tool reads the Screen Recorder SDK software's license key to check if the software is installed in the local machine. This introduces the problem of 'Must-be Camtasia installed' in the local machine. Camtasia Screen Recorder SDK costs nearly $200 US dollars for a single licensed copy and to get it installed in every remote user's machine is simply impossible.

One workaround for this problem is for the Camtasia SDK to be installed in a couple of machines in a big organization. Users in the organization can then log into this machine to report critical incidents. Another workaround for this problem is to install Camtasia SDK in the server and connect to the server over the Internet. More research is yet to be done on whether this solution is feasible.

4.5.4. FTPing The Screen Capture File

The screen capture file (.avi) generated from the screen recording needs to be sent to the developers. The developers can then go through the file and probably fix the critical incident if it is a genuine one. A Visual Basic File Transfer Protocol (FTP) utility that transfers a file to a remote server was downloaded from the Internet. Adding this utility to the remote evaluation tool created problems and proved to be fruitless. This idea was dropped after a lot of research and unsuccessful attempts.

A solution to this problem was achieved by executing the shell command. The FTP application from the Windows System folder is called by the Shell command. Shell depends on the ftp application being available in one of the Windows system's folder to work. The login name and password for the server is hardcoded inside the visual basic code. The file gets transferred to the server without any effort from the part of the user.

5. Formative Evaluation

Formative Evaluation is "evaluation of the interaction design as it is being developed, early and continually throughout the interface development process" [Hartson and Hix, 1993]. A lab based formative evaluation study was conducted to find problems with the tool employing real users performing benchmark tasks in a usability environment. The qualitative and quantitative data collected would be helpful to improve the tool in future versions [Hartson, Shivakumar & Perez, 2003]. Formative evaluation was conducted over the whole life cycle of the design and consisted of two kinds of evaluation: large number of cycles of design walk-throughs discussed in section 5.1 and one final usability testing study discussed in section 5.2.

5.1. *Iterative design walk-throughs with Dr. Hartson*

Prior to the lab-based usability study, frequent and regular iterative discussions with my advisor Dr. Hartson focused on the design aspects of the tool. Sequencing behavior and wording problems within the tool were rectified during these meetings. Dr. Hartson corrected designer-centered terms to user-centered terms. The discussions also explored various technical options to make the tool more stable. Certain features like asking the user if they want feedback about the report they submitted were added to the tool later. This iterative walk-through process led to the next phase of my thesis – formative evaluation study of the remote evaluation tool.

5.2. *Laboratory-based study*

5.2.1. Equipment

Usability Methods Research Lab

The Usability Methods Research Lab at Virginia Tech is a state-of-the-art HCI laboratory for doing formative usability evaluation. Beyond usability evaluation, the lab facilities support development of new usability methods and tools by providing for a new role, that of meta-observer/evaluator to study both user and evaluator in their use of a particular method. Separated by a wall, the subject is placed in one room and the evaluator in the

other room. They can talk between themselves by means of an intercom present in both the rooms, as shown in Figure 5-1.
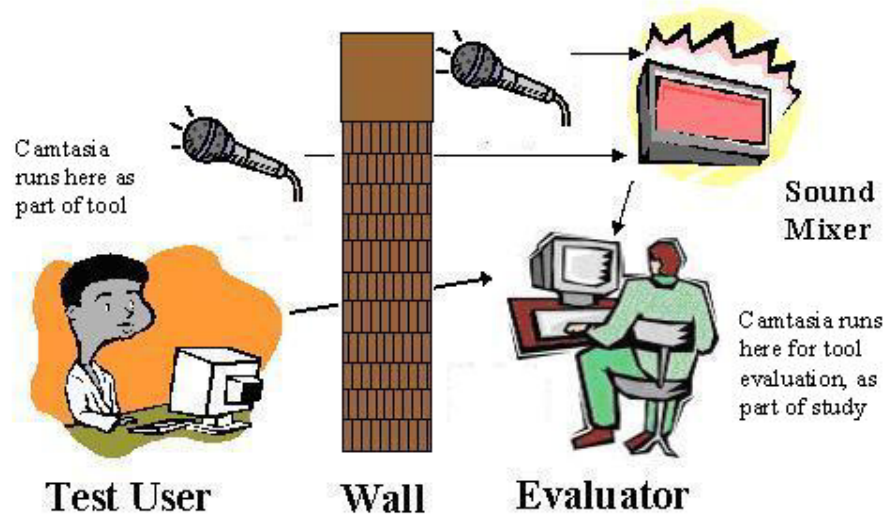
Figure 5-1: Laboratory Setup for Usability Testing

Dual monitors

Dual monitors were used for the test user and the evaluator present during the evaluation study. The dual monitor display helps in seeing multiple windows at the same time and prevents the user from switching between different applications.

In the subject's machine, the remote evaluation tool was open in one monitor and the website of the target application (www.imdb.com for the study) was opened in the other monitor. In the evaluator's side, the user's desktop was captured with the evaluator's comments about the user's action and reaction to events on his screen.

Use of VNC Software

Virtual Network Computing (VNC) is a remote display system that allows the Evaluator to view and *manipulate* the 'Desktop' or 'Screen' view of the Participant sitting in the other room. This way, the Evaluator can remotely and non-intrusively observe the Participant's screen. The VNC runs in the application mode in participant's machine and

listener mode in the evaluator's machine. The evaluator can make comments about the user's performance with the tool.

Microphone and Mixer Setup

The microphone and mixer setup used for the evaluation study allows a dual sound stream to be recorded, one sound input from the participant's microphone and the other from the evaluator's microphone. Both the inputs go through the audio mixer and into the evaluator's machine. A visual description of this scheme is shown in Figure 5-1.

Video Capture - Recording using Camtasia

Camtasia runs on the evaluator's machine from the time the study starts with a particular user. The Camtasia software records on-screen activities with audio input. The on-screen activity in the evaluator's machine is the participant's window and the audio input is the combined audio of the participant and the evaluator. Another Camtasia would be running in the participant's machine as part of the tool.

5.2.2. Participant Selection

A total of seven users participated in the study. Of the seven users, five had usability engineering background by way of the course CS5714 (Usability Engineering) while the other two did not have any usability engineering background.

5.2.3. Benchmark Tasks

The users were given four benchmark tasks to be performed with the remote evaluation tool. Each benchmark task focused on a very narrow usage situation, identified a critical incident with the Internet Movie Database (http://www.imdb.com) and asked the user to follow a certain path of reporting it. The four paths discussed in section 4.1 were used, one for each benchmark task.

5.2.4. Procedure

5.2.4.1 Before Study

The human-computer interaction laboratory at Virginia Tech was used for the study. The actual laboratory set-up has been described in the section 5.1 and more details are available in Appendix section C.

5.2.4.2 During Study

Users were greeted with a warm welcome and taken to the usability lab. They were not required to sign any forms as video recording was not involved. After a small introduction about the study, users were given the benchmark tasks and I went to the evaluator's room. Users were asked to use the tool to report critical incidents and were advised to follow the verbal protocol. Hence, usability problems associated with IMDB was not of any significance as finding usability problems with the tool was the goal. While participants were finding problems in IMDB and reporting using the tool, I was observing problems they had with the tool from the other room and noted them using pen and paper. The study took about 30 minutes per person. Users were not expected to take the critical incident training during the study. After the study got over, a small demographic survey was given to the user to complete followed by an informal open-ended interview to get their opinion on the tool. The demographic survey is available in Appendix section D.

5.2.4.3 After Study

After the study got over, problems collected over different user sessions were compiled. A total of 24 usability problems were identifies and cost-importance analysis was done. This has been discussed in section 6.3.

