THE USER-REPORTED CRITICAL INCIDENT METHOD FOR REMOTE USABILITY EVALUATION

By

José C. Castillo

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

H. Rex Hartson, Chair
Deborah Hix
Mary Beth Rosson
Robert C. Williges

July, 1997
Blacksburg, Virginia

Keywords: Remote Usability Evaluation, Usability Method, Critical Incidents, User-Initiated, User-Reported, Usability Data, Post Deployment

THE USER-REPORTED CRITICAL INCIDENT METHOD FOR REMOTE USABILITY EVALUATION

by

José C. Castillo

(ABSTRACT)

Much traditional user interface evaluation is conducted in usability laboratories, where a small number of selected users is directly observed by trained evaluators. However, as the network itself and the remote work setting have become intrinsic parts of usage patterns, evaluators often have limited access to representative users for usability evaluation in the laboratory and the users' work context is difficult or impossible to reproduce in a laboratory setting. These barriers to usability evaluation led to extending the concept of usability evaluation beyond the laboratory, typically using the network itself as a bridge to take interface evaluation to a broad range of users in their natural work settings. The over-arching goal of this work is to develop and evaluate a cost-effective remote usability evaluation method for real-world applications used by real users doing real tasks in real work environments. This thesis reports the development of such a method, and the results of a study to:

- investigate feasibility and effectiveness of involving users with to identify and report critical incidents in usage,
- investigate feasibility and effectiveness of transforming remotely-gathered critical incidents into usability problem descriptions, and
- gain insight into various parameters associated with the method.

DEDICATION

To my family, whose unconditional love and support
inspire me and keep me reaching for higher stars.
"¡Los quiero mucho!"

ACKNOWLEDGMENTS

I would like to thank my advisor and role model, Dr. H. Rex Hartson for his guidance throughout my graduate work, and Rieky Keeris for her caring and effort in ensuring I always had enough "research fuel" to keep on going with my work. Their sincerity, love, and friendship are invaluable and I feel proud to be considered their "adopted son". Dr. Deborah Hix, leading member of the "thesis demolition team", kept my work focused and I thank her for her guidance and devotion. I would also like to thank the other members of my committee, Dr. Mary Beth Rosson and Dr. Robert C. Williges, who have been helpful and supportive, and contributed greatly to this document.

Many other people have contributed to the completion of this work, and I am happy to have an opportunity to thank them: John Kelso, who has worked in the remote evaluation project since we started, for being supportive and for being patient when things went wrong in the usability lab; Pawan Vora, for his help in revising the thesis and contributing to this document; Jürgen Koeneman, Jim Eales, Ray Reaux, Dennis Neale, Mike McGee, Jonathan Kies, Brian Amento, Joe Riess, Winfield Heagy, and other members of the HCI group, for their advice and their help with my study; Dr. Wolfgang Dzida, for his assistance and for providing us with the ERGOguide; Don Hameluck at IBM in Toronto, for his interest in collaborating with our project and providing us with UCDCam; Lúcio C. Tinoco, for being a good friend and for letting me use his QUIZIT project in my research; all the people who participated as "subjects" for my thesis experiment, for their help and time; the secretaries of the Computer Science department, Tammi Johnston, Sharon Donahue, and Jessie Eaves, for giving me that smile that always kept my days shining; my good friend Mourad Fahim, who saw me through my graduate years and was always there for me, "muchas gracias, amigo"; Linda van Rens, my loyal lab partner, for supporting me at all times and for your contributions to this document; and everyone else who supported me and wondered why I was still in school, now you are holding the reason in your hands.

Special thanks also go to my cousins from Floyd County, Benito and Irma Pastrana, for their love and affection, and for accepting me as another son. To my family — Pepe, Dalila, Chelo, and Rafi — who always believed in me… we made it! And to God, who blessed me with such a great family… without You my victories are failures!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1:
INTRODUCTION

1.1 PROBLEM STATEMENT

Although existing lab-based formative evaluation is frequently and effectively applied to improving usability of software user interfaces, it has limitations. Project teams want higher quality, more relevant, usability data – more representative of real world usage. The ever-increasing incidence of users at remote and distributed locations (often on the network) precludes direct observation of usage. Further, transporting users or developers to remote locations can be very costly. As the network itself and the remote work setting have become intrinsic parts of usage patterns, the users' work context is difficult or impossible to reproduce in a laboratory setting. These barriers led to extending usability evaluation beyond the laboratory to the concept of remote usability evaluation [*], typically using the network itself as a bridge to take interface evaluation to a broad range of users in their natural work settings.

Perhaps the most significant impetus for remote usability evaluation methods, however, is the need for a project team to continue formative evaluation downstream, after implementation and deployment. Most software applications have a life cycle extending well beyond the first release. The need for usability improvement does not end with deployment, and neither does the value of lab-based usability evaluation, although it does remain limited to tasks that developers believe to represent real usage. Fortunately, deployment of an application creates an additional source of real-usage usability data. However, these post deployment usage data are not available to be captured locally in the usability lab. Thus, the need arises for a remote capture method.

In this regard, post-deployment evaluation often brings to mind alpha and beta testing, but these kinds of testing usually do not qualify as formative usability evaluation. Typical alpha and beta testing in the field (Nielsen, 1993; Rubin, 1994) is accomplished by asking users to give feedback in reporting problems encountered and commenting on what they think about a software application. This kind of post hoc data (e.g., from questionnaires and surveys) is useful in determining user satisfaction and overall impressions of the software. It is not, however, detailed data observed during usage and associated closely with specific task performance – the kind of data required for formative usability evaluation.

[*] The term "usability testing" is often used to refer to the evaluation of the user interaction design of an application. However, users might misinterpret this term and think that they are being evaluated and not the application. For that reason, this study uses instead the term "usability evaluation".

1.2 RELEVANCE OF CRITICAL INCIDENT DATA

This detailed data, perishable if not captured immediately and precisely as it arises during usage, is essential for isolating specific usability problems within the user interaction design. This is exactly the kind of data one obtains from the usability lab, in the form of particular critical incident data and usability problem descriptions. In real world task performance, users are perhaps in the best position to recognize critical incidents caused by usability problems and design flaws in the user interface. Critical incident identification is arguably the single most important kind of information associated with task performance in a usability-oriented context.

1.3 RESEARCH GOALS

1.3.1 Overall goal of the project

Because of this vital importance of critical incident data and the opportunity for users to capture it, the over-arching goal of this work is to develop and evaluate a remote usability evaluation method for capturing critical incident data and satisfying the following criteria:

- tasks are performed by real users,
- users are located in normal working environments,
- users self-report own critical incidents,
- data are captured in day-to-day task situations,
- no direct interaction is needed between user and evaluator during an evaluation session,
- data capture is cost-effective, and
- data are high quality and therefore relatively easy to convert into usability problems.

Several methods have been developed for conducting usability evaluation without direct observation of a user by an evaluator (see Section 3.2 entitled "Types of remote evaluation methods"). However, none of these existing remote evaluation methods (nor even traditional laboratory-based evaluation) meets all the above criteria. The result of working toward this goal is the user-reported critical incident method, described in this thesis.

1.3.2 Steps and objectives

The over-all goal of developing and evaluating a new method for remote usability evaluation is comprised of several steps, each representing a substantial project on its own:

1. feasibility case study to explore relevant issues, develop the operative research questions;
2. development of the user-reported critical incident method;
3. extensive exploratory study of the method to gain understanding and insight about the method;

4. controlled laboratory-based experiments validating research hypotheses; and
5. field studies conducted in real work environments, accounting for work context factors (e.g., noise, interruptions, multi-thread task performance).

Steps 1, 2, and 3 comprise the work completed for this thesis. The first step, the case study described in Chapter 5, was reported in Hartson, Castillo, Kelso, Kamler, and Neale (1996), where the method was called semi-instrumented critical incident gathering. The objective of this step was to judge feasibility of the method. Based on the insights gained from the case study, in the second step a new method for conducting remote usability evaluation was developed called the user-reported critical incident method. Step 3 is an in-depth exploratory study reported in Chapters 4, 6, and 7.

Step 4 is a formal study to compare the effectiveness in producing usability problem descriptions of the user-reported critical incident method for remote evaluation to traditional usability laboratory-based evaluation. Originally, there was hope to include at least part of step 4 in this thesis work, but it became clear, as work progressed, that steps 1, 2, and 3 each constituted a full project in their own right, and, thus, steps 4 and 5 are reserved for future work.

The exploratory study in step 3 was performed to gain insight and understanding (under practical constraints) about the strengths and weaknesses of the method. Since step 3 is the primary focus of this thesis work, it deserves a more detailed discussion here. In particular, the objectives were:

Objective 1: Investigate feasibility and effectiveness of employing users to identify and report their own critical incidents during usage.

Objective 2: Investigate feasibility and effectiveness of transforming remotely gathered critical incident data into usability problem descriptions.

Objective 3 Gain insight into various parameters associated with the user-reported critical incident method.

Each of these objectives, respectively, maps to a research question of the study, discussed in detail in Chapter 7.

Objective 1 translates to the following research question: Can users report their own critical incidents and how well can they do it? In this study, Objective 1 is divided into the following sub-objectives:

1. Explore issues about user-subject performance in identifying and reporting critical incidents:
    - User-subject ability to identify and report critical incidents during task performance
    - User-subject activity sequencing and timing in reporting critical incidents
    - Level of time and effort required to report critical incidents

- User-subject ability to rate severity of critical incidents
- User-subject ability to identify high severity critical incidents as well as low and medium severity critical incidents

2. Obtain subjective data about user-subject perceptions, preferences, and attitudes towards remotely-reporting critical incidents:
    - User-subject attitudes towards remotely-reporting critical incidents
    - User-subject preferences with respect to reporting critical incidents anonymously
    - User-subject perceptions with respect to interference with user tasks
    - User-subject preferences relating to reporting negative and positive critical incidents

Objective 2 translates to the following research question: Can evaluators use critical incident data to produce usability problem descriptions and how well can they do it? In this study, Objective 2 is divided into the following sub-objectives:

- Ability of evaluator-subjects to analyze critical incident data
- Role of textual reports in data analysis
- Role of video in data analysis
- Role of audio in data analysis
- Time and effort required to analyze critical incident data
- Level of agreement with user-subject critical incident severity ratings

Objective 3 translates to the following research question: What are the variables and values that make the method work best? In this study, Objective 3 is divided into the following sub-objectives:

- Preferred location for the critical incident reporting tool
- Using the *Remote Evaluation Report* window
- Role of training in reporting critical incidents
- Verbal protocol issues
- Role of audio in reporting critical incidents
- Role of video in the study
- Packaging critical incident data
- Issues relating to the proximity of a critical incident and its cause

1.4 CONTRIBUTION OF RESEARCH

This thesis reports on development of a new technique for conducting remote usability evaluation. Called the user-reported critical incident method, this new technique was designed to meet all the criteria stated in Section 1.3.1. With this method, real users are located in their own working environment, working on everyday tasks, and reporting their own critical incidents without direct interaction with an evaluator.

An overview of the user-reported critical incident method is illustrated in Figure 1-1.

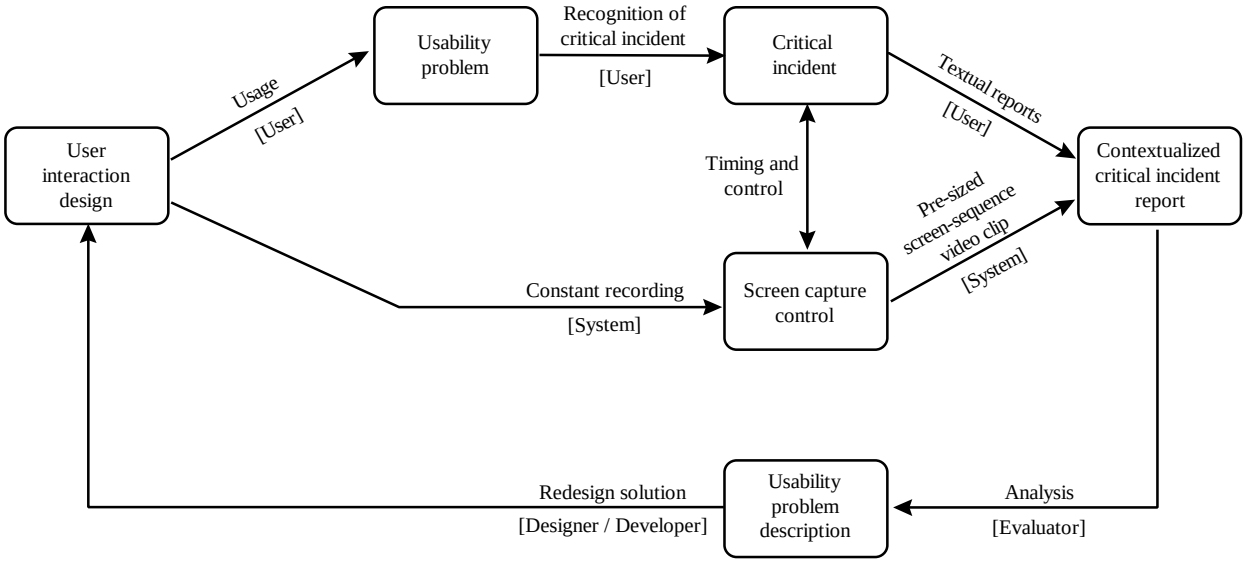


Figure 1-1. Overview of the user-reported critical incident method

Users identify and report critical incidents occurring during use. The user's computer system augments critical incident reports with task context in the form of screen-sequence video clips. Evaluators analyze these contextualized critical incident reports to create a list of usability problem descriptions. Outside the scope of this evaluation method, designers utilize usability problem descriptions to feed redesign solutions back to the interaction design.

Because the captured data are centered around critical incidents during task performance, they are concise and have the potential to eliminate the usual evaluator time and effort of separating significant data (i.e., critical incidents) from the total flow of events in a user work session. Moreover, because the user identifies the critical incidents, and the user has the most background knowledge of the problem, high quality incident reports are more likely, assisting conversion into usability problem descriptions.

1.5 OVERVIEW OF THESIS

Following a review of related work in Chapter 2, a description of remote evaluation methods appears in Chapter 3. Chapter 4 contains a description of the user-reported critical incident remote usability evaluation method, followed by a description of an earlier case study of this method in Chapter 5. Chapter 6 reports a qualitative study to evaluate the method, leading to a discussion of results and lessons learned in Chapter 7.

CHAPTER 2:
RELATED WORK

2.1 TRADITIONAL LABORATORY-BASED USABILITY EVALUATION

Traditional laboratory-based usability evaluation is included in this study as a class of benchmark methods for comparison with remote evaluation methods. Lab-based evaluation is usually considered "local evaluation" in the sense that user and evaluator are in the same or adjacent rooms at the same time (Figure 2-1). This kind of usability evaluation, conducted in a usability laboratory, is quite formal, using predefined tasks and driven by quantitative usability specifications (Whiteside, Bennett, and Holtzblatt, 1988). Sessions are usually videotaped for backup and review.

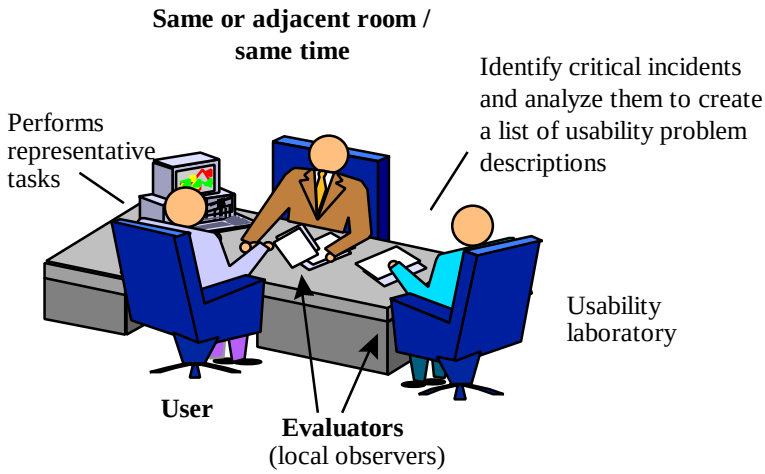


Figure 2-1. Scenario of a typical laboratory-based usability evaluation session

Data collected are both quantitative (e.g., task performance, time and error count, user preference questionnaire scores) and qualitative (e.g., critical incident descriptions and verbal protocol). Typically, quantitative data are used to determine if and when usability specifications are met so that iteration can halt. Qualitative data serve to identify usability problems, their causes within the interface design, and potential redesign solutions (Hix and Hartson, 1993).

2.2 THE CRITICAL INCIDENT TECHNIQUE

This section provides background information about critical incidents and a brief explanation of the original critical incident technique with some modern variations of the method.

2.2.1 Definition of a critical incident

Despite numerous variations in procedures for gathering and analyzing critical incidents, researchers and practitioners agree about the definition of a critical incident. 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 and Nilsson (1964) defined a critical incident as an incident "in which the holder of a position in a certain occupation has acted in a way which, according to some criterion, has been of decisive significance for his success or failure in a task." Thus, for example, in the context of safety studies, a critical incident is an event that indicates something significant about safety. In the context of formative usability evaluation, a critical incident is an occurrence during user task performance that indicates something (positive or negative) about usability.

2.2.2 Origins of the critical incident technique

The origins of the critical incident technique can be traced back to studies performed in the Aviation Psychology Program of the Army Air Forces in World War II. The technique was first formally codified by the work of Fitts and Jones (1947) for analyzing and classifying pilot error experiences in reading and interpreting aircraft instruments. Fitts and Jones used the term "errors" rather than "critical incidents", gathered data retrospectively (accounts of errors were obtained through recorded interviews and written reports from pilots), and based their analysis of human performance on significant events that occurred while a task is performed. As opposed to Fitts and Jones' way of collecting data, data gathering during task performance is now considered a defining criterion for critical incident methods.

The work of Flanagan (1954) became the landmark critical incident technique, after his article entitled "The Critical Incident Technique" appeared in the *Psychological Bulletin*. Since this publication, the critical incident technique has been thoroughly described by other researchers (Andersson and Nilsson, 1964; Meister, 1985), and has also become a common technique among human factors practitioners, though it has often been modified.

2.2.3 Relevance of critical incident data

Critical incident data, which is perishable if not captured immediately and precisely as it arises during usage, is essential for isolating specific usability problems within the user interaction design. This is exactly the kind of data one obtains from the usability laboratory, in the form of particular critical incident data and usability problem descriptions.

In real world task performance, users are perhaps in the best position to recognize critical incidents caused by usability problems and design flaws in the user interface. Critical incident

identification is arguably the single most important kind of information associated with task performance in a usability-oriented context.

2.2.4 Flanagan's original critical incident technique

Beyond this general agreement on what constitutes a critical incident, most discussion in the literature has been devoted to describing Flanagan's technique for gathering and analyzing critical incidents as well as several variations of the technique for various purposes. Flanagan (1954) originally defined the critical incident technique as a set of procedures designed to describe human behavior by collecting descriptions of events having special significance and meeting systematically defined criteria: "…only simple types of judgments are required of the observer, reports from only qualified observers are included, and all observations are evaluated by the observer in terms of an agreed upon statement of the purpose of the activity."

Prior to the critical incident technique, field observations consisted mainly of retrospective and anecdotal reports by untrained observers operating under completely uncontrolled conditions. Flanagan provided a systematic way of collecting data from a variety of simple domains while ensuring some degree of validity. Using his method, observers could gather direct observations of human behavior in a way to facilitate solving practical problems and developing psychological principles.

Flanagan envisioned trained observers, who were domain knowledgeable data collectors, making observations of ongoing activities in a user's normal working environment. The technique called for collection of critical incidents as they occurred. He generally ruled out retrospective construction of critical incident data on events that had occurred previously. He was concerned that the quantity and quality of data that could be obtained retrospectively could not support the analysis needed within the critical incident method. That would mean, for example, that reconstruction of events leading to an aircraft accident would not qualify as the critical incident method, because no critical incident data were taken at the time of the accident; all data would have been reconstructed. Presumably, however, critical incident analysis could be applied to the much richer detail of a videotape made as a task was performed.

In addition, Flanagan encouraged using the critical incident technique for both research and application-oriented work, and generalization of data so inferences could be made for the larger population. Domains cited by Flanagan (e.g., airplane cockpit, hospital operating room) were all open to intrusive data collection (decades later, as is, the domain of human-computer interfaces). In these domains, most observers were supervisors trained in critical incident observation techniques, and users were accustomed to being observed by these people. In sum, Flanagan's critical incident technique, by focusing on events occurring in real task performance, brought together more closely the operational world of users and the human factors laboratory.

2.2.5 Variations of the critical incident technique

Flanagan did not, however, see the critical incident technique as a single rigid procedure. He was in favor of modifying this technique to meet different needs as long as the original criteria (Section 2.2.3) were met. In fact, 40 years after the introduction of Flanagan's critical incident technique, Shattuck and Woods (1994) reported a study that revealed this technique has rarely been used as originally published. Instead, variations of the method were found (as Flanagan anticipated), each suited to a particular field of interest, often with constraining parameters. First, Shattuck and Woods reviewed materials commonly used to educate human factors practitioners (e.g., textbooks, handbooks, professional journals, conferences) and found that little mention was made of the critical incident technique. A survey conducted with practitioners-in-training to highlight some misconceptions concerning the critical incident technique found that many practitioners claimed to be familiar with the technique, and perhaps had even used it. But when they were asked to describe the methodology, it was often a much different version than that originated by Flanagan.

Human Factors practitioners today still find value in Flanagan's work and have continued using the critical incident technique as a basis for their methodologies, modifying it to meet their specific requirements. Shattuck and Woods (1994), in this 40 year retrospective on the critical incident technique, offer interesting details on several successful variations of the technique.

Who identifies critical incidents

One factor in the variability of the critical incident technique is the issue of who (which role) makes the critical incident identification. Flanagan originally used trained observers to collect critical incident information while observing users performing tasks. In contrast, del Galdo, Williges, Williges, and Wixon (1986) involved users to identify critical incidents during task performance. The user-reported critical incident method is similar to that of del Galdo et al. in this regard, but different in other ways that will become apparent in Chapter 4. Dzida, Wiethoff, and Arnold (1993) and Koenemman-Belliveau, Carroll, Rosson, and Singley (1994) adopt the stance that identifying critical incidents during task performance can be an individual process or a mutual process between user and evaluator.

Critical incidents and critical threads

Carroll, Koenemann-Belliveau, Rosson, and Singley (1993) and Koenemann-Belliveau et al. (1994) defined a critical incident in the context of human-computer interaction as an event that stands out during usability evaluation (e.g., a major breakdown during task performance). In that context, empirical usability evaluation, particularly formative evaluation, depends heavily on observing and interpreting critical incidents during user task performance. This process is time consuming and laborious since evaluators need to observe users, collect and analyze data, and generalize results. The authors made the case for leveraging the work done in empirical formative evaluation by generalizing and collecting data to gain more general knowledge and formulate theories about usability problems. To this end, they proposed an extension to the critical incident technique that includes critical threads. Causes of a critical incident are not necessarily found in the immediate context of its occurrence and can be distributed throughout a

user's prior experience. A set of earlier episodes, called *critical threads*, may occur, each of which in isolation may not be noteworthy but together can explain a critical incident.

In the study of Koenemann-Belliveau et al., two participants spent 12 hours learning concepts of the Smalltalk/V® object-oriented language with either the Smalltalk standard tutorial or MiTTS (Minimalist Tutorial and Toolset for Smalltalk). After the learning phase, participants spent about four hours on a series of six programming and software design projects. Learners were videotaped and asked to "think aloud" (Lewis, 1982; Nielsen, 1992; Wright and Monk, 1991) during their whole activity. Trained observers (as in Flanagan's technique) viewed the tapes and identified critical incidents learners encountered. Although evaluators identified critical incidents, Koenemann-Belliveau et al. agreed that users and/or evaluators could do the critical incident identification process.

ERGOguide, The Quality Assurance Guide to Ergonomic Software

Dzida et al. (1993) define critical incidents (or error events) as circumstances that especially lead to failure or success in an individual's action. They argue that errors should be taken seriously not only during heuristic inspections but primarily during day-to-day use of a system. They also indicate that there are two possible definitions of user errors: system-oriented errors (e.g., violating a system constraint) and user-oriented errors (e.g., blocking a user's goal achievement).

User-oriented errors are identified with the help of a user, applying specific methods for observing errors and eliciting users' background information on cognitive processes during error events. These methods may be quite extensive and labor intensive, and an evaluator facilitates the procedure by post-session interviews of users based on videotapes showing critical interaction events. Dzida et al. claimed that, by seeing the videotape, users are stimulated to recall the mental events that occurred during their interaction with the system. This technique is commonly known as retrospective verbal protocol taking (Bowers and Snyder, 1990; Ohnemus and Biers, 1993). During the interview, each user is asked to tell the experimenter what was going on "inside his or her head" during the session, as they watch the videotape. Contrary to what Flanagan presented, Dzida et al. adopted the perspective that identification of critical incidents is a mutual process between user and experimenter.

2.2.6 Software tools to help identify critical i ncidents

Human factors and human-computer interaction researchers have developed software tools to assist identifying and recording critical incident information. Three such applications are described here.

On-line critical incident tool

del Galdo et al. (1986) investigated use of critical incidents as a mechanism to collect end-user reactions for simultaneous design and evaluation of both on-line and hard-copy documentation. The method required user-subjects (not the trained observers of Flanagan's technique) to report critical incidents involving both types of documentation after completing 19 benchmark tasks. A tool was designed to collect critical incidents from user-subjects. An experimenter monitored all subject activity and prompted subjects when they neglected to report incidents (i.e., sending a one-line message stating: "Please remember to report all documentation incidents.")

The on-line critical incident tool consisted of three components: format of questions for data collection, procedure for data collection, and summary of incident data into lists of interface problems and assets. The format of questions for data collection included both open-ended and checklist style questions to include a description of the incident, an outcome classification of the incident (success or failure), and a severity rating of the incident (extremely critical to extremely non-critical on a scale from 1 to 7).

Data collection involved presentation of an on-line questionnaire and collection of critical incident data. The task and the tool were presented simultaneously, using two terminals, one which presented the software to be evaluated, and a second terminal which presented the online tool to collect critical incident data. Creating critical incident data included data summaries examining problems contained within critical incidents, and prioritizing incidents by frequency of occurrence within their outcome categories.

IDEAL

IDEAL (Ashlund and Hix, 1992; Hix and Hartson, 1994) is an interactive tool environment that supports user interaction development activities, especially focusing on formative evaluation. IDEAL is an integrated environment comprised of several different tools, each of which supports a different activity in the user interaction development process, and users of IDEAL are experienced usability evaluators. A typical configuration for IDEAL consists of a DECstation 5000/133, two Sony video monitors, two Sony Hi-8 VISCA-controlled videotape decks, a headset microphone for an evaluator to speak with a user and/or record comments on videotape, a video camera for recording a user's face, a scan converter for recording screen actions, and a lapel microphone for recording a user's voice on videotape (the evaluator can also monitor this microphone so that the two microphones also function as an intercom).

IDEAL supports:

- capture of evaluation session information such as user identification code, date, trial number, task description, associated usability specifications;
- control of recording and playback of both tape decks; and
- creation and editing of critical incident records, each tagged to videotapes by time codes.

The feature of tagging critical incidents on the tapes allows evaluators to go directly to each critical incident for analysis without having to view tapes between critical incidents.

UCDCam

Researchers at IBM in Toronto developed a software system called UCDCam (Hameluck and Velocci, 1996), based on Lotus® ScreenCam™. This application, running in Windows3.1 ™, is used for capturing digitized video of screen sequences during task performance, as part of critical incident reports.

When a user first activates UCDCam, the application opens a "Session Info" window to store the user name, name of users' organization, name of the product being evaluated, and the hard drive where video clips and reports would be stored. While users work with their normal tasks, UCDCam runs as a "background" process, continuously recording all screen activity in a current buffer, and it also retains a separate holding buffer that holds the two-minutes of screen activity that occurred prior to the initialization of the current buffer. To report critical incidents, users click a button that opens an editor for entering comments about the problem. Users make selections from various list-boxes to indicate what they were trying to do when the problem occurred (i.e., user task) and to rate critical incident severity. List box entries (possible choices) are configurable by the evaluator. UCDCam also counts the number of reports sent by each user.

UCDCam automatically saves a screen-sequence clip of all activity that occurred for an interval prior to clicking the button (current n-second buffer plus two-minute previous buffer if the current buffer is less than one-minute long). Approximately 200 KB is required to store one minute of screen action in the user's computer, depending on display resolution and amount of screen updates. If the user has not pressed the "Incident" button within the two-minute buffer interval, then a new current buffer is initialized, and the old current buffer is used to replace the old holding buffer. That way, the most recent interval of screen action is always captured at any point in time, with screen-clips of 1 to 2 minutes in duration. Buffer duration is static, but configurable by the evaluators (up to 20 minutes in length).