5.2.5. Data Collection

The primary means of data collection during the study was by taking notes on paper. The Camtasia Recorder running on the evaluator's machine recorded the entire session with the user and acted as the secondary means of data collection. These .avi files were played back many times and more problems associated with the tool were collected by the

evaluator. The size of the Camtasia files produced for different user sessions is shown in Table 5-1.

Camtasia File Name	Size of the file
participant1.avi	138MB
participant2.avi	124MB
participant3.avi	107MB
participant4.avi	101MB
participant5.avi	97MB
participant6.avi	142MB
participant7.avi	112MB

Table 5-1: Details of Camtasia Files used for study

5.2.6. Pilot Study

An initial pilot study was conducted with a fellow research student to avoid trivial bugs and problems associated with the tool. Some of the usability issues and minor problems were identified during this study. The task given to the pilot study user was to find a critical incident from a website and report it using the remote evaluation tool. In normal usage, critical incidents often occur over long periods of time in which the tool is not tested. For the compressed time conditions of this study, revised benchmark tasks were prepared and used for the user-subjects. This was an important lesson learnt from the pilot study. For the actual test users, critical incidents were actually given under the assumption that users can identify, report and rate critical incidents [Hartson & Castillo, 1998] and were just asked to use the tool to report the problem.

5.2.7. Study with Test Users

The target users were people with experience using tools and applications. The main purpose of the study was to identify usability problems with the tool in an effort to help improve usability of the tool in the interaction design. It was not aimed at finding usability issues with the website used as a target application system for the study or evaluating the relative performance of users. So, each benchmark task given to the user had a critical incident already identified (see Appendix A). All that was required from the user was to use the tool to report the problem. The users were asked to use various features of the tool and the problems they encountered with the tool were identified. I

acted as the evaluator to check if users had any problems with navigating the tool. The problems users experienced in understanding different features of the tool were noted and later analyzed.

One major fact found from the evaluation study was that users with usability experience were able to give valid usability-related inputs about the tool. Users without usability experience could just give inputs regarding the overall performance of the tool.

6. Results, Expectations, Discussion, Lessons learned

6.1. *Usability Problems found*

Problems identified during the pilot study were fixed before going ahead with the study with test users. A number of additional usability problems were found in the usability study. "A list of problems at the end of the day is an indication of success for the process and should not spell embarrassment or criticism for the designers themselves" [Hartson, Shivakumar & Perez, 2003]. Problems were identified by me and were only encountered by user subjects. The quality of inputs given by the participants was very high. A total of 24 usability problems in the tool, ranging from major to minor problems, were identified during the study with test users. The graph in Figure 6-1 shows the problem number for problems identified along the X-axis and the number of users who found it along the Y-axis. Nearly 40 percent of the problems were identified by the majority of subjects who participated in the study.
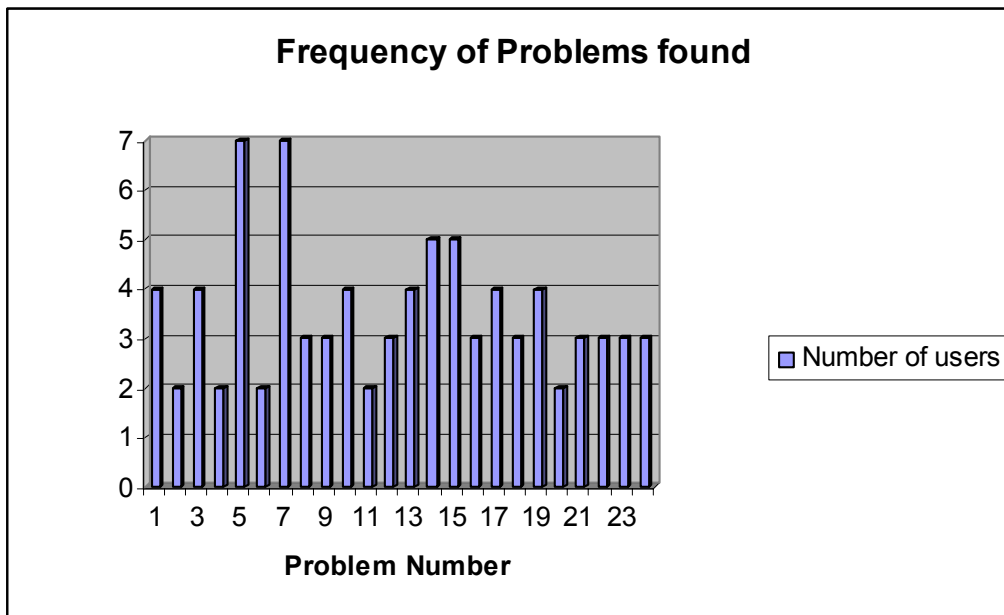
Figure 6-1: Frequency of Usability Problems found by users

The entire list of usability problems found during the evaluation study is available in the appendix section B. A few of those are discussed below.

The benchmark tasks provided to the user specified whether screen capture is required or not for that particular task. Though usability problem descriptions were complete and clear, users were confused on how to relate the task with the screens in the tool. Figure 6-3 shows the screen that asks the user if they can reenact the critical incident. The wording that appears in the screen confused users. They were unaware that clicking 'No' wouldn't allow them the screen recording feature. The implicit connection between reenactment and screen recording was not clear enough.
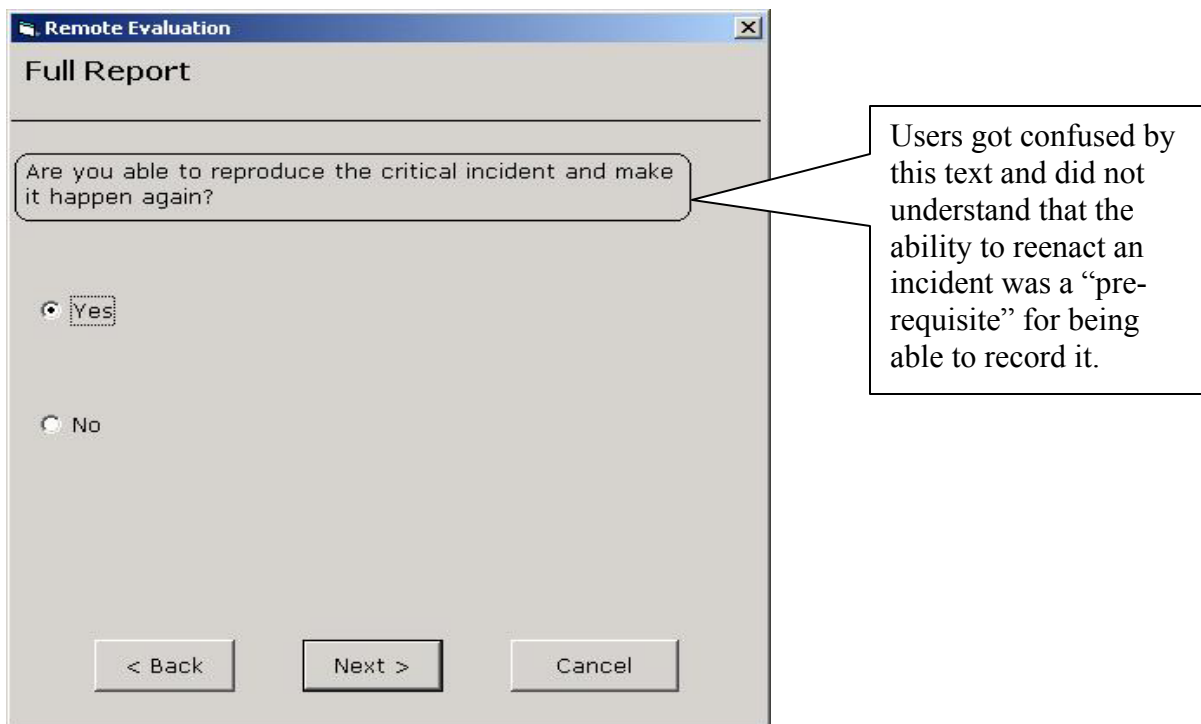
Figure 6-2: Ability to Reproduce Critical Incident Screen

Users felt that the severity ratings shown by Figure 6-4 were not clear and contained too many words. Instead, they preferred each severity rating to be marked by a couple of words like minor problem, major problem or critical problem. Some users felt that a color scheme needs to be given for each severity rating that shows the increase in the severity level. Hence, red would indicate the highest severity; orange would indicate the next higher severity etc. A redesigned version of these screens is available in section seven.

Comparing these screens with the redesigned screens would present a clear picture of the problems actually faced by users.
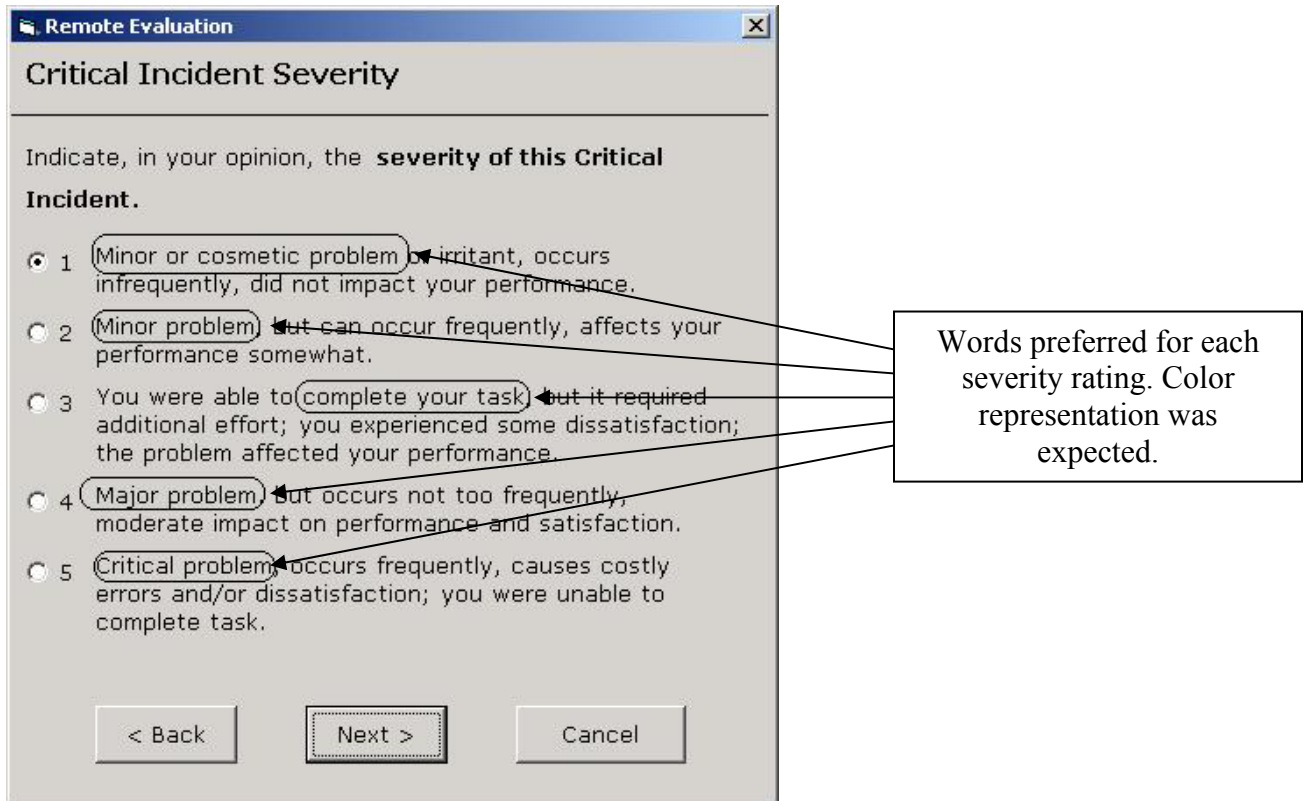
Figure 6-3: Severity Rating Screen

6.2. *Feedback from users*

All the users who participated in the evaluation study gave positive and negative features of the tool.

Some of the positive features include:

1. Good tool for reporting problems remotely
2. Easy navigation within the tool
3. Good flow of information within the tool

Some of the negative features were reported as usability problems and a few are discussed in the next section. A full list is available in the Appendix section B.

6.3. *Cost-importance Analysis*

Doing formative evaluation usually results in identifying a number of usability problems larger than what resources allow to be fixed. Cost-importance analysis is typical of a usability evaluation process [Hartson, Shivakumar & Perez, 2003]. Cost-importance analysis is a way to give each problem a priority ratio, a numeric estimation of importance to fix divided by estimated cost to fix. The problems are then sorted and ranked by this metric, favoring the highest importance and lowest cost. In a cost-importance analysis, the development team considers the relative importance of the problems and cost of the solutions [Hix & Hartson, 1993]. Importance in this context is importance to fix and can be based on problem severity and also other variables such as marketing inputs, management judgment etc. Table 6-1 shows a small summary top five critical incidents found during the study with their importance ratings and cost estimates. Section B in the Appendix shows a full list of the problems identified. The importance column is based on an importance rating of 1 to 5. The cost column indicates the number of man hours needed to fix the problem. The priority column indicates the importance to fix the problem.

Usability Problem Number	Usability Problem Description	Suggestions	Interface Location	Importance (to fix)	Cost (to fix)	Importance / Cost * 1000	Priority to fix
17	Closing 'Report Incident' window closes the entire application.	Pop-up a message box to the user saying he'll exit out of the application after this point.	Report Incident button screen	5	6	833.3333	1
7	Inability to reproduce Critical Incident meant that the user cannot do a video recording of the problem encountered. But, users got confused on the wording in the screen.	Make the wording more clear for the user to understand.	Can you reproduce Critical Incident screen	5	2	2500	2

1	"Report Incident" button in the instructions screen. Misled users to think that they can start reporting the problem from here.	Change the name of the button	Instructions Screen	4	1	4000	3
3	No link to the video recording file from the tool.	Give a link to the .avi file from the tool	Video Instructions screen	4	3	1333.333	4
4	No visual effects to differentiate between the severity of Critical Incidents	Give different colors to the severity ratings	Severity Rating Screen	4	3	1333.333	5

Table 6-1: Cost-Importance Analysis Table

7. Redesign for improvement

The remote evaluation tool was redesigned to incorporate some of the inputs given by the users. 16 usability problems out of the total 24 with descending order of priority were fixed as part of the redesign phase.