The intention is that UCDCam will automatically package a screen-sequence clip with user comments (textual report) and send this package of information via the network to evaluators. Evaluators then use UCDCam to watch the clips, analyze activity that occurred prior to when the user reported an incident, and create usability problem descriptions.

As part of this thesis research, in fact, a beta version of UCDCam was investigated. However, continuous video was used to record users during evaluation sessions to analyze some timing problems discussed in Chapter 7. This decision turned out fortunate, indeed, since much of the

important data was outside the range of the capture interval we intended to use. This choice was to meet requirements of the study and does not reflect a problem with UCDCam.

2.2.7 Adaptation of the critical incident technique in this study

Given Flanagan's critical incident technique and its variations early in this chapter, this thesis research specifies how the technique has been adapted to fit the requirements of the user-reported critical incident method for remote usability evaluation. This section defines how the term critical incident is used in this study, including a list of relevant information that must be captured for each critical incident.

Definition of a critical incident for this study

To relate the way the critical incident technique is used here to the various approaches described in the literature, a *critical incident* is defined in this study to be an observable event, either positive or negative, that stands out during user task performance and reveals something significant about a usability of the interface being evaluated. An occurrence that causes a user to express satisfaction in some way (e.g., "I like that!", "Oh, good graphics!", "I see why the button is grayed out now.") is considered a positive critical incident (Hix and Hartson, 1993). Conversely, any event that causes errors, dissatisfaction ("This Help doesn't help me!"), impact on effort, and/or task performance is considered a negative critical incident. Emphasis in this study is on identifying negative critical incidents because they usually reflect usability problems in the interaction design.

Identifying the cause of a critical incident

Although Koenemann-Belliveau et al. (1994) indicated that the cause of a critical incident could be far from its occurrence, this study is interested in remote capture of the immediate context of critical incidents (e.g., video clip of screen sequence before critical incident was identified) and investigation to see if this is sufficient for evaluators to infer usability problems. By looking at the exact beginning and ending of a critical incident, it may be possible to investigate the "distance" (e.g., in time) between immediate context and cause of the problem. If the immediate context of a critical incident does not reveal its cause, it can probably show the usability problem itself, which is what remote evaluation needs to discover. Evaluators can later analyze the entire interface and user tasks to find the cause of a critical incident.

2.3 MINIMALIST TRAINING

2.3.1 Need for critical incident training

Success of the user-reported critical incident method depends on the ability of typical users to effectively recognize and consistently report critical incidents. Users cannot be expected to be generally trained in human-computer interaction, but they can be given minimal training for the specific task of identifying and reporting critical incidents.

2.3.2 Using the minimalist approach to instructional design

During the 1980s, a group at IBM Watson Research Center developed the minimalist approach to instructional design (Carroll, 1984; Carroll, 1990). Using numerous psychological methods to study a variety of commercial systems, prototypes, and training approaches, they logged nearly 1,000 hours of one-on-one observation of learner activities, and several thousand more hours of less intensively monitored experimental studies of new user performance. In the years since its creation, minimalist instruction has received widespread attention as an effective way to develop training (van der Meij and Carroll, 1995).

The minimalist approach consists of design of instruction in a way that is effective and sensible for a learner. Designing materials requires a balance between a learner's need for knowledge (e.g., acquiring explanation information by reading manual or watching videotape) with a learner's need for immediate opportunity to act (e.g., practice with real examples from application).

2.3.3 Principles for minimalist training

To develop minimalist instructions, practitioners apply a set of principles and their corresponding heuristics (van der Meij and Carroll, 1995) as illustrated in Figure 2-2. These principles are not applied as a strict set of rules, but as guidelines for instructional design.

Principle 1: Choose an action-oriented approach

Heuristic 1.1: Provide an immediate opportunity to act

Heuristic 1.2: Encourage and support exploration and innovation

Heuristic 1.3: Respect the integrity of the user's activity

Principle 2: Anchor the tool in the task domain

Heuristic 2.1: Select or Design Instructional Activities that are Real Tasks

Heuristic 2.2: Components of the Instruction should Reflect the Task Structure

Principle 3: Support Error Recognition and Recovery

Heuristic 3.1: Prevent mistakes whenever possible.

Heuristic 3.2: Provide error-information when actions are error-prone or when correction is difficult..

Heuristic 3.3: Provide error-information that supports detection, diagnosis, and recovery.

Heuristic 3.4: Provide on-the-spot error-information.

Principle 4: Support Reading to Do, Study, and Locate

Heuristic 4.1: Be brief; don t spell out everything

Heuristic 4.2: Provide closure for chapters

Figure 2-2. Four major principles and heuristics for designing minimalist instruction (van der Meij and Carroll, 1995)

Training was expected to be necessary for remote users to effectively identify their own critical incidents during task performance. After considering various training techniques, minimalist instruction principles were applied to design the training for user-subjects for the specific task of identifying and reporting critical incidents.

CHAPTER 3:
REMOTE USABILITY EVALUATION

3.1 DEFINITION

Remote evaluation is defined as usability evaluation where evaluators are separated in space and/or time from users (Hartson et al., 1996). For consistency of terminology throughout this thesis, the term *remote*, used in the context of remote usability evaluation, is relative to the developers and refers to users not at the location of developers. Similarly, the term *local* refers to location of the developers.

Sometimes developers hire outside contractors to do some usability evaluation in a usability laboratory at the contractor's site. Neither term (local or remote) per the above definitions, applies very well to these third-party consultants, but they (as surrogate developers) could have remote users.

In traditional laboratory-based usability evaluation, users are observed directly by evaluators. However, remote and distributed location of users precludes the opportunity for direct observation in usability evaluation. Therefore, with remote evaluation (Figure 3-1), the network serves as a bridge between users and evaluators, taking interface evaluation to a broad range of networked users (e.g., representative population of users) in their natural work settings.

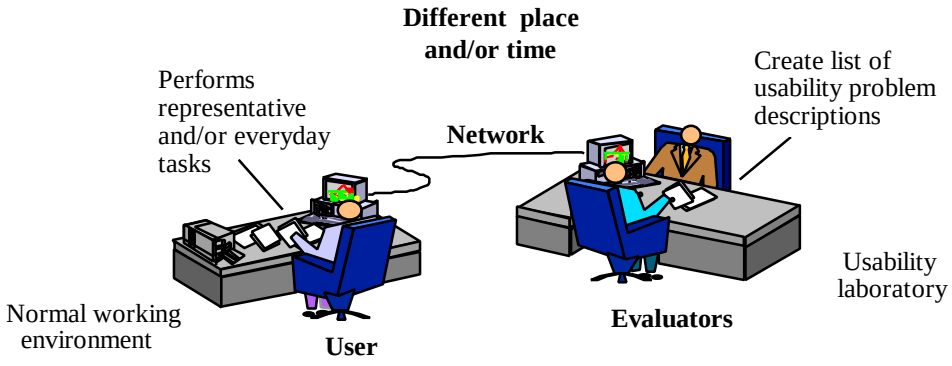


Figure 3-1. Scenario of a remote usability evaluation session

3.2 TYPES OF REMOTE EVALUATION METHODS

Identifying and distinguishing various approaches to remote evaluation is useful to understand the range of possibilities and to avoid comparison of unlike methods. No claim is made for the completeness of this list of types.

- Remote questionnaire or survey
- Live or collaborative remote evaluation
- Instrumented or automated data collection for remote evaluation
- User-reported critical incident method
- Commercial usability services
  - Third-party laboratory evaluation
  - Third-party usability inspection

Some remote evaluation situations call for a portable usability evaluation unit, by means of which the laboratory is taken to users in their normal work environment. Portable units often contain special usability evaluation equipment, including a laptop computer, video, and audio equipment. While this is an area of growing interest, it is outside the scope of this work.

3.2.1 Remote questionnaire or survey

Without any added instrumentation, evaluators can send questionnaires (Fleming, Kilgour, and Smith, 1993) to users (via mail or email) or can give users the URL to a Web-based questionnaire. In an approach more directly associated with usage, software applications can be augmented to trigger the display of a questionnaire to gather subjective preference data about the application and its interface. Appearance of the questionnaire, requesting feedback related to usability, is triggered by an event (including task completion) during usage. Responses are batched and sent to the developers.

As an example, the User Partnering (UP) Module® from UP Technology® (Abelow, 1993) uses event-driven triggers to "awaken" dialogues that ask users questions about their usage. Approaches based on remote questionnaires have the advantage that they capture remote user reactions while they are fresh, but they are limited to subjective data based on questions pre-defined by developers or evaluators. Thus, many of the qualitative data normally acquired in laboratory evaluation (i.e., the data directly useful in identifying specific usability problems) are lost.

3.2.2 Live or collaborative remote evaluation

In collaborative usability evaluation via the network (Hammontree, Weiler, and Nayak, 1994), evaluators (at a usability laboratory) and remote users (at their natural work setting) are connected through the Internet and/or a dial-up telephone line, using commercially available teleconferencing software as an extension of the video/audio cable (Hartson et al., 1996). Typical tools of this kind (e.g., Microsoft® NetMeeting™ , Sun Microsystems® ShowMe™), support several features including:

- real-time application sharing (e.g., the capability to open a word processor at the user's computer and share it with the evaluator, even when the evaluator does not have that application installed, such that the user and evaluator take turns to modify a document);
- audio conferencing links for user and evaluators to talk during the evaluation session (e.g., audio could be sent via an Internet telephone tool or regular telephone);
- shared whiteboard with integrated drawing tool for viewing and editing graphic images in real time (e.g., pointing out specific areas by using a remote pointer or highlighting tool, or taking a "snapshot" of a window and then pasting the graphic on whiteboard surface); and/or
- file transfer capabilities (e.g., the capability of a user to transfer a document including snapshots of the user interface being evaluated to the evaluators' computer).

Using video teleconferencing software as a mechanism to transport video data in real time over the network, perhaps, comes the closest to the effect of local evaluation. Currently, the primary obstacle to this approach is the limited bandwidth of the network, occasioning communication delays and low video frame rates. However, evaluators are not confined to use teleconferencing software since many software tools support the additional collaborative features mentioned above.

TeamWave Software Ltd. (1997) is a collaborative tool that supports both synchronous and asynchronous collaboration for a work group. As in the case of real-life team meeting rooms, a shared space or virtual room may contain several objects (e.g., documents, screen shots) relevant to the team's work. The user and evaluator can be present in the room at the same time, so they are working together synchronously in real-time, or they may be working in the room at different times, so the user can leave things in that room for the evaluator to analyze asynchronously. This type of asynchronous collaboration gives the evaluator the flexibility to do the evaluation all at once or at a time that is convenient.

3.2.3 Instrumented or automated data collection for remote evaluation

An application and its interface can be instrumented with embedded metering code to collect and return a journal or log of data occurring as a natural result of usage in users' normal working environments. Data captured by such applications (e.g., the WinWhatWhere ™ family of products) represent various user actions made during task performance, are often very detailed, and include logging of:

- program usage (e.g., which programs users utilize most frequently),
- project time (e.g., time spent working on specific application),
- Internet usage (e.g., monitor use of on-line time),
- comments to the system,
- keystrokes and mouse movements, and
- any other activity, producing custom-built reports.

The logs or journals of data are later analyzed using pattern recognition techniques (Siochi and Ehrich, 1991) to deduce where the usability problems have occurred. This approach has the advantage of not interfering at all with work activities of the user and can provide automated usability evaluation for certain kinds of usability problems. However, for formative evaluation it can be difficult to infer some types of usability problems effectively. This method has been used successfully for summative evaluation, marketing trials, and beta evaluation.

As an example, the ErgoLight™ family of products (ErgoLight™ Usability Software, 1997), used to enhance the usability of Windows™ applications, provides detailed and statistical information regarding a variety of usability problems. ErgoLight™ records user actions when operating the application, identifies the points of user confusion, and prompts the user to store records of his/her intention at these points. Based on the correlation between the user intention and the actual actions, ErgoLight™ extracts information, which is useful for understanding the circumstances of user confusion. The extracted information includes reports on problems in the user documentation and Help system, statistics on user confusion, and backtracks analysis of these circumstances. ErgoLight™ also integrates the data collected at the testing stage with the evaluator's comments, classifies the knowledge by organizational roles and provides reports adequate for changing the user model, the user interface design, and user documentation.

3.2.4 User-reported critical incident method

This method applies selective data collection triggered directly by users while performing tasks in their normal work context (Hartson et al., 1996). Users are trained to identify critical incidents and report specific information about these events (e.g., description and severity of problem). The reports are transmitted to developers along with context information about the user task (i.e., what user was trying to do when problem occurred) and the system itself (e.g., name and/or location of screen where problem occurred), as well as video clips containing screen-sequence actions. Evaluators use these data, approximating the qualitative data normally taken in the usability laboratory, to produce usability problem descriptions.

The user-reported critical incident method for remote usability evaluation, which is the subject of the work reported here, has potential for cost-effectiveness, since the user gathers the data and evaluators look (at least in theory) only at data that relate to usability problems.

3.2.5 Commercial usability services

A number of commercial usability evaluation services are now available to software developers. Developers use the network to communicate design documents, software samples, and/or prototypes to remote contractual evaluators, but the network is not used to connect to remote users. The evaluation is local to the contractual evaluators and remote from the developers, with results being returned via the network.

Contractual evaluation services can be a good bargain for developer groups who have only occasional need for usability evaluation and cannot afford their own facilities. However, the quality of these services can vary, the methods can be ad hoc (e.g., intuitive inspection without

use of guidelines), and the process is not always suitable for the specific needs of a development group. Two possible variations of local evaluation at remote sites are described below.

Third-party laboratory evaluation

Formal laboratory-based usability evaluation offered by consulting groups provides usability evaluation with representative users and tasks. Results include quantitative performance measures, user opinions and satisfaction ratings, recommendations for application improvement, and sometimes even a copy of evaluation session videotapes for review by the development team .

Third-party usability inspection

Some 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. Without empirical observation of users and a more formal process, results can vary depending on the knowledge and skills of the evaluators. As an example, this kind of evaluation is done by Vertical Research, Inc. (1997) for Microsoft® Windows™-based products.

3.3 CLASSIFICATION OF REMOTE EVALUATION METHODS

Remote usability evaluation encompasses several methods for evaluating user interfaces at a distant location. Since it is useful to understand how each method works under different situations, this section presents a classification that distinguishes among the characteristics of each remote evaluation method. The classifications made here are not absolute, but are relative and representative, and are based on insight gained during this study, offered to promote an intuitive comparison of the methods. The following attributes of remote usability evaluation methods are considered:

- type of users involved,
- time of evaluation,
- user location during evaluation,
- person or role who identifies critical incidents and/or problems during task performance,
- type of tasks (user's own tasks or tasks predefined by evaluator),
- level of interaction between user and evaluator,
- type of data gathered,
- type of equipment used for collecting data (e.g., videotape),
- cost to collect data,
- cost to analyze data and create usability problem descriptions, and
- quality or usefulness of collected data.