Figure 7-1 shows the redesigned severity rating screen after the evaluation study. The text associated with each severity rating has been reduced to a couple of words to prevent confusion. A color scheme is also available, with red indicating the highest severity and black indicating the lowest severity.
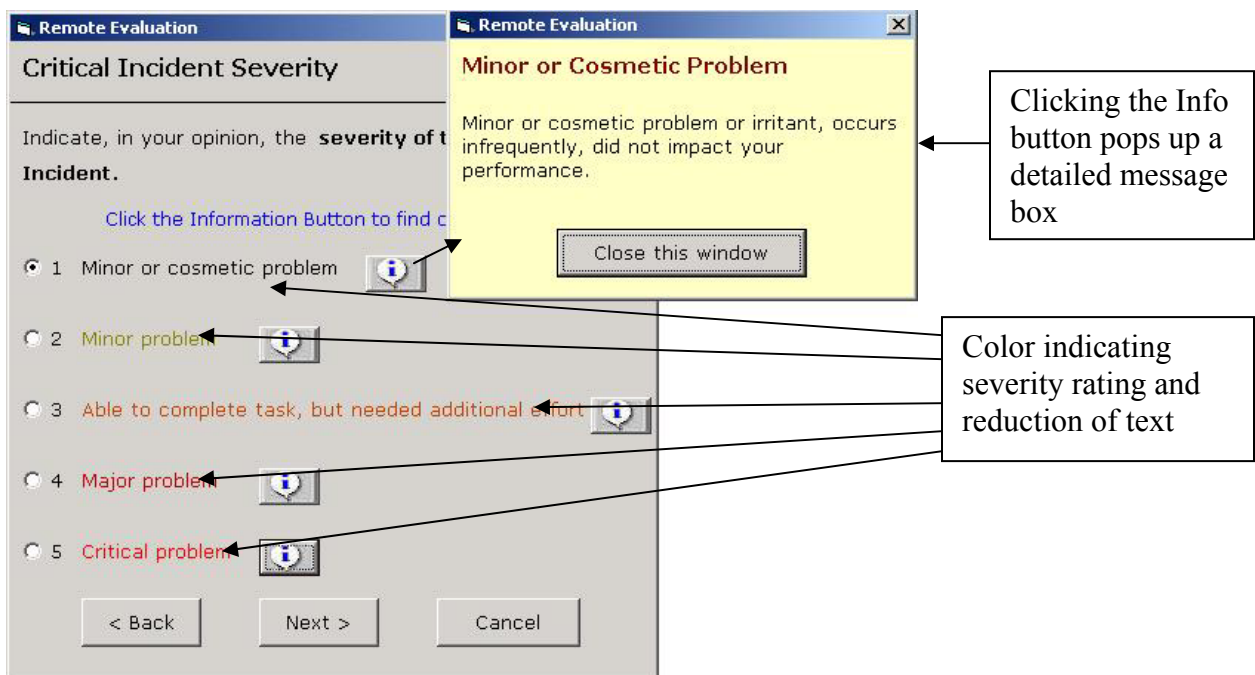
Figure 7-1: Severity Rating Screen after Redesign

Figure 7-2 is another screen after redesign. Users expected to know at what stage they were while answering the Full report. This screen shows that the user is answering the third question out of a total eight questions. This also helps the user in planning the answers for each question.
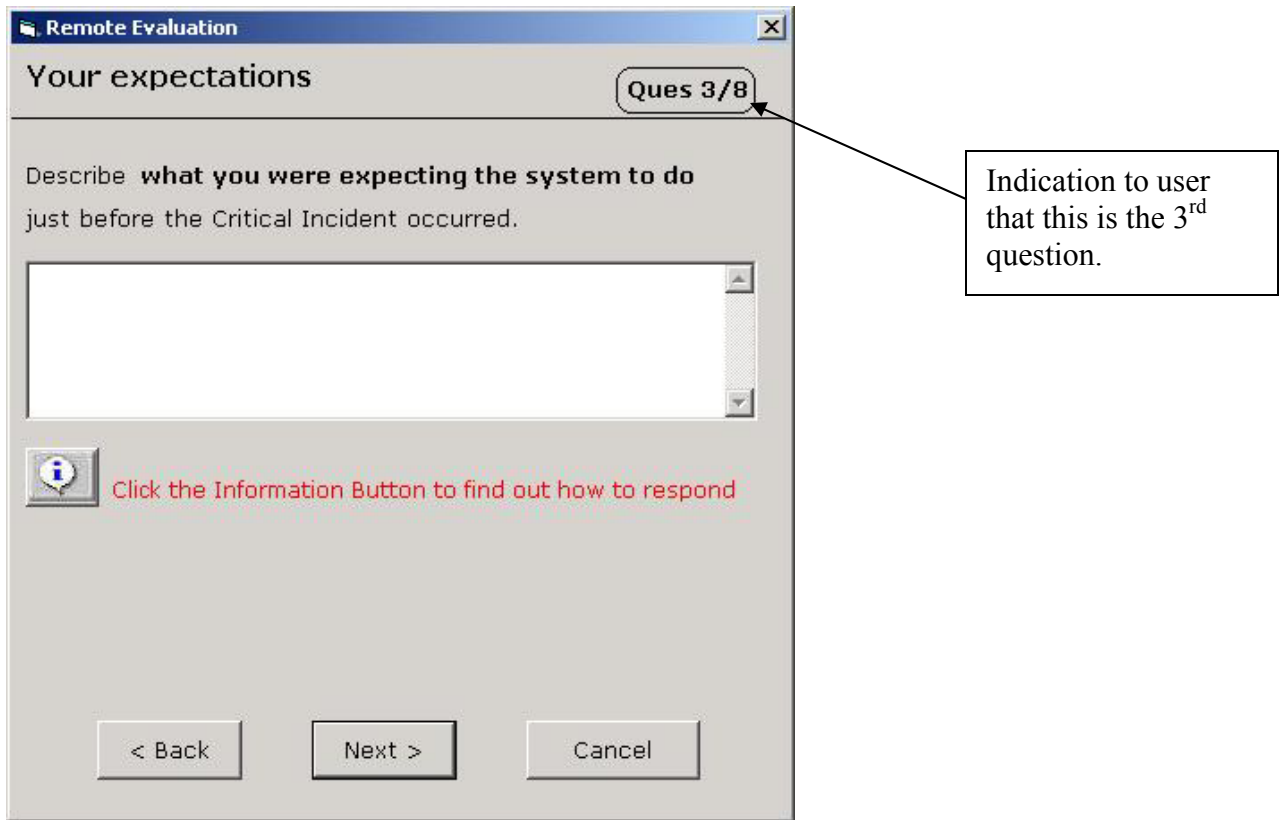
Figure 7-2: Full Report Question Screen after Redesign

The screen from which the 'Report Incident' button is clicked to launch the wizard is shown in Figure 7-3. Three usability problems fixed in this screen are shown with numbers one, two and three marked.

1. **Warning the user:** On closing the Report Incident window, the tool didn't warn the user that they are about to exit from the tool. Number 1 on the screen shows this fix. If the user tries to close the window, a message box pops up warning the user that the resulting action will exit from the tool. If the user clicks 'Yes', the tool closes. If the user clicks 'No', the window remains open without exiting from the tool. The message box that pops up is also shown in Figure 7-3.

2. **Inconsistency within the tool:** The Instructions screen indicated a 'Report Incident' window in the windows task tray while the tool showed it wrongly as 'Remote Evaluation'. Number 2 in Figure 7-3 shows this fix.

3. **Color cognitive effect:** The text about the information button shown in red confused the user as they felt that red is associated with an error or mistake. So, the redesigned screen has the text shown in blue. Number 3 in Figure 7-3 shows this fix.
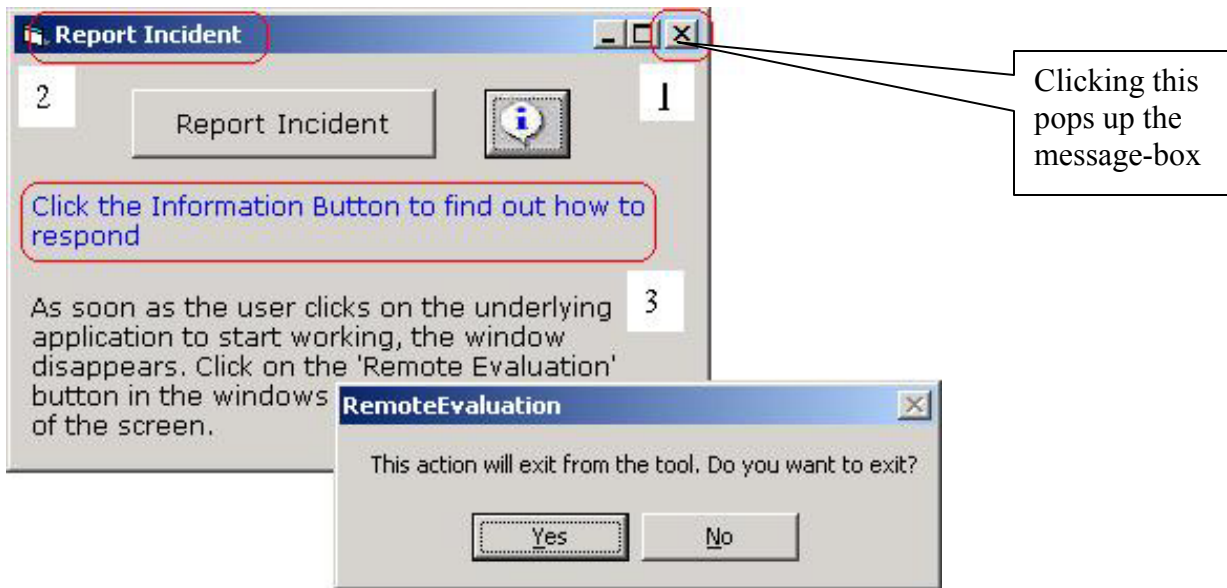
Figure 7-3: Report Incident Screen after Redesign

8. Summary

Though most of the usability evaluation is conducted in proper usability laboratories, this often not feasible with the presence of users at remote locations around the globe. The use of Internet as the main media for communicating among different parties has added to this effect. Remote evaluation is the evolving topic of discussion to solve this problem. Remote evaluation is defined as "usability evaluation where evaluators are separated in space and/or time from users" [Hartson, Castillo, Kelso, Neale & Kamler, 1996].

The main goal of my thesis is to develop and evaluate a cost-effective remote usability evaluation tool for real-world applications. Real users in real work environments can use the remote evaluation tool when problems are encountered while working with applications. This cost-effective way of capturing data helps developers to re-evaluate their application while simultaneously helping the evaluators in writing usability problem descriptions.

The remote evaluation tool designed as part of the thesis work is a wizard-based model. Developed in Microsoft Visual Basic platform, the tool captures problems encountered by users while working with an application, by taking users through a set of questions. The tool allows recording screen sequences of where the problem actually occurred with user's audio feedback describing the problem. This is implemented by connecting to Camtasia Screen Recorder SDK, a screen capturing software. The .avi file generated is then transferred to a remote machine without the user's knowledge for developers and evaluators to analyze the problem. The tool then went through the testing phase with users in a usability lab to identify usability problems associated with it. Benchmark tasks given to the users focused on using all features of the tool. The major Usability problems with their importance ratings and cost estimates were analyzed. Though further work needs to be done to add more features to the tool, thereby enhancing it, the present state of the tool is quite stable.

9. Conclusion

This thesis involved most phases of the usability development life cycle, namely – design, prototyping, implementation, testing and redesign. The feasibility study was one phase that didn't receive enough attentions and the effect of it was seen in the need to migrate from ASP to the Visual Basic platform.

Some of the goals set before starting work on my thesis and the end results of the thesis are compared in the table below.

Research Goals Set at start of thesis	How it was accomplished
Developing the original work done by Castillo and creating an advanced version of the tool	An advanced version of the remote evaluation tool was created as a result of this thesis with Castillo's work as the basis.
Send critical incident screen capture as a video sequence for evaluators and usability practitioners	The screen capture file is transferred automatically to the server for evaluators and usability practitioners to create usability problem descriptions
Broaden the features and functionality of the tool	The tool offers different paths for the user to take, as mentioned in section 4.1, for better functionality
Tool based on a cost-effective method	The tool is based on cost-effective method [Hartson & Castillo, 1998] because: • users collect the data • data is structured for easy analysis
High usability in the tool itself	Tool usability was addressed by the formative evaluation study to:

	• identify and remove tool usability problems
	• cost-importance analysis on problems found with tool and incorporating ideas to improve tool

Table 9-1: Goals Vs Accomplishment

I achieved the task of implementing a tool that helps remote users in reporting problems encountered in their normal working environment. The wizard-based navigation within the tool proved to be a positive feature as users found the tool easy to use. The technical difficulties that arise while developing a tool were tackled successfully. Usability testing of the remote evaluation tool through lab-based formative evaluation technique helped in finding problems with the tool and in fine-tuning it. The cost-importance analysis done for this purpose helped understand the perspective of usability inspection.

Continuing on my work, future researchers can focus on making the tool and the research concept reachable to more people in the field of usability engineering.

10. Future Work

There is scope for improvement in the tool via continued research. Each section described below stand as a separate line of research by itself for future researchers. Section 10.6 can stand as a separate thesis by itself while the other sections are non-thesisable.

10.1. *Web-enabled tool*

Maintainability, configurability, and scalability all contribute to the cost of implementing and supporting applications. Web-based applications provide unique advantages with regard to these qualities. A web-enabled remote evaluation tool would increase scalability as it could be accessed by anyone with internet access. Two possible choices available in making the remote evaluation tool web-enabled include:

1. Moving from Camtasia Screen Recorder SDK to different screen capture software that can be supported from an ASP platform. NetRaker, screen capture software similar to Camtasia Screen Recorder SDK, supports asynchronous video capture and transfer. NetRaker Experience Recorder combines actual videos of on screen activities with path data. More research has to be done as to whether this can be a good substitute for Camtasia Screen Recorder SDK. An initial feasibility study on NetRaker support from the ASP platform would be a good starting point for this line of research.

2. Techsmith coming out with Screen Recorder SDK's release that supports connection from the ASP platform. The present version of Camtasia Screen Recorder SDK could be supported only from C++ or Visual Basic platform. Support for an ASP platform could be achieved by collaborating with people in Techsmith. A web-enabled tool is more reachable and this could be a good incentive for the Techsmith people to come forward and provide support from an ASP platform.

10.2. *Presence of Camtasia in each machine*

The remote evaluation tool reads the license number of the Camtasia SDK installed on the local machine. The tool displays a screen for the user to enter the partition in their local machine where Camtasia SDK has been installed. The license key of the Camtasia

SDK software is got by reading the LicenseKey.txt file in the local machine. Hence, for the tool to connect to Camtasia SDK, it must be installed in the local machine. For the remote evaluation tool to be used by multiple users, numerous licenses for the software need to be purchased. This is a big bottleneck and would incur as a very high cost in the long run. Alternative software that can run from the server would be a better choice, if the features provided were equally good as Camtasia SDK. This software could run as a streaming video from the server that accepts certain number of connections at any point of time.

10.3. *Bookmark in the browser*

An important goal is to launch the remote evaluation tool by clicking a bookmark in the browser. MSN Messenger is a bookmarklet present in Internet Explorer that can be launched by a single click of the icon. Bookmarklets are small, but powerful Javascript applications that can be used to easily extend the functionality of a web browser. As of the present version, the remote evaluation tool can be launched as an executable file. Hence, a single click on the bookmark can launch the tool easily and efficiently. Presence of the remote evaluation tool as a bookmark in the browser would help remote users working on a web-based application to report critical incidents with more ease. One disadvantage is that bookmarklets would prove helpful only for a web-based application. More research needs to be done towards adding remote evaluation tool as a bookmarklet in the browser.

10.4. *Status of the report*

The reports submitted by a user go to usability practitioners and developers for analysis. Usability practitioners convert the reports to usability problem descriptions. After submitting a critical incident report, users expect to know the status of their report – the analysis and conclusions reached by the usability practitioners as well as developers. The feedback given by the evaluators and developers after going through the critical incident can help users in a constructive way. Users could probably fine-tune the way they report critical incidents by seeing the feedback. Research needs to be done on how the status of reports can be updated and users could view their reviewed reports.