Laboratory-based usability evaluation is included in each classification as a benchmark method for comparison with the remote evaluation methods.

3.3.1 Types of users involved

The diagram in Figure 3-2 characterizes typical users involved in each type of remote evaluation method presented in Section 3.2. In the user-reported critical incident method, remote questionnaire, and instrumented or automated data collection, only real users participate in remote evaluation, as seen in the left of Figure 3-2. Live or collaborative evaluation and third-party laboratory evaluation (center of the figure) can use both real and/or representative users during remote evaluation. Finally, third-party usability inspection is the only method that does not involve users during evaluation (evaluator performs intuitive or heuristic inspection of the user interface).

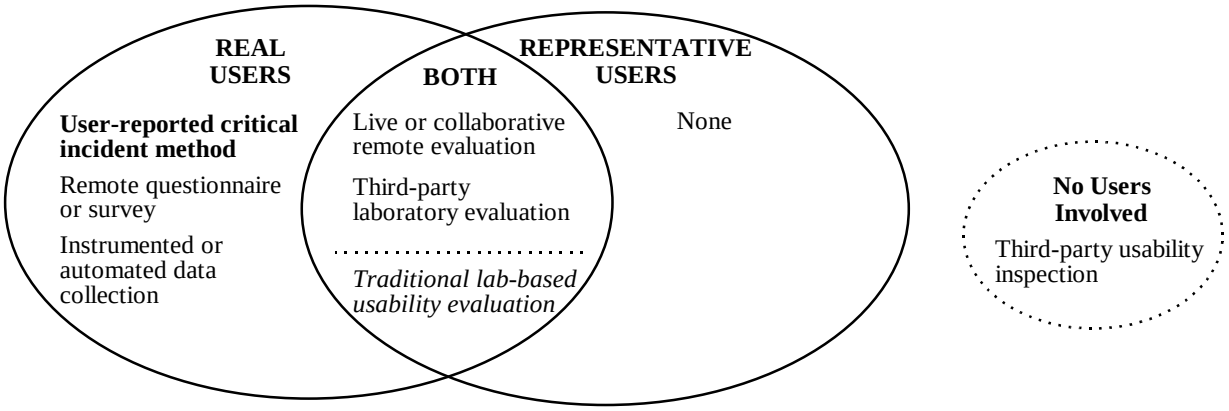


Figure 3-2. Characterization of types of users typically involved in remote evaluation

3.3.2 Time and user location

Johansen (1988) distinguished several different situations of group work along the dimensions of time and place. Adapting Johansen's work, Figure 3-3 presents different situations that can occur during task performance with remote evaluation along the dimensions of time and user location (place).

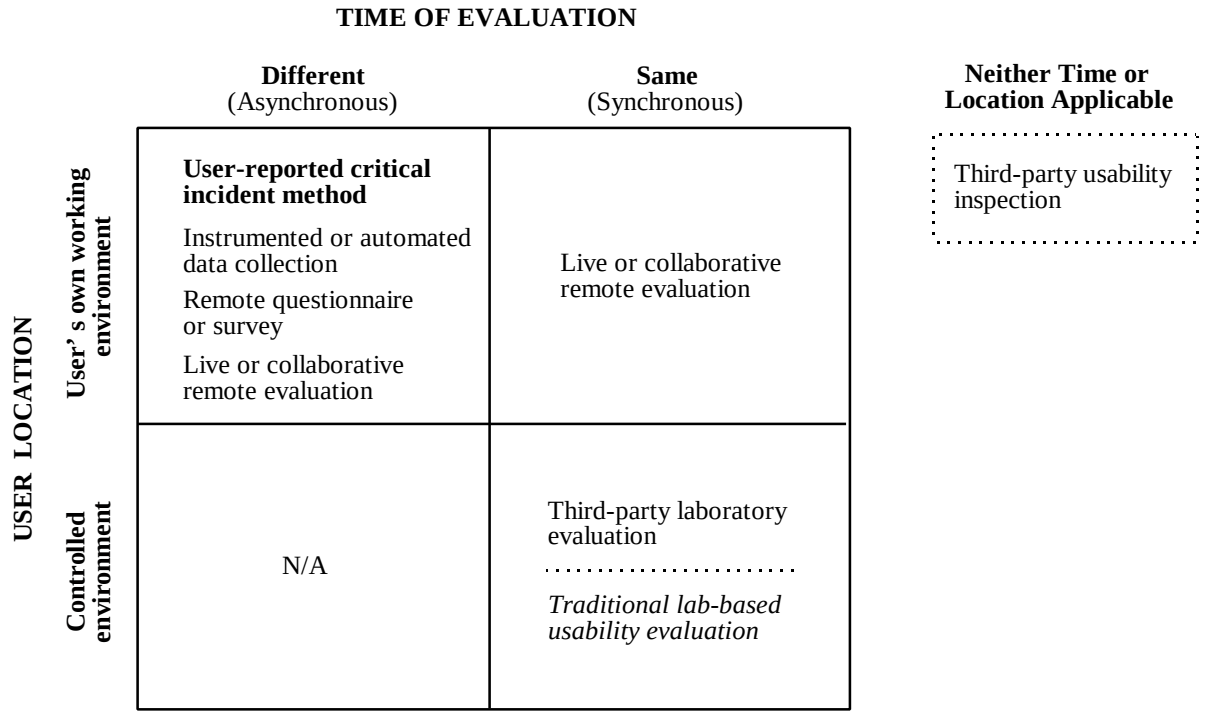


Figure 3-3. Characterization of user location and time of evaluation

In the user-reported critical incident method, instrumented data collection, and remote questionnaire, remote evaluation occurs within the user's normal work setting. Time of evaluation is asynchronous, meaning that evaluators participate at a different time than do users. User critical incident reports, log files, and questionnaires, respectively, are collected during usage, and evaluators later analyze them to create a list of usability problem descriptions. Third-party laboratory evaluation involves local evaluation with representative users at the third-party contractor/evaluator's usability laboratory (i.e., controlled environment). As in traditional lab-based usability evaluation (same time/controlled environment quadrant), users and evaluators are located in the same place and time.

Because of the generality of the concept, live or collaborative remote evaluation is the only method than spans across more than one quadrant. Generally, collaborative remote evaluation occurs when user and evaluator are connected in real-time (i.e., synchronously) over the network. When using teleconferencing software, user and evaluator are examining the user interface at the same time from different locations (e.g., adjacent rooms or distant geographical locations), where the video and audio cable (or wireless or network communication) operates as an extension of the usability laboratory. The evaluator can also use software (e.g., Compaq® Carbon Copy™) to remotely control the user's computer to observe the user interface. In the asynchronous mode for this method, the user can leave information (e.g., snapshot of one screen of the user interface) in a place known by the evaluator (e.g., screens directory), and the evaluator later connects to the user's computer to retrieve the information. Third-party usability inspection is not included in

the table on the left of the figure because it does not involve users during evaluation of the user interface.

In the current research, the "different time / user's own working environment" quadrant was of most interest because the "different time" characteristic means the evaluator is not present, making the cost of data gathering very low (relative to traditional lab-based usability evaluation). The "user's working environment" characteristic can yield more realistic qualitative data, since occurs within the user's natural work setting.

3.3.3 Person who identifies critical incidents and problems during task performance

The diagram in Figure 3-4 presents a characterization of the person who identifies and/or reports critical incidents and/or usability problems during evaluation of the user interface.

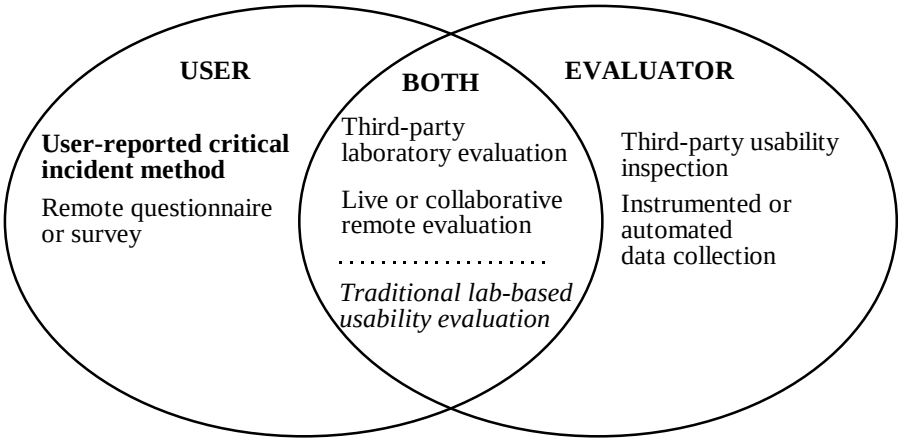


Figure 3-4. Characterization of person who identifies and/or reports critical incidents and/or usability problems during task performance

In the user-reported critical incident method, the user both identifies and reports critical incidents during task performance and the evaluator analyzes these reports to create a list of usability problem descriptions. When answering remote questionnaires or surveys, users provide subjective data and may indicate problems that they encounter during task performance. Collaborative remote evaluation and third-party laboratory evaluation involve both the user and the evaluator (trained in human-computer interaction and usability methods) to identify critical incidents and/or usability problems during task performance. For instrumented data collection, the evaluator analyzes log files of usage data and deduces possible usability problems encountered by users. Finally, third-party usability inspection exclusively involves only the evaluator in identifying and reporting critical incidents during task performance.

3.3.4 Type of tasks and level of interactio n during remote evaluation

The diagram shown in Figure 3-5 characterizes various situations that can occur along dimensions for type of tasks and level of interaction between users and evaluators during remote evaluation. Unlike Figure 3-3, none of the remote evaluation methods span quadrants of this table.

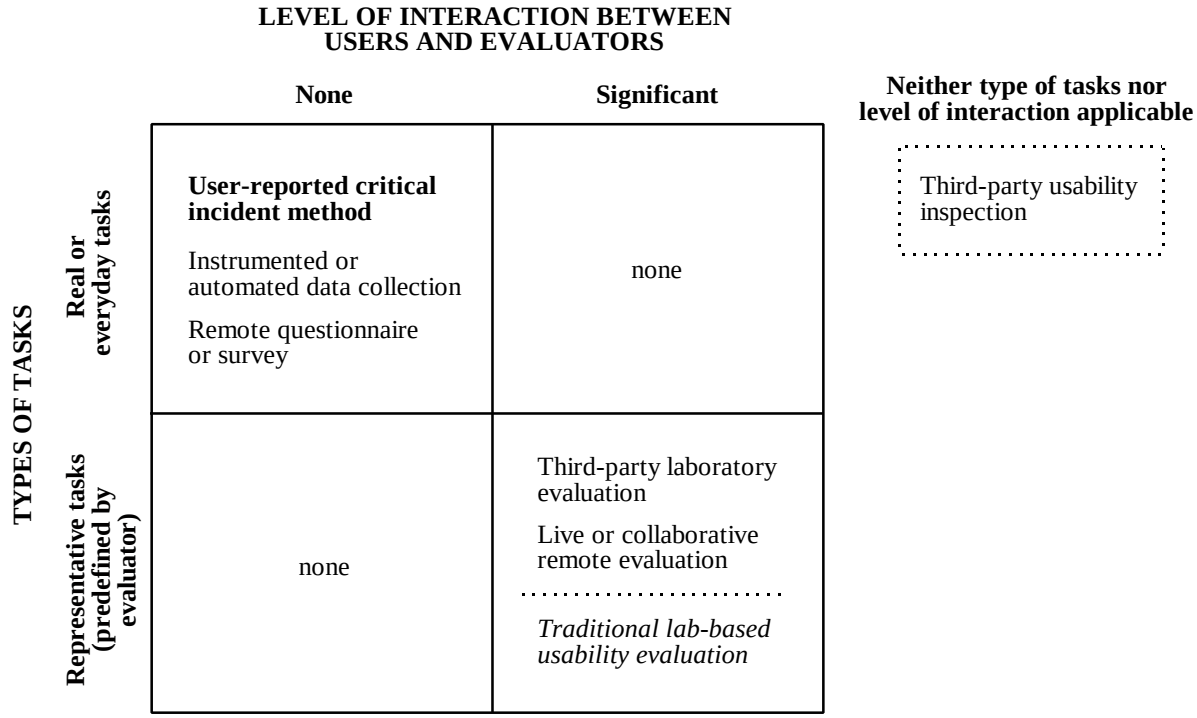


Figure 3-5. Characterization of types of tasks performed by users and level of interaction between users and evaluators during usability evaluation

The top left quadrant of this table contains methods (user-reported critical incident method, instrumented data collection, and remote questionnaire or survey) in which remote evaluation occurs while users perform real or everyday tasks, with no interaction with evaluators during task performance and/or evaluation. Similar to laboratory-based usability evaluation, the lower right quadrant includes remote evaluation methods (third-party laboratory evaluation and collaborative remote evaluation) in which users generally perform representative tasks (predefined by evaluator) and significantly interact with evaluators during usability evaluation. Again, third-party usability inspection is located outside the table on the left of the figure because this method does not involve users during evaluation of the user interface.

3.3.5 Types of data gathered

Table 3-1 shows the various types of data gathered within each remote evaluation method. With the exception of instrumented data collection, remote questionnaires or survey, and third-party usability inspection, most remote evaluation methods typically gather three or more types of qualitative data. A check in a cell of Table 3-1 indicates that the corresponding method has the capability to gather the associated type of data, but does not indicate that this kind of data is always used or can be used at low cost. More types of data involved typically imply more equipment required for data gathering. On the other hand, for evaluators, gathering different types of data may be important to ensure that the most useful data are captured (e.g., context information about critical incidents). For example, in the case of collaborative remote evaluation, evaluators can obtain several different kinds of data (e.g., videotape showing the user's face and including audio of user comments, videotape of screen action recorded via scan converter, evaluator notes taken during evaluation session, subjective information obtained from an online satisfaction questionnaire). In contrast, instrumented methods yield only logs of user actions and commands, and the remote questionnaire method yields only subjective opinion of users.