10.5. *Input into UAF*

The User Action Framework (UAF) is a unifying framework developed as a foundation for structured methods and integrated tools to get the most value from usability data [Hartson, Andre, Williges & Van Rens, 1999]. The Usability Problem Classifier is a useful method for classifying usability problems by type. The Classifier would extract information from the Full report or Quick comments generated by a remote user and populate the Raw Data component (the first component in UAF with very basic information of what the name of the problem is, description etc.). These usability problems would help the evaluator to further classify the problems based on the types. Classifying the usability problems is a very important post-usability-testing activity and leads to a better analysis of the usability problems. IDEAL is a tool that helps usability practitioners in this regard as discussed in section 2.2.2.

10.6. *Summative Evaluation of the Tool*

Summative evaluation is defined as "evaluation at the conclusion of an activity or plan to determine its effectiveness" [Hartson & Hix, 1993]. Summative evaluation of the tool was beyond the scope of this thesis and the effectiveness of the tool was not tested as stated in section 1.4. Future researchers can take the thread from here and determine the actual effectiveness of the tool.

References

Andersson, B.E. & Nilsson, S.G. (1964). Studies in the Reliability and Validity of the Critical Incident Technique. Journal of Applied Psychology; 48(6), 398-403.

Balbo, S. & Paris, C. (2002). Multidisciplinary Considerations and Remote Usability Testing Tools.

Castillo, J.C. (1997). The User-Reported Critical Incident Method for Remote Usability Evaluation. MS Thesis, Department of Computer Science, Virginia Tech, Blacksburg, VA 24061 USA.

Del Galdo, E.M., Williges, R. C., Williges, B. H., and Wixon, D. R. An Evaluation of Critical Incidents for Software Documentation Design. In proceedings of *Thirtieth Annual Human Factors Society Conference* Human Factors Society, Annheim, CA, 1986, 19-23.

Flanagan, J. C. The Critical Incident Technique. *Psychological Bulletin.* 51, 4 (1954), 327-358.

Hammontree, M., Weiler, P. & Nayak, Nandini. (1994). Methods and Tools: Remote Usability Testing. Proceedings of HCI '94, 21-25.

Hartson, H.R., Andre, T.S., Williges, R.C. & Van Rens, L. (1999). The User Action Framework: A Theory Based Foundation For Inspection and Classification of Usability Problems. Proceedings of HCI '99, 1058-1062.

Hartson, H.R. & Castillo, J.C. (1998). Remote Evaluation For Post-Deployment Usability Improvement. Proceedings of AVI '98, 25-27.

Hartson, H.R. & Castillo, J.C. (2000). Critical Incident Data and Their Importance in Remote Usability Evaluation. In *Proceedings of the Human Factors and Ergonomics Society 44th Annual Meeting*, Human Factors and Ergonomics Society: Santa Monica, CA, 590-593.

Hartson, H.R., Castillo, J.C., Kelso, J., Kamler, J. & Neale, W.C. (1996). Remote Evaluation: The Network as an Extension of the Usability Laboratory. Proceedings of CHI '96 Human Factors in Computing Systems, 228-235.

Hartson, H.R., Hix, D., & Castillo, J.C. (1998). Remote Usability Evaluation: Can Users Report Their Own Critical Incidents. Proceedings of CHI '98 Human Factors in Computing Systems, 18-23.

Hartson, H.R., Shivakumar, P., & Pérez-Quiñones, M.A. (2003). Case Study of a Digital Library Usability Inspection. *to appear in Special issue of Journal of Digital Libraries: Usability of Digital Libraries.*

Hix, D. & Hartson, H.R. (1993). Developing User Interfaces: Ensuring Usability Through Product and Process. New York: John Wiley & Sons, Inc.

Hix, D. & Hartson, H.R. (1994). IDEAL: An Environment to Support Usability Engineering. *Proceedings of EWHCI '94: Fourth East-West International Conference on Human-Computer Interaction*, 195-211.

Millen, D.R. (1999). Remote Usability Evaluation: User participation in the Design of a Web-Based Email Service. Special Interest Group on Supporting Group Work (SIGGROUP) Bulletin '99, 40-45.

Nielsen, J. (1993). *Usability Engineering*. Massachusetts: Academic Press, Inc.

Swallow, J., Hameluck, D., and Carey, T. (1998). *User Interface Instrumentation for Usability Analysis: A Case Study.* HCI + Learning Laboratory, University of Waterloo (March). This paper was previously published at Cascon '97 (Toronto, Ontario) Nov, 1997.

Appendix –

Appendix A: Benchmark Tasks

<u>Task 1</u>

You are to use the Internet Movie Database (http://www.imdb.com/) to help us test our Remote Usability Reporting Tool by pretending that you have encountered the following Critical Incident.

Click on the 'Now Playing' tab in the homepage. It leads to a page which gives a list of currently playing movies and also upcoming releases.

Click on the back button in your browser.

Under the 'Coming Soon' heading in the left tab, click the 'more' link. It also leads to the same page.

Two links with different names and seemingly different content actually lead to the same page. This is a Critical Incident.

Please use the Remote Evaluation tool to report the problem using a Full Report **with** Video Capture.

Notice: In these tasks remember that the purpose is to evaluate the Critical Incident reporting tool and not really evaluate the IMDB site.

<u>Task 2</u>

You are to use the Internet Movie Database (http://www.imdb.com/) to help us test our Remote Usability Reporting Tool by pretending that you have encountered the following Critical Incident.

In the homepage, click the 'Opening this Week' heading in the left tab. This gives a list of all the movies to be released in the present year, not just the movies for this week.

The link is misleading. This is a Critical Incident.

Please use the Remote Evaluation tool to report the problem using Quick Comments **with** Video Capture.

Notice: In these tasks remember that the purpose is to evaluate the Critical Incident reporting tool and not really evaluate the IMDB site.

Task 3

You are to use the Internet Movie Database (http://www.imdb.com/) to help us test our Remote Usability Reporting Tool by pretending that you have encountered the following Critical Incident.

Click the 'Tops at the Box Office' heading in the left tab. It leads to the Box-office summary page.

Click on the back button in your browser.

Click the 'New Videos this Week' heading in the left tab. It is not clickable, unlike the other headings.

There is lack of consistency in the feature provided. This is a Critical Incident.

Please use the Remote Evaluation tool to report the problem using a Full Report **without** Video Capture.

Notice: In these tasks remember that the purpose is to evaluate the Critical Incident reporting tool and not really evaluate the IMDB site.

Task 4

You are to use the Internet Movie Database (http://www.imdb.com/) to help us test our Remote Usability Reporting Tool by pretending that you have encountered the following Critical Incident.

Click on 'U.S Movie Showtimes' tab in the homepage. It leads to a page where you can search for movies in Blacksburg (Zip code - 24060). Upon searching, it retrieves no results and gives the option of searching in the 'surrounding area'. Click this link and it goes back to the previous screen.

This is misleading to the user who expects to see the zipcodes of surrounding areas (which possibly exist in the database). This is a Critical Incident.

Please use the Remote Evaluation tool to report the problem using a Quick Comments **without** Video Capture.

Notice: In these tasks remember that the purpose is to evaluate the Critical Incident reporting tool and not really evaluate the IMDB site.

Appendix B: Cost Importance Analysis

| Usability Problem Number	Usability Problem Description	Suggestions	Interface Location	Importance (to fix)	Cost (to fix)	Importance / Cost * 1000	Priority to fix
1	"Report Incident" button in the instructions screen. Misled users to think that they can start reporting the problem from here.	Change the name of the button	Instructions Screen	4	1	4000	3
2	User forgot to click the "Record button" before reenacting the Critical Incident	Ensure that they click the button. By means of a message box.	Video Instructions screen	2	1	2000	18
3	No link to the video recording file from the tool.	Give a link to the .avi file from the tool	Video Instructions screen	4	3	1333.333	4
4	No visual effects to differentiate between the severity of Critical Incidents	Give different colors to the severity ratings	Severity Rating Screen	4	3	1333.333	5
5	Answer to Question 3 can be typed in a textfield. The text doesn't wrap automatically	Use a textarea instead of a textfield	Quick Comments Screen	4	1	4000	7
6	Button and the text in a screen are of the same font and size	Make the text in the background or in the button bold to differentiate them.	All screens	2	4	500	22
7	Inability to reproduce Critical Incident meant that the user cannot do a video recording of the problem encountered. But, users got confused on the wording in the screen.	Make the wording more clear for the user to understand.	Can you reproduce Critical Incident screen	5	2	2500	2

8	Not much use of the information button	Make the importance of the information button more prominent.	All screens	1	2	500	24
9	URL reported doesn't make much sense (like http://www.imdb.com/ for some screens)	Give a little description with the Url. Advise user accordingly	URL reporting screen	2	2	1000	19
10	Pop-up windows may be irritating	Reduce the number of pop-up windows	All screens	4	4	1000	8
11	Says 'Report Incident' in the text inside, but the window name is 'RemoteEvaluation'.	Change the name of the window	Instructions Screen	4	1	4000	9
12	"Your previous recording has been saved" should be changed to 'Your recording has been saved'	Change the wordings in the message box	Recording saved pop-up screen	2	1	2000	20
13	Questions 'Describe the CRITICAL INCIDENT' and 'Why you think it happened' is redundant in the same screen.	Remove the 'Describe the CRITICAL INCIDENT' question.	Full Report Questions screen	4	1	4000	10
14	No uniformity in the appearance of message boxes and pop-up windows in the screen	Set the position of the windows fixed in the screen.	All screens	3	4	750	16
15	Less text in the Severity rating description would avoid user from being confused	Reduce the text for each severity	Severity Rating Screen	3	3	1000	17
16	Instructions may be boring for first-time users to read	Reduce the content in the instructions screen	Instructions Screen	2	2	1000	23
17	Closing 'Report Incident' window closes the entire application.	Pop-up a message box to the user saying he'll exit out of the application after this point.	Report Incident button screen	5	6	833.3333	1

18	User not aware where in the wizard they are while answering the Full report	Give a description at the top of the wizard like question 5/8	All Full Report screens	4	5	800	6
19	User confused on which URL to enter while reporting the problem. Whether the page that leads to the problem or the page that has the problem	Make it more clear to the user.	URL reporting screen	3	2	1500	15
20	Users confused on whether to follow any convention while naming the usability problem	Explain to the user on following a naming convention	Usability Problem Name screen	2	3	666.6667	21
21	User confused on what to do if the application does not have a URL (not a web-based application)	Explain the user accordingly for a non web-based application	URL reporting screen	4	2	2000	11
22	The text 'first 3 questions ..'' creates a confusion among users as they are not sure how to divide the answers.	Give an idea to the user on how to divide the answers	Full Report - first question screen	3	3	1000	14
23	Emphasis on voice input & read-aloud technique not emphasized for people without usability experience	Lay emphasis on using read-loud technique	Video Instructions screen	4	3	1333.333	12
24	The text about Information button in red may alarm the users	Use a different color for the same	All screens	3	3	1000	13

Appendix C: Laboratory Set-Up Equipment Manual

Dual Monitor Setup

Enabling Dual-Monitor display	Disabling Dual-Monitor Display
Start> Settings> Control Panel> Display> Settings> Advanced> Dual-Head	Start> Settings> Control Panel> Display> Settings> Advanced> Dual-Head
Click on *Dual Monitor MultiDisplay*	Click on *Dual Monitor Disabled*
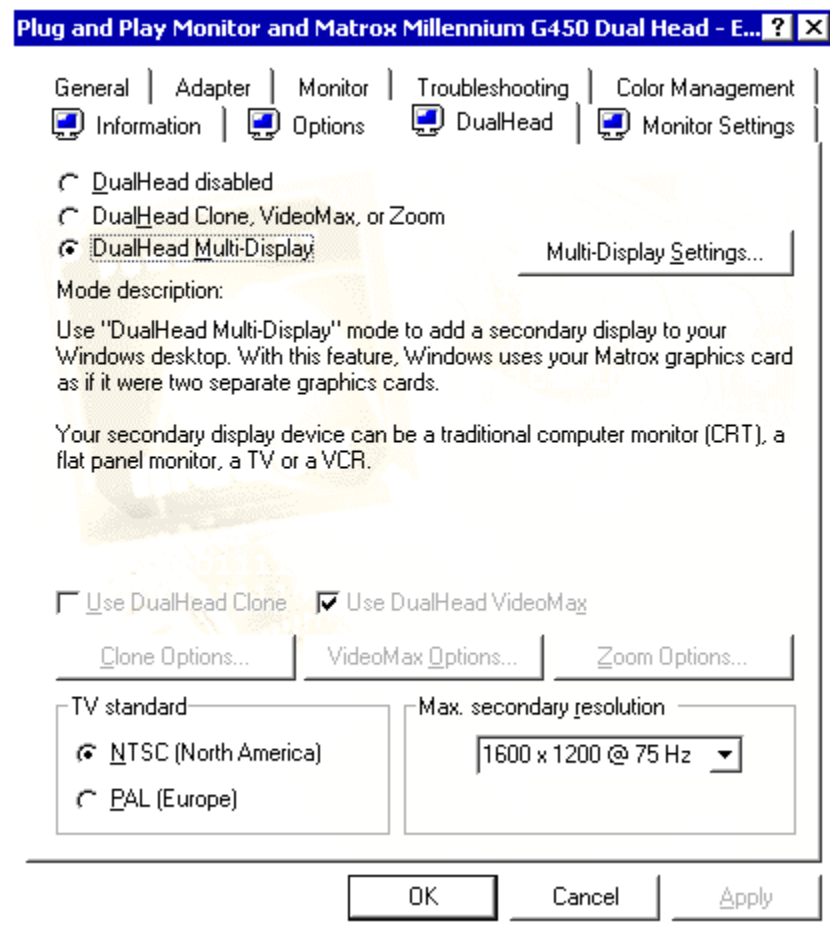
VNC (Virtual Network Computing) Setup

VNC allows the Evaluator sitting in room McBryde 102 B to view and *manipulate* the 'Desktop' or 'Screen' view of the Participant sitting in room McBryde 102 A. This way, the Evaluator can remotely and non-intrusively observe the Participant's screen. The Evaluator can also *take control* of the Participant's computing environment by manipulating the Participant's mouse and cursor.