Table 3-1. Types of data gathered by each remote evaluation method

	Continuous videotape of entire session	**Audio with user comments**	**Evaluator notes from evaluation session**	**Log files, automatically recorded**	**Data obtained from collaborative work**	**Critical incident data (reports from users)**	**Critical incident data (video clips)**	**Subjective data from user questionnaire**
Remote questionnaire or survey								√
Live or collaborative remote evaluation	√	√	√		√		√	√
Instrumented or automated data collection				√				
User-reported critical incident method		√				√	√	√
Third-party usability inspection			√					
Third-party laboratory evaluation	√	√	√				√	√
Traditional lab-based usability evaluation	√	√	√				√	√

3.3.6 Type of equipment used and quantity of data gathered

Figure 3-6 illustrates how the equipment required to collect data and the quantity of data collected vary among remote evaluation methods.

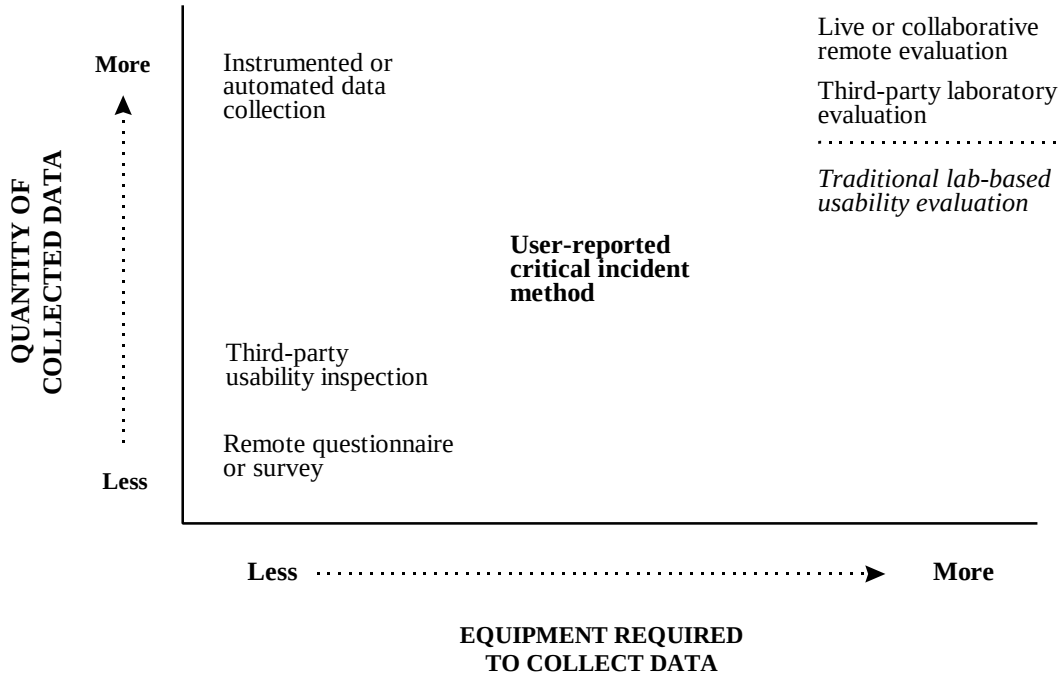


Figure 3-6. Equipment required to collect data and quantity of data gathered by each remote evaluation method

To illustrate the concept, collaborative remote evaluation, third-party laboratory evaluation, and traditional lab-based evaluation all require a relatively high level of equipment (e.g., video and audio equipment) to collect data and each provides a large quantity of data. Therefore, this group of similar methods appears in the upper right portion of Figure 3-6. In contrast, instrumented approaches require a low level of equipment (e.g., embedded metering code to collect a log of data including information such as keystrokes and mouse movements) but also yield large quantities of data. Thus, instrumented methods appear in the upper left portion of the figure.

Remote questionnaires and surveys require almost no added equipment, but produce only a small amount of data. This constraint on the amount of data stands for the need to minimize workload for users and interference with task performance. Remote questionnaires typically contain relatively few questions, not extensive surveys. Therefore, this method appears in the lower left of the figure. The user-reported critical incident method requires somewhat less total equipment and produces somewhat less data than traditional lab-based evaluation, so this method appears roughly in the middle of the figure.

3.3.7 Cost to collect and analyze data

Figure 3-7 illustrates the relationship between cost to collect data and cost to analyze it, for each remote evaluation method. Cost is limited to those incurred by an "in house" development team (mainly referring to evaluators). Developers are only "in house" personnel" and not, for example, a third party retained to perform external evaluation. Costs to the development team, which, however, include fees to third-party evaluators, include:

- equipment and technology,
- time and effort (person-hours),
- external services (third party),
- training (all, including developer and user training for evaluation),
- travel and time of user-subjects, and
- travel and time for developers to visit user sites.

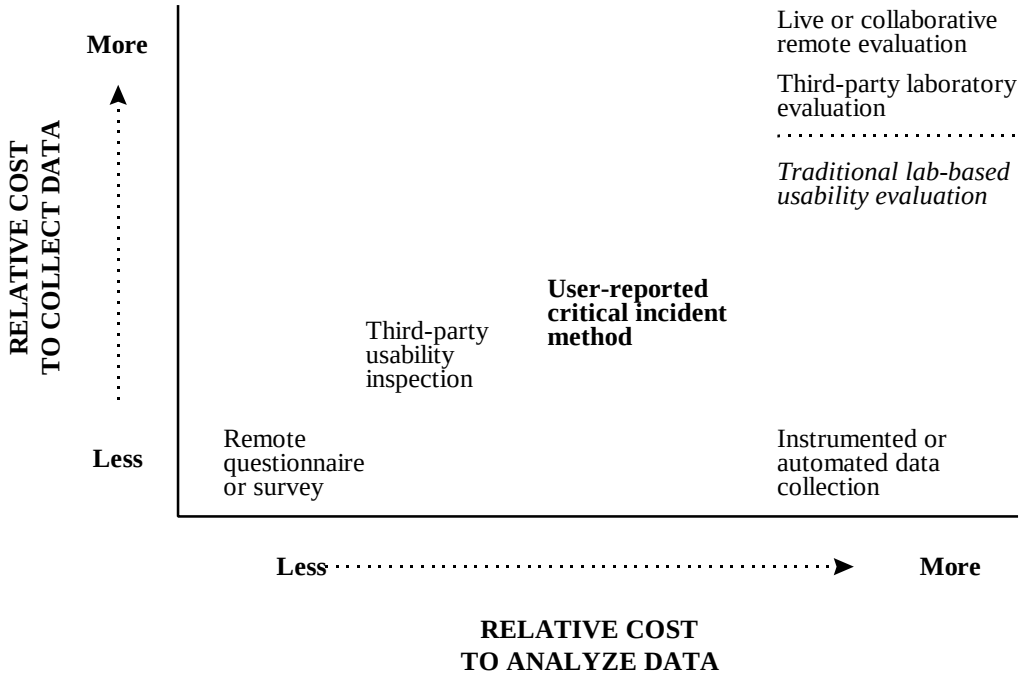


Figure 3-7. Relative costs to collect and analyze data

Remote questionnaire or survey appears again in the lower left of the figure because the cost to collect data are low and, since there is a small amount of data, the cost to analyze it is also low. The group of methods similar to traditional lab-based evaluation appear again in the upper right because the cost for both collection and analysis of data are high relative to the other methods. Instrumented data collection methods are somewhat an extreme case in that they are probably the

lowest in cost to collect data and possibly among the highest to analyze them. In the user-reported critical incident method, the cost for collecting data is higher than third-party usability inspection, but significantly lower than the traditional-lab based method. Although analysis is very similar, the cost for analyzing data for the user-reported critical incident method is lower than the lab-based methods. Interestingly, third-party usability inspection appears to make a good compromise in terms of cost when resources are limited.

Figure 3-8 shows a comparison of total data cost (i.e., collection and analysis) for a single project against quality or usefulness of data in identifying usability problems. Remote questionnaire or survey appears again in the lowest left of the figure, and once more, the group related to traditional lab-based evaluation appear in the upper right because they are the most costly but also yield the most useful data. The user-reported critical incident method is somewhat lower on each axis, costing less but producing data that are almost as useful. Third-party usability inspection appears somewhat in the middle of the graph, once again representing a possible compromise when resources are limited.

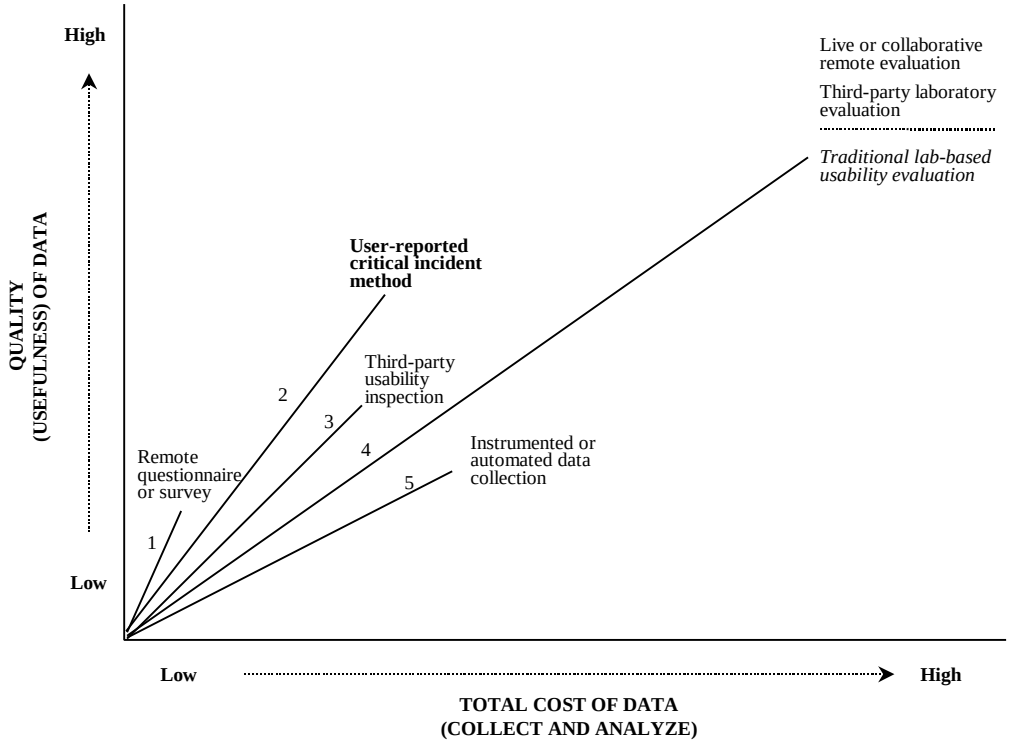


Figure 3-8. Quality of usability data by remote evaluation method

The interesting aspect of Figure 3-8[*] is the slope of the lines from the origin to each method. This slope is a measure of a kind of efficiency, namely the quotient of quality or usefulness per unit cost, of data collected by each method. In the graph, a higher slope suggests higher efficiency. Interestingly, remote questionnaire or survey has the steepest slope in the graph. Thus it has high efficiency. However, the high slope comes from a quotient from low usefulness and low cost. This limited usefulness may not alone be sufficient for the needs of the development team. On one hand, if the budget is extremely limited, some data are better than nothing. But a questionnaire only produces subjective data and, as Elgin (1995) states, "Subjective feedback is generally harder to interpret than objective feedback in a known setting…" The next most efficient method according to the graph is the user-reported critical incident method. It was, in fact, a goal of this research to develop critical incident reporting as an efficient remote evaluation method, one that was less costly than traditional lab-based evaluation but still maintained its usefulness in determining usability problems. Third-party usability inspection comes next, perhaps, again as a middle-of-the-road possibility.

Group 4, the group of remote evaluation methods related to traditional lab-based usability evaluation, appears a bit less of a bargain, despite the high usefulness of data, because of the higher cost. Because costs portrayed here are costs for one project, they include amortization of equipment, lab facilities, and training costs over several projects. Thus, the relative difference between the slopes of method 3 and the group represented by 4, and indeed the methods within group 4, can vary depending on how these fixed costs are amortized. In particular, if they are not well amortized, third-party laboratory evaluation or third-party usability inspection might be more attractive than setting up a traditional lab-based facility. Additionally, both traditional lab-based approaches and third-party usability evaluation can be quite expensive in terms of travel and time for representative users to visit the laboratory. Thus, these two methods of group 4 could have decreased slopes due to this factor.

[*] Of course, this graph is based on intuitive reasoning and represents general characteristics of the methods as qualified by the rationale set forth in this chapter. In specific instances of these methods as used by particular development groups, the results can vary.

CHAPTER 4:
THE USER-REPORTED CRITICAL INCIDENT METHOD

4.1 OVERVIEW OF METHOD

The user-reported critical incident method is a usability evaluation method that involves real users located in their own working environment, doing everyday tasks, and reporting critical incidents (after receiving minimal training) without direct interaction with evaluators. Critical incident reports are augmented with task context in the form of screen-sequence video clips and evaluators analyze these contextualized critical incident reports, transforming them into usability problem descriptions.

4.2 EVOLUTION OF METHOD

The first step in the development of the user-reported critical incident method was to conduct a feasibility case study, described in Chapter 5, with the objective of judging feasibility of the method. This study was reported by Hartson et al. (1996), where the method was called semi-instrumented critical incident gathering. Based on the insights gained from the case study, the second step was an exploratory study, to which a new method for conducting remote usability evaluation was developed called the user-reported critical incident method.

4.3 RELEVANCE OF CRITICAL INCIDENT INFORMATION

In real world task performance, users are perhaps in the best position to recognize critical incidents (specifically negative critical incidents) caused by usability problems and design flaws in the user interface (Figure 4-1). Critical incident identification is arguably the single most important kind of information associated with task performance in a usability-oriented context. This detailed data, perishable if not captured immediately and precisely as it arises during usage, is essential for isolating specific usability problems within the user interaction design. Whether gathered by users or evaluators, once critical incident information is collected, evaluators analyze it and create a list of usability problem descriptions used by interaction designers to determine design solutions and correct the problem.

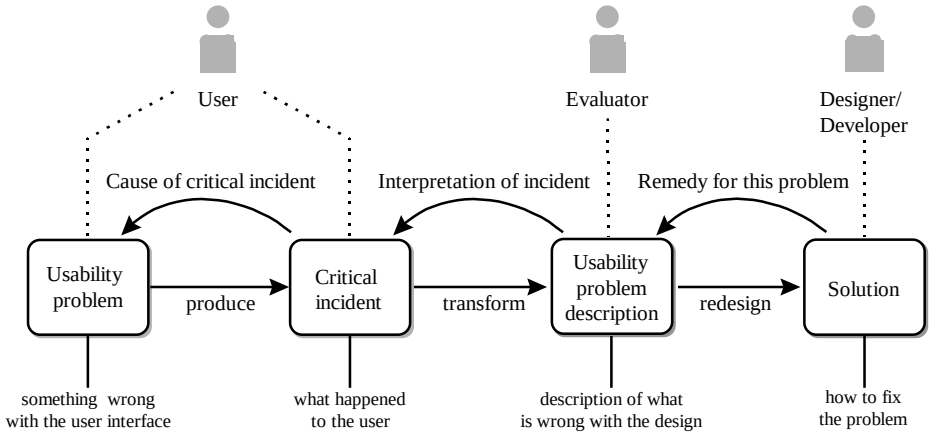


Figure 4-1. Overview of process for improving interaction design

4.4 RELEVANCE OF USER TRAINING

Success of the user-reported critical incident method depends on the ability of typical users to recognize and report critical incidents effectively, but there is no reason to believe that all users have this ability naturally. Therefore, this study was designed with the assumption that some training would be required, an expectation supported by an initial case study discussed in Chapter 5.

Fitts and Jones (1947) obtained detailed factual information about "pilot-error" experiences in reading and interpreting aircraft instruments from people not trained in the critical incident technique (i.e., an eyewitness or the pilot who made the error). In contrast, Flanagan (1954) employed trained observers (not users), who were domain knowledgeable data collectors, making observations of ongoing activities in the user's normal working environment. Laboratory-based formative usability evaluation has critical incident identification done by an evaluator trained in human-computer interaction, which might lead to skepticism about casting a user in that role. However, in fact, the critical incident technique has been successfully adapted for human-computer interaction so that critical incidents are identified by untrained users during their own task performance (del Galdo et al., 1986). Although users cannot be expected to be generally trained in human-computer interaction, the work reported here shows that users can play a more valuable role in usability evaluation if they are given minimal training for the specific task of identifying and reporting critical incidents.

4.5 THE METHOD

4.5.1 Description

As mentioned in Section 1.3.1, the user-reported critical incident method is a remote usability evaluation method for capturing critical incident data and satisfying the following criteria:

- tasks are performed by real users,
- users are located in normal working environments,
- users self-report own critical incidents,
- data are captured in day-to-day task situations,
- no direct interaction is needed between user and evaluator during an evaluation session,
- data capture is cost-effective, and
- data are high quality and therefore relatively easy to convert into usability problems.

Critical incident reports are augmented with task context in the form of screen-sequence video clips and evaluators analyze these contextualized critical incident reports, transforming them into usability problem descriptions.

4.5.2 Comparison with laboratory-based usability evaluation

Figure 4-2 depicts a view of the traditional laboratory-based formative evaluation process for comparison with the setup for the user-reported critical incident method.

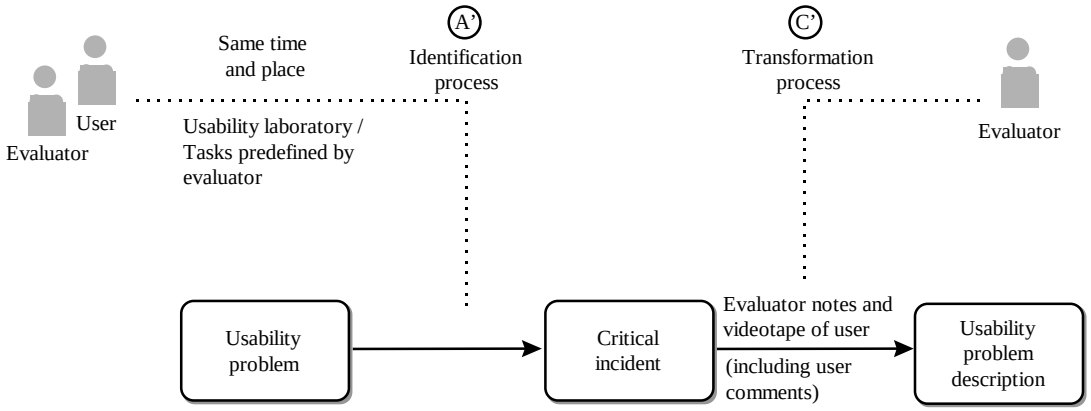


Figure 4-2. Configuration for traditional laboratory-based usability evaluation

While directly observing a user during task performance, an evaluator produces a list of critical incidents (at point A'), and later analyzes it to create a list of usability problem descriptions (at point C'). Users often help evaluators identify critical incidents by talking and explaining usage difficulties they encounter (at point A'). However, users themselves do not produce a list of critical incidents and/or report critical incidents (e.g., typing a report to describe the problem) to evaluators

4.5.3 Critical incident reporting tool

The user-reported critical incident method is illustrated in Figure 4-3. A software tool residing on the user's computer is needed to support collection of critical incident reports about problems encountered during task performance. Users are trained to identify critical incidents and use this tool to report specific information about these events.

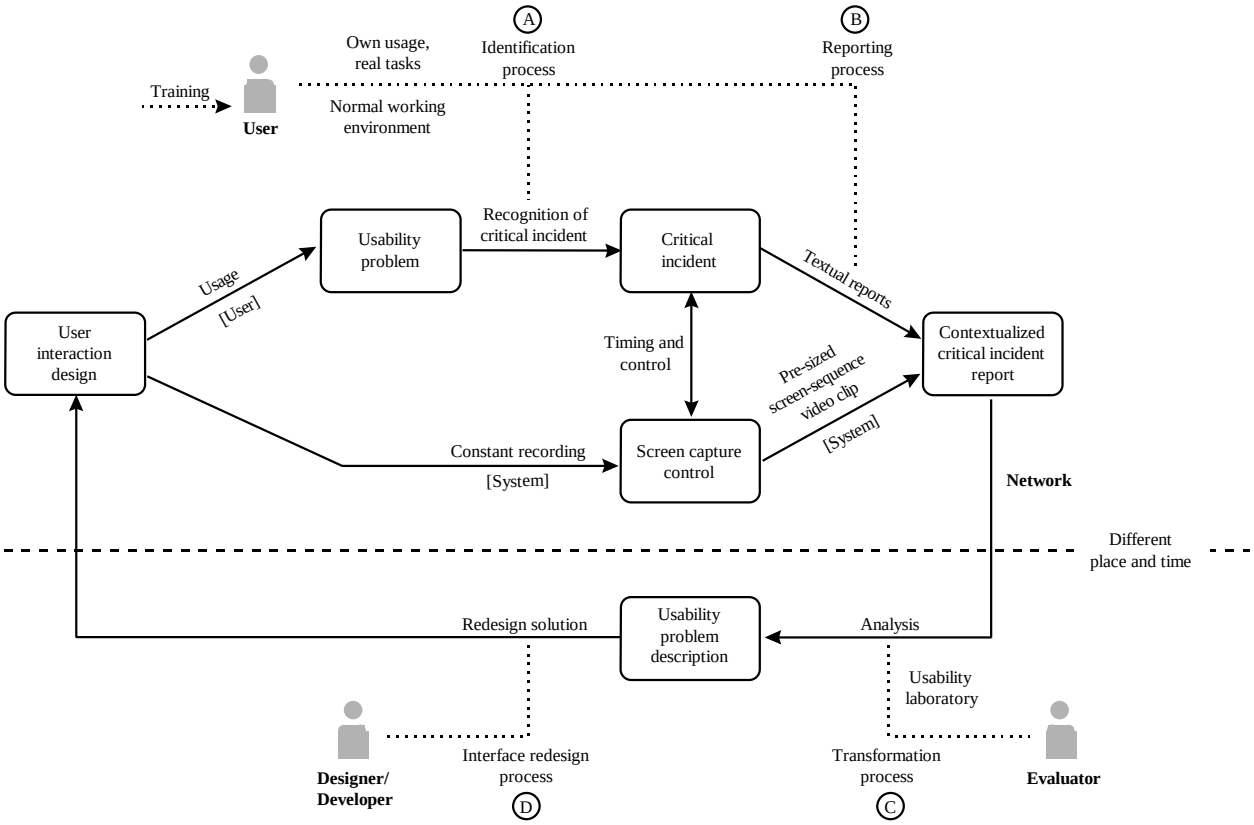


Figure 4-3. Setup for the user-reported critical incident method

Users are located in their own working environment, identifying critical incidents as they occur during the normal course of on-the-job task performance (at point A). Whenever usage difficulty is encountered, users click on a *Report Incident* button (at point B), a single interface object available from every screen of the application being evaluated. The click activates an instrumentation routine external to the application that:

- opens a textual form, in a separate window from the application, for users to enter a structured report about the critical incident encountered, and
- causes the user's computer to store a screen-sequence video clip showing screen activity immediately prior to clicking the button for the purpose of capturing the critical incident and events leading up to it[*].

This package of data, the critical incident report and the screen-sequence clip taken together, is called a contextualized critical incident report and is sent asynchronously via the network to evaluators to be analyzed into usability problem descriptions (at point C). Outside the scope of this evaluation method, designers use problem descriptions to drive redesign solutions (at point D), which go back into the interaction design.

4.5.4 Contextual factors for critical incident reports

Dzida et al. (1993) discussed various contextual factors that can be associated with a critical incident. This present study also uncovered some other similar factors, resulting in the combined list that follows:

- Specific beginning and ending of critical incident
- Screen or URL (i.e., Web location) where user encountered critical incident
- Description of user task in progress when critical incident occurred
- Expectations of user about what system was supposed to do when critical incident occurred
- Detailed description of critical incident (what happened and why user thought it happened)
- Indication of whether user could recover from critical incident and, if so, description of how user did so
- Indication of user's ability to reproduce critical incident
- Severity rating of critical incident
- Additional comments or suggested solutions to problem

[*] This describes the way the tool is intended for normal use. In order to capture more complete data in our present study, screen activity was captured continuously via a scan converter and videotape.

4.5.5 Screen-sequence video clips and timing aspects

One challenge for this study was to collect screen-sequence video clips containing meaningful information about usability problems that caused critical incidents. To ensure that the captured information was useful, it was necessary to determine:

- an effective point in time to start recording the clip, and
- an effective length of the clip.

The effective clip length is reserved for future study. In any case, the recording time of screen sequences must be of fixed length because the screen sequence capture software (e.g., IBM® UCDCam, Lotus® ScreenCam™) cannot be adaptive in the field (i.e., cannot produce video clips of varying length depending on the type of critical incident). Longer length gives a higher probability that the most useful data will be included. Shorter length allows evaluators to spend less time and effort analyzing the clip into usability problem descriptions, and requires less bandwidth to transmit each clip (e.g., at least 200 KB per minute of clip) over the network.

The user-reported critical incident method sends, as part of a contextualized critical incident report, a screen-sequence clip of action occurring just *before* the user clicks the *Report Incident* button. For that reason, a mechanism is required to ensure that this recent history of user and system actions is available for capture when the user does click the button. To accomplish this, an instrumentation routine continuously records all screen actions in a background process. When the user clicks the *Report Incident* button, the routine stores (in the user's computer) a video clip containing, say, the last two minutes of screen usage.

The chronology of events associated with a user-reported critical incident starts the moment a user begins to have difficulty with a task. At this point, the user may not yet have recognized it as a critical incident. Eventually, the user recognizes that the situation or event is a critical incident, but has not yet sent a report.

After the user is sure this event is a critical incident, and after enough is known about it to make a report, the user clicks on the *Report Incident* button and an interval of screen-sequence action occurring just prior to that point is captured as the video clip component of the report. Storage and bandwidth requirements impose economic pressure to keep the clip length short. Thus, the key factor in getting the most useful data (in terms of being able to identify the associated usability problem) is the starting point of the clip. Unfortunately, in practice the best starting point can occur over a broad range of time, and no method can be devised to automatically detect this ideal reporting point. As discovered in this study, users often wait until the most useful screen-sequence data have disappeared before they initiate a report.

A possible solution to solve this problem is to separate the act of identifying critical incidents from the act of reporting them, and to encourage users to click a button labeled to the effect of "I am just starting to recognize the occurrence of a critical incident" immediately after they recognize the critical incident (close to the effective beginning of the critical incident), but to click on the *Report Incident* button to report the incident (i.e., create a report) at a later time when

more is known about the nature of the critical incident. See Section 7.1.1 for more discussion of this difficult problem.

4.6 POSSIBLE APPLICATIONS OF METHOD

4.6.1 Early formative evaluation

Early in the software development process (i.e., after paper prototyping during iterative development), selected users can be involved in evaluating prototypes remotely. These users would identify critical incidents during performance of tasks they choose to represent their own work patterns and a set of representative tasks predefined by evaluators, while using the user-reported critical incident method to convey critical incidents to developers.

4.6.2 Alpha, beta, and other field usability evaluation

Before new software is released to the market, selected users can assist in evaluation of alpha and beta versions. Rather than the usual "Try it and let us know what you think", remote users, using the user-reported critical incident method, can send data about software bugs and critical incidents of usage that are potentially as useful as qualitative data collected in a usability laboratory.

4.6.3 Usability evaluation after software deployment

Perhaps the most significant impetus for the user-reported critical incident method is the need for a project team to continue formative evaluation downstream, after deployment. The usual kinds of alpha and beta testing do not qualify as formative usability evaluation because they do not yield detailed data observed during usage and associated closely with specific task performance. Critical incident identification is arguably the single most important source of this kind of data. Consequently, the user-reported critical incident method was developed as a cost-effective remote usability evaluation technique, based on real users self-reporting critical incidents encountered in real tasks performed in their normal working environments.

4.6.4 Customer support

After software is released to the market, the user-reported critical incident method can be utilized to assist in customer support services (e.g., help desk application). Instead of calling a customer support telephone number or sending email to developers, remote users, via this method, now have a more immediate and direct channel for feedback to developers, giving developers more detailed and specific information about systems and interface weaknesses while getting the help they need. The appropriate person or team of developers can receive the report, solve the specific problem, and email solutions back to users.

4.6.5 Marketing strategies

Subjective data (e.g., from a satisfaction questionnaire) can be broadened by marketing personnel into market-oriented product reviews revealing:

- user needs,
- new features users want added to the application,
- features that have not been used, and
- ideas to make new products.

CHAPTER 5:
FEASIBILITY CASE STUDY

5.1 GOALS OF THE CASE STUDY

Step 1 of the overall research plan of the user-reported critical incident method, stated in Section 1.4, was a case study, the primary goal of which was to judge feasibility of a user-reported critical incident method. Assessing feasibility involved

- determining if this new method could provide approximately the same quantity and quality of qualitative data that can be obtained from laboratory-based formative evaluation, and
- determining if this is possible in a way that was cost-effective for both evaluator (e.g., minimal resources for data collection and analysis) and user (e.g., minimal interference with work).