1. Run WinVNC(App Mode) in the machine in 102 A. (This is the server)

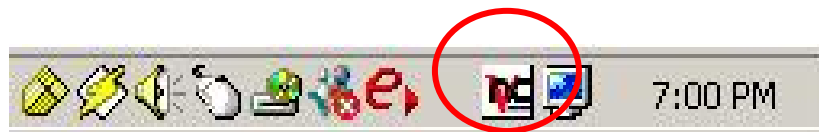   *Start>Programs>VNC>Run WinVNC(App Mode)*

This will start the WinVNC server and produce an icon in the system tray

2. Run VNCViewer in Listen mode in 102 B. (This is the viewer)

   Start>Programs>VNC>Run VNCViewer(Listen Mode)

This will start the WinVNC Viewer and produce an icon in the system tray
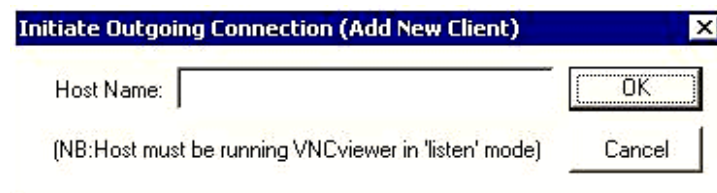

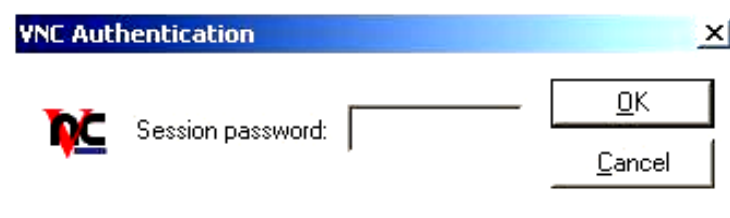The order in which you start the Viewer and Server does not matter.

3. In the machine in **102 A:** *Start>Run>Type "command"*. This will open the Command Prompt. Now *Type "ipconfig"*. Note down the IP Address. Then type *"exit"* to close the Command Prompt.

```
C:\WINNT\System32\command.com
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 128.173.41.22
        Subnet Mask . . . . . . . . . . . : 255.255.255.248
        Default Gateway . . . . . . . . . : 128.173.40.1

C:\>_
```

4. Now Double-click on the VNCViewer icon in the system tray of the machine in **102B** and enter the IP Address of the machine in 102 A.

```
Initiate Outgoing Connection (Add New Client)          [×]

Host Name: [                              ]      [   OK   ]

(NB:Host must be running VNCviewer in 'listen' mode)    Cancel
```

5. The VNCViewer now asks you the password of the machine in 102 A so that it can connect to it. The password is currently set to "*********".

```
VNC Authentication                                     [×]

                                               [   OK   ]
  VNC  Session password: [              ]
                                                 Cancel
```

Alternatively, you can also directly grant access to the Evaluator machine in 102 B from the machine in 102 A. For this, the VNC Listener should already be running in 102 B. Note down the IP Address of the machine in 102 B in a similar way as above. Then right-click on the VNC icon in the system tray of the machine in 102 A and select "Add a New Client". Enter the IP address of the Evaluator machine (102 B machine) here.
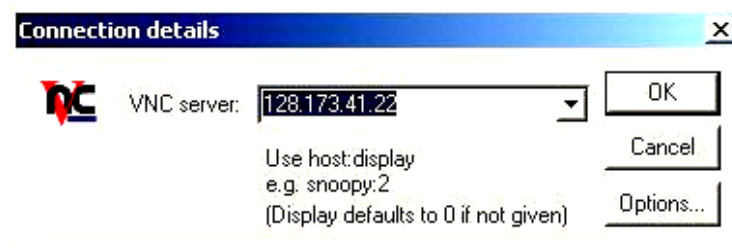
Microphone Setup

The Microphone Setup in 102 A allows a dual sound stream to be recorded. One sound input comes from the participant microphone in 102 A and the other from the evaluator microphone in 102 B. In the event that either of the microphones are not working, follow the instructions given below :

1. The Microphone Preamplifier (dbx 760x) in 102 B should be connected to the power supply.

   The Microphone Preamplifier has two sets of controls. The controls on the left are for the evaluator and the duplicate set of controls on the right are for the participant.

2. The evaluator's microphone in 102 B should be connected to the balanced input on the left in the Microphone Preamplifier.

3. The participant's microphone in 102 A should be connected to the balanced input on the right in the Microphone Preamplifier.

4. The unbalanced outputs from the evaluator and the participant should be connected to a connector which is then connected to the microphone input on the evaluator's machine in 102 B.

5. Adjust the volume levels on the Microphone Preamplifier.

Recording using Camtasia

1. *Start> Programs> Camtasia> Camtasia Recorder*

2. Press the record button.
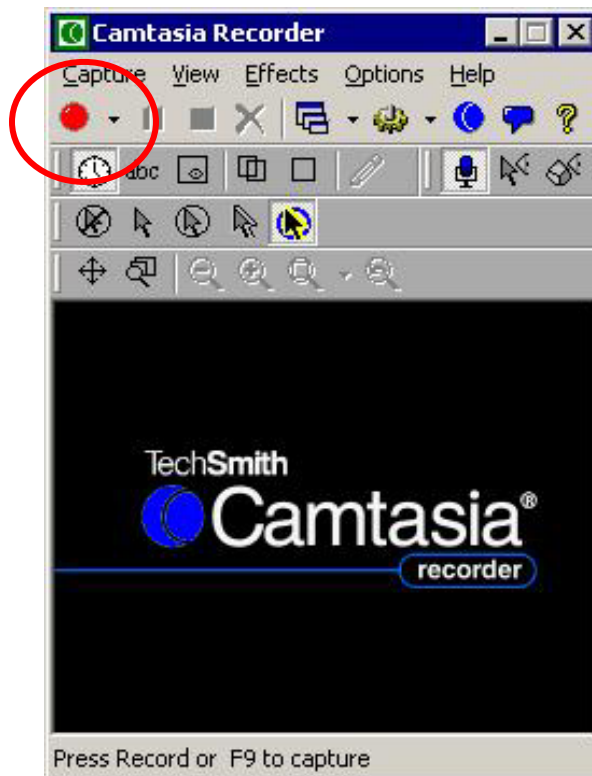
3. When done, press the square STOP button.

4. Save the file. The file will be saved with a "*.avi"* extension.

Playback

The created file can be best played with "Real Player" which allows the Evaluator to scan back and forth through the file. Alternatively, it can be played on Windows Media Player or the Camtasia Player.

Communicating with the Participant in 102 A from 102 B

Use the Intercom to communicate one-way with the participant in 102 B. The evaluator can hear the participant in 102 A through the speakers in 102 B. Hence if the evaluator wishes to speak to the participant he can use the Intercom. This can be turned on by pressing the 'TALK' button. If the evaluator wishes to keep the communication on continuously, he can do so by pressing the TALK and LOCK buttons simultaneously.

Recording the VNC Listener view using Camtasia.

After starting the VNC viewer, the evaluator will have a view of the Participant's screen. He can then start Camtasia and by capturing the VNC Listener Window a screen capture video can be made. With both Microphones on, the video will record both the evaluator's and participant's voices simultaneously.

Appendix D: Demographic Survey

How would you rate your usability knowledge, on a scale of 1-5?

Scale:

1 None	2	3 Good	4	5 Excellent

How would you rate your knowledge of the Remote Usability Evaluation tool on a scale of 1-5?

Scale:

1 None	2	3 Good	4	5 Excellent

Have you taken the 'Usability Engineering' course (CS5714)?

☐ Yes   ☐ No

Have you ever participated in a usability evaluation of software?

☐ Yes   ☐ No

Did you have any problems with the Remote Usability Evaluation Tool?

Do you have any positive/negative comments about the Remote Usability Evaluation Tool?

Were you comfortable with the experimental setup? Did you have any problems? Suggestions?

Vita

Narayanan Gopalakrishnan Kodiyalam, the son of Gopalakrishnan Kodiyalam and Mrs. Vijayalakshmi Gopalakrishnan, was born on June 24th, 1979, in Chennai, India. He graduated from Birla Institute of Technology and Science, Pilani in May 2000 with a Master of Science degree in Information Systems. After working for a year in Citicorp Overseas Software Limited, Chennai, he came to the United States to study at Virginia Tech in August 2001. This thesis completes his Master's degree in Computer Science and Applications.