The case study, reported in detail in Hartson et al. (1996), is summarized here.

5.2 QUESTIONS ADDRESSED BY THE CASE STUDY

The case study employed two kinds of subjects, user-subjects and expert-subjects. User-subjects had no prior training in usability methods, and, as part of the study, were given minimal training to recognize critical incidents during their own usage (five minutes of lecture and a 10-minute demonstration). Expert-subjects were interface developers, who were already trained in usability methods and who performed evaluations in a usability laboratory.

Assessing feasibility, as stated in Section 5.1, translated into two research questions related to the hypotheses for the method:

1. Can user-subjects identify critical incidents approximately as well as expert-subjects?
2. How easily can expert-subjects transform contextualized critical incident data (e.g., screen-sequence clips with verbal critical incident reports in the audio) from user-subjects into usability problem descriptions?

5.3 STEPS OF THE CASE STUDY

The case study consisted of several steps:

1. Three user-subjects not knowledgeable in usability methods were trained to identify critical incidents during their own task performance.
2. Sessions of these user-subjects performing tasks and simultaneously identifying critical incidents were videotaped. The experimental application was one for viewing and manipulating images from a digital still camera.
3. A panel of three expert-subjects viewed the tapes together to detect any critical incidents missed by user-subjects.
4. The experimenters edited the tapes into sets of video clips (sequences of video with verbal reports in the audio relating to a single critical incident), each centered around the critical incident.
5. Two expert-subjects (different from the first three) analyzed the video clips, converting them into usability problem descriptions.
6. Experimenters compared the usability problem descriptions of user-subjects with those of expert-subjects.

Results of Step 3 provided answers to Question 1 (“Identifying critical incidents”)and results of Step 5 gave answers to Question 2 (“Transforming critical incidents”).

5.4 USING VIDEOTAPES AS INPUT TO EXPERT-SUBJECTS

Traditional laboratory-based evaluation uses evaluators (i.e., experts in human-computer interaction and usability methods) recording their own lists of critical incidents while observing users in the usability laboratory. For evaluation, and excluding possible interaction with the users, experts viewing complete tapes of users obtain data essentially equivalent to that obtained from observing them in real-time via video monitors in an adjacent room.

In the case study, two tapes were made simultaneously during each user subject’s task performance. One video camera was used to record user-subjects (e.g., facial expressions), including audio of user-subject comments. The second tape captured screen activity via a scan converter connected to the computer monitor. The experimenter set up the hardware and software such that user-subjects were in a controlled setting in a laboratory, with no interruptions.

It was necessary to make direct comparisons, between expert-subjects in the laboratory-based case and user-subjects in the remote case, of exactly the same critical incidents. The best way to accomplish this was for expert-subjects to view tapes of user-subjects performing tasks and encountering critical incidents, without seeing user-subjects identify the critical incidents. However, critical incident identification could not be edited from the tapes without gaps or "glitches" in the tape, thus still revealing the presence of critical incidents. The experimenter could have masked this effect somewhat by introducing additional "decoy" glitches, but it was felt that would unnecessarily interfere with the case study. This meant that expert-subjects would

have had to view the tape of each entire usability session with user-subjects identifying their own critical incidents. The expert-subjects could not ignore this aspect of user-subject behavior. Thus, the study settled on using expert-subjects to judge user-subject performance in critical incident identification. It was felt that this produced results essentially equivalent to a direct comparison, especially for the most important case of critical incidents not found by user-subjects.

To address Question 1 (“Identifying critical incidents”), the three expert-subjects viewed unedited tapes on a pair of monitors, watching user-subjects identify their own critical incident as they performed tasks, especially looking for critical incidents user-subjects failed to identify. Anything considered by a user-subject to be a critical incident was deemed so, by definition. Thus, expert-subjects did not look for any "false positive" identifications.

Incidentally, the experimenter found that IDEAL (Ashlund and Hix, 1992; Hix and Hartson, 1994), a tool designed to support laboratory-based formative user interface evaluation, was also useful in supporting evaluation methods research. IDEAL provided controls for marking, viewing, editing, and synchronizing the videotapes. IDEAL also supported marking tapes where critical incidents began and end, allowing rapid wind/rewind to view and analyze a given critical incident. Critical incidents could also be annotated in IDEAL by any of the expert-subjects.

5.5 REPORTING CRITICAL INCIDENTS

The user-reported critical incident method requires software instrumentation to gather critical incident reports from users. For the case study, the instrumentation was easily simulated. First, to simulate clicking of the software *Report Incident* button, user-subjects pushed the space bar on the keyboard, which was programmed to produce a “gong” sound that could be heard in the audio portion of the tape. As a substitute for entering text in a dialogue box (for example) to describe a critical incident, user-subjects gave a verbal description that was captured on the audio track of the videotape. Verbal descriptions (which use a non-visual output channel) did not interfere, to the extent that typing might, with task performance during capture and observation during evaluation.

5.6 CRITICAL INCIDENT CONTEXTS

The videotapes, including verbal comments by user-subjects, provided a complete history of user activity. To address Question 2 (“Transforming critical incidents”), the context of each critical incident was "packaged" by editing a short clip from the tapes. By informal experimentation it was determined to use a 60-second video clip centered around the critical incident. This interval provided economical coverage for most of the data in this study. The tradeoff between bandwidth requirements to transmit clips on the network and richness of context was reserved for a future study. Expert-subjects reviewed the contextualized critical incidents captured from user-subjects and tried to transform each one into more specific usability problem descriptions, while the experimenter judged how well they did this.

5.7 RESULTS, DISCUSSION, LESSONS LEARNED

Although the case study was exploratory and no quantitative data were recorded, it did yield considerable understanding of problems and solutions regarding the user-reported critical incident remote evaluation method and did lend insight useful in directing future studies.

User-subjects liked the idea of having a sounding mechanism for reporting critical incidents, activated by pressing the space bar. A "gong" sound was chosen to distinguish it from all other system sounds, a choice that user-subjects liked because they said it was a bit like "gonging out" the designer for bad parts of the interface. User-subjects indicated a preference for using the same sound to report both positive and negative critical incidents but user-subjects rarely, if ever, identified positive critical incidents. Also, user-subjects usually reserved the "gong" for the most severe critical incidents where task performance was blocked and often did not "gong" for less severe situations where they could perform the task but it was awkward or confusing. For example, most user-subjects had minor trouble rotating an image, but they all eventually figured it out and none signaled an associated critical incident. Expert-subjects did see this trouble with rotation as a lower severity critical incident, an example of a critical incident recognized by expert-subjects but not by user-subjects. There was also one case of a critical incident identified by the experimenter but missed by all user-subjects and expert-subjects. To summarize the results for Question 1 ("Identifying critical incidents"), expert-subjects found very few critical incidents missed by user-subjects, and those problems missed were ones of less importance.

Of the two video sources, the tape of the scan-converted screen provided the most valuable data for the expert-subjects and the experimenter. The camera on the user-subjects was only occasionally useful for revealing when a user-subject was struggling or frustrated (e.g., pounding on the "gong" key).

The principal problem exposed by the case study was the need to prompt users continually for verbal protocol that is so essential in the tape clips for establishing task context for the critical incidents. Even though user-subjects were asked up-front to give a continual verbal commentary, "thinking aloud", they generally did not speak much without prompting. This problem is exacerbated in the case of remote evaluation, since an observer will not be present and the user must be self-actuated in talking. In the case study, verbal protocol was essential for expert-subjects to establish for each critical incident what the task was, what the user-subject was trying to do, and why the user-subject was having trouble.

A second major area where insight was gained involved the question of how to "package" contextualized critical incidents in the most cost-effective way with respect to network transmission cost (e.g., bandwidth) and usefulness to developers. The case-study indicated the need to examine the use of screen capture only. For remote evaluation, contextualized critical incident packages are sent over the network from users to developers. Existing digital screen capture programs can overlay audio and text (e.g., for task descriptions) on screen images, requiring less storage and bandwidth to transmit than continuous video. One further problem with automatically packaging contextualized critical incidents is that different kinds of critical incidents need different intervals. For example, it is not surprising to find that showing more

time of the beginning of a critical incident is needed to establish clear context for more complicated goal-related problems than for simpler user action-level or cosmetic problems.

Results for Question 2 ("Transforming critical incidents") showed that when task information was not given (i.e., expert-subjects were not told what user-subject was trying to do), expert-subjects expectedly had difficulty guessing what was happening and did not do well in identifying usability problems associated with a critical incident. When critical incident clips were augmented with verbal protocol about intended task and context of where the user-subject was in the task when the critical incident occurred, expert-subjects were generally able to identify associated usability problems and design flaws that led to them.

CHAPTER 6:
EXPLORATORY STUDY OF METHOD

6.1 GOAL AND OBJECTIVES

A detailed outline of the steps and objectives of this research were presented in Section 1.2. An exploratory study, step 3 of the overall research plan, is the primary focus of this thesis work. The study is described as exploratory because, although quantitative data was obtained, it was not the kind of summative study that uses statistically significant results to prove or refute an experimental hypothesis. Rather, it was an exploratory study to gain insight and understanding, under practical operating conditions, about the strengths and weaknesses of the method. In particular, objectives of the exploratory study were to:

- investigate feasibility and effectiveness of involving users with the user-reported critical incident method to identify and report critical incidents in usage,
- investigate feasibility and effectiveness of transforming remotely-gathered critical incidents into usability problem descriptions, and
- gain insight into various parameters associated with the user-reported critical incident method.

6.2 PILOT STUDY

Three volunteer pilot subjects assisted in pre-testing for this study. Two pilot subjects, who were human-computer interaction researchers trained in usability methods, helped the experimenter evaluate the overall setup of the study. A third pilot subject, an undergraduate student who had no prior training in usability methods, pre-tested the evaluation tasks to ensure that typical users could perform them. In addition, this pilot subject tested a Web-based tool that user-subjects would use to report critical incidents during performance of the evaluation tasks.

6.3 PHASE I: CRITICAL INCIDENT GATHERING

6.3.1 Participants

Twenty-four students (6 female and 18 male, 22 undergraduate and 2 graduate) participated in the study as volunteer user-subjects. These user-subjects came from a variety of academic disciplines including Accounting and Information Systems, Chemistry, Child Development, Computer Science, Economics, English, Engineering, Management Science, Math, Nutrition and Dietetics, Philosophy, and Psychology.

To obtain a group of participants that was representative of a large population of users, the experimenter administered a background questionnaire to students from two introductory Computer Science courses and selected user-subjects based on a minimum knowledge of Web browsing and Web-based information retrieval. No experience was required with the Internet Movie Database, the application evaluated in the study (three participants had used this application before participating in the evaluation session).

6.3.2 Location

The best location for users in a study of a remote evaluation method is their own workplace. However, the study itself (not the user-reported critical incident method) required a scan converter and videotape deck to make a complete continuous recording of the computer screen during task performance. Since it was not feasible to lend this equipment to each user subject, the experimenter provided the next best thing for the user-subject: a closed and quiet room, isolated from other people, including the experimenter. (Once the user-reported critical incident method is fully developed and disseminated, screen capture software and disk storage will suffice for any user in any location.)

The experimenter was located in a room adjacent to that of the user subjects. An intercom system was installed in both rooms as a safety net in case user-subjects experienced any hardware or software problems during the evaluation session, but the evaluator did not have any interaction with user subjects during task performance.

6.3.3 Equipment

Hardware

Figure 6-1 illustrates the setup of the equipment used in the user-subject room for Phase I.

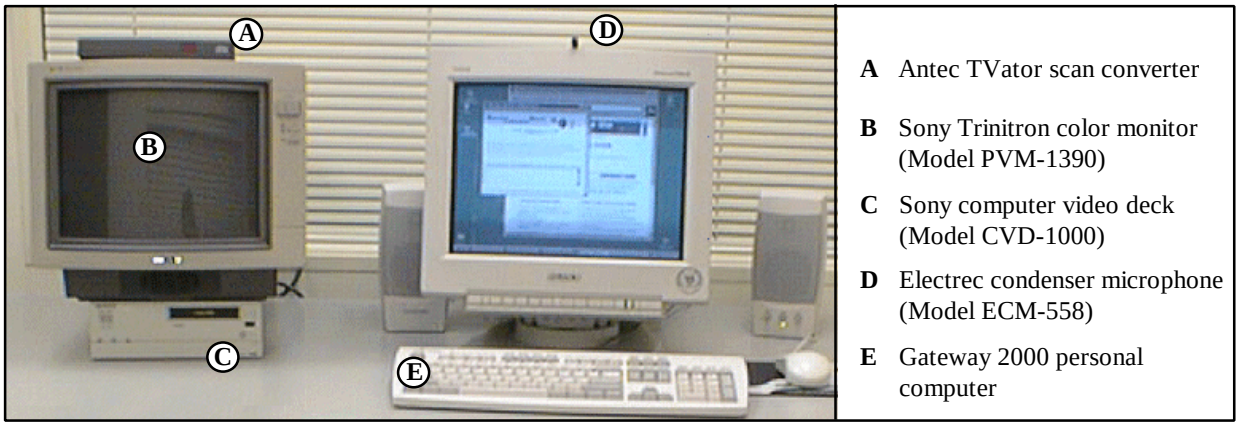


Figure 6-1. Equipment used for Phase I

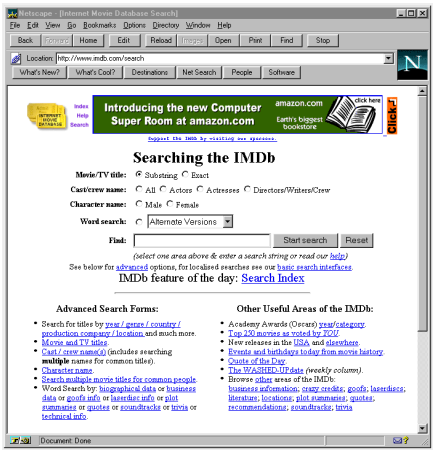
Each user-subject individually performed the evaluation tasks using a personal computer (E). A scan converter (A) captured the computer's screen action, which was recorded via a Hi-8 videotape deck (C). A lapel microphone, located on top of the computer's monitor (D), was installed to record user-subject comments on the audio track of the videotape. In addition, a color monitor (B) was included in the set-up, but was used only for the experimenter to verify that the screen image was being recorded on the video deck.

Software

Experimental application

The application evaluated in this study was the Internet Movie Database (http://www.imdb.com). As described by its developers, the Internet Movie Database (IMDb) is a service from an international organization whose objective is to provide useful and up-to-date movie information freely available on-line, across as many systems and platforms as possible. Advertising and sponsorship finance this service, which currently contains over 100,000 movies with over 1,500,000 filmography entries.

The user interface of the main search page of the Internet Movie Database (http://www.imdb.com/search) is depicted in Figure 6-2. This search page contains mechanisms for simple and advanced searching. Simple searching is accomplished by clicking radio buttons to establish the search category (e.g., Character name: female) and entering a search-string query, for that category, in a text-edit box labeled "Find". To access an advanced search form for a more specific query using more categories, users click on a link below the simple search form.

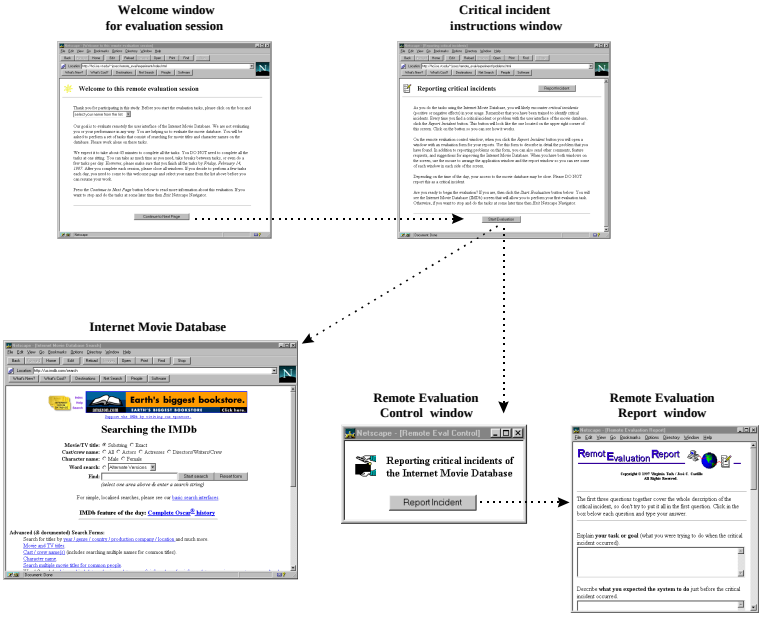


Click figure to see enlarged image.

Figure 6-2. User interface of main search page of the Internet Movie Database

Critical incident reporting tool

The critical incident reporting tool was a Web-based tool that allowed user-subjects to report, in a structured way, all critical incidents that they identified during their experimental session. Figure 6-3 depicts the sequencing relationships among the critical incident reporting tool screens and the application screen. The two windows that appear at the top of the figure provide preliminary information about the session (e.g., general instructions for the study, information on reporting critical incidents). After reading these introductory windows, user-subjects are able to access the experimental application and a window containing the *Report Incident* button. This latter window contains an instrumentation routine that allows user-subjects to report and store critical incident data.

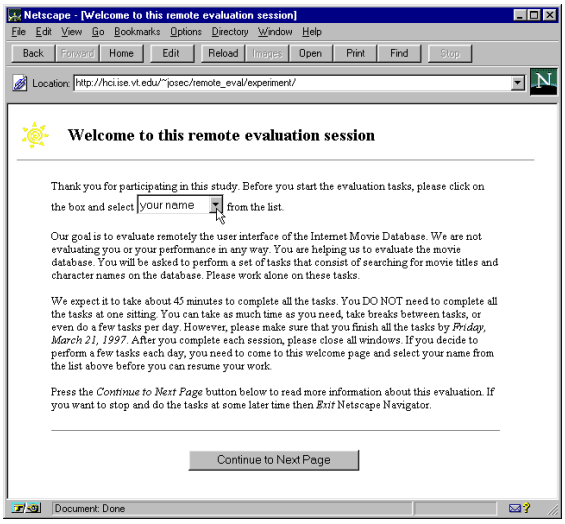


Click figure to see enlarged image.

Figure 6-3. Sequencing relationships among the screens for Phase I

Welcome window

The Welcome window, illustrated in Figure 6-4, provides general information and instructions for participation in the study.



Click figure to see enlarged image.

Figure 6-4. Welcome window for remote evaluation study

In this window, user-subjects first selected their name from a pull-down list (Figure 6-5).

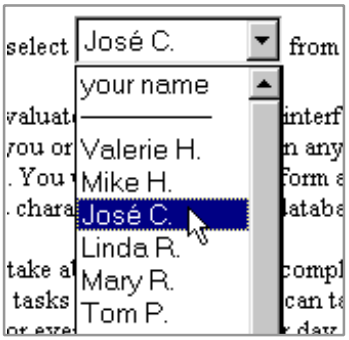


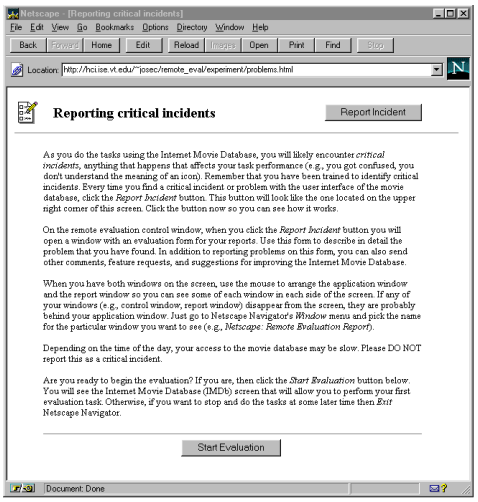
Figure 6-5. User-subject selecting his name from the list (fictitious list here)

A name list was used to avoid errors that might occur with typed name entries. To somewhat protect the identity of each user-subject, the list showed the participant's first name and last initial (e.g., José C.). These names were internally mapped (by an instrumentation routine) to a user identification number (e.g., User #3 for José C.). This was done to couple each critical incident report to the corresponding user-subject, while still providing anonymity to the experimenter and evaluator-subjects. When user-subjects finished reading the Welcome

window, they clicked on the *Continue to Next Page* button at the bottom of the screen, to access another introductory page with information about reporting critical incidents.

Critical Incident Instructions window

The window illustrated in Figure 6-6 provides general information about reporting critical incidents during task performance.

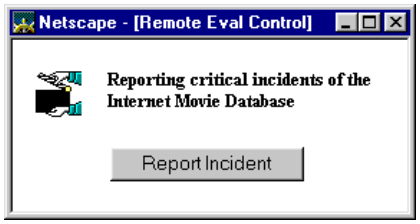


Click figure to see enlarged image.

Figure 6-6. Critical Incident Instructions window

Remote Evaluation Control window

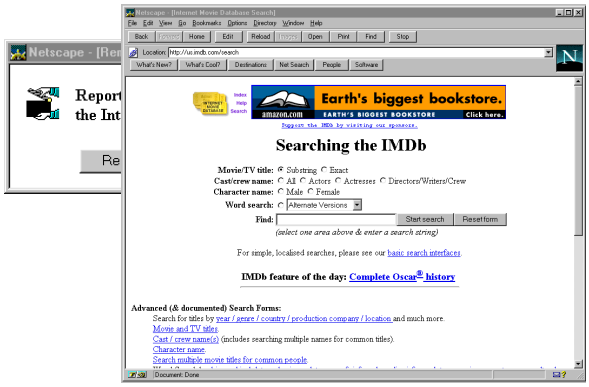
Figure 6-7 illustrates the Remote Evaluation Control window.



Click figure to see enlarged image.

Figure 6-7. Remote Evaluation Control window

To begin the tasks, a user-subject clicked the *Start Evaluation* button in the Critical Incident Instructions window (Figure 6-6), which opened the application window (with the Internet Movie Database) and the Remote Evaluation Control window. As shown in Figure 6-8, the Remote Evaluation Control window "floated" on the desktop, running independently from the application. When both windows were on the screen, user-subjects could use the mouse to arrange the application window and the control window so that they could see some of each window on the screen.



Click figure to see enlarged image.

Figure 6-8. Positioning of the Remote Evaluation Control window and the application window

Whenever user-subjects clicked on the *Report Incident* button, an instrumentation routine was activated that opened the Remote Evaluation Report window, for user-subjects to report critical incident information.

Remote Evaluation Report window

User-subjects, as shown in Figure 6-9, used the Remote Evaluation Report window (Figure 6-10) to report all critical incidents they identified during task performance.

Click figure to see enlarged image.

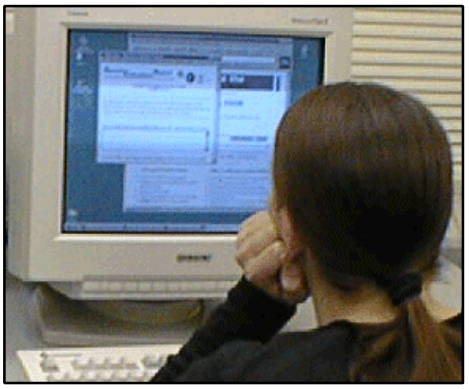


Figure 6-9. Snapshot of a user while performing the experimental tasks

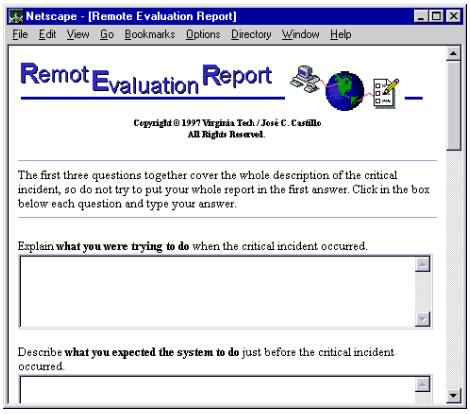
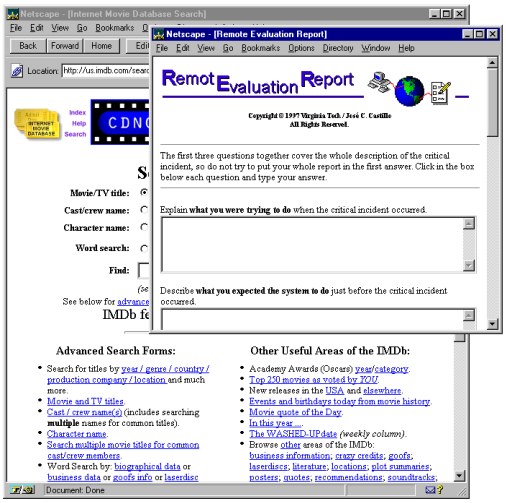


Figure 6-10. Remote Evaluation Report window

The Remote Evaluation Report window is smaller in size than, and independent from, the application window (Figure 6-11), allowing user-subjects to click back and forth between the report and application windows to work on both the task and the critical incident report.



Click figure to see enlarged image.

Figure 6-11. Positioning of the Remote Evaluation Report window and the application window

6.3.4 Protocol

Critical incident training

As discussed in Section 6.3.4, all user-subjects were trained to identify and report critical incidents during task performance. Although user-subjects learned to identify and report both positive and negative critical incidents, they were encouraged to report the negative incidents because these are the ones that reflect usability problems.

Location and time of training

Critical incident training was conducted in an isolated room (different from the user-subject room) in a laboratory of the Computer Science department. The experimenter assigned user-subjects to four groups of six people and the training was conducted over a period of two weeks, because it was difficult to schedule a time for carrying out the training with all user-subjects at once. The first group of user-subjects participated in the training one week prior to the start of the evaluation study.

Assignment of user-subjects to type of training

User-subjects were randomly assigned to two separate groups, twelve people in each group. Group 1 watched a training videotape (discussed below), but Group 2 did not. Both groups received a brief explanation (about five minutes) of identifying and reporting critical incidents and immediately, in a practice session, identified and reported critical incidents while performing a representative task using the Internet Movie Database.

Table 6-1 summarizes the division of critical incident training into video presentation and practice for the two groups.

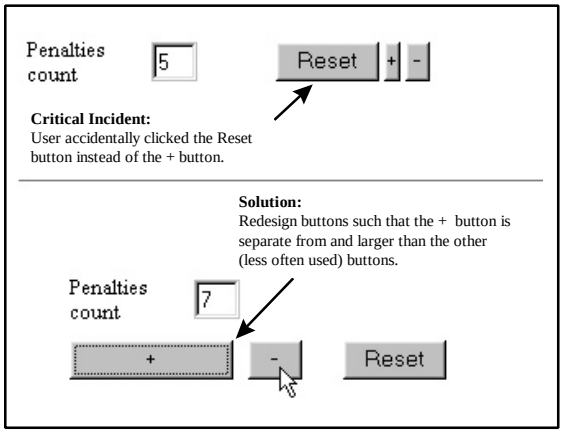
Table 6-1. Structure of critical incident training

SESSION	DESCRIPTION	TIME (MIN.)
Video Presentation (Group 1 only)	Video with extensive explanation and examples about identifying critical incidents	20
Practice (Both Groups)	Brief review and discussion about both identifying and reporting critical incidents	5
	Immediate practice for identifying and reporting critical incidents while doing a representative task on the movie database	20
	Total, Group 1	45 min.
	Total, Group 2	25 min.

Video-presentation session

The experimenter developed a 20-minute videotape for a consistent presentation about identifying critical incidents (not including information about reporting critical incidents). Only user-subjects of Group 1 saw the videotape, which showed a narrator first playing the role of a user who was identifying critical incidents while working on the following tasks:

- deleting a file in a personal document retrieval system,
- formatting a diskette in DOS format using a Macintosh computer, and
- counting game penalties using a Web-based counter (Figure 6-12).



Click figure to see enlarged image.

Figure 6-12. Critical incident found on a Web-based counter

After encountering each critical incident, the narrator of the video changed his role from a user to a knowledgeable evaluator, who carefully explained:

- the reason why that particular situation was considered a critical incident,
- the severity of the critical incident, and
- a possible solution to fix the problem.

The last part of the videotape showed the user experiencing critical incidents while doing a fourth task, changing the auto-save option in Microsoft® Word to store the file every 1 ½ minutes. The experimenter stopped the videotape before it revealed the explanation of the critical incidents and initiated a discussion so that user-subjects, as a group, identified and explained the critical incidents that occurred during that particular task.

Practice session

The practice session, taken by all 24 user-subjects, gave hands-on experience in reporting critical incidents using the Web-based tool. The practice session for Group 1 was given immediately after they watched the training videotape. Practice began with a five-minute review by the experimenter of identifying and reporting critical incidents. This brief review and discussion was followed by a twenty-minute session, where user-subjects performed a representative task with the Internet Movie Database (i.e., Find the biography of actor Denzel Washington) and practiced identifying and reporting critical incidents with the Web-based tool.

Search tasks

Each user-subject performed the same six search tasks using the Internet Movie Database. The experimenter created the tasks shown below to approximate tasks that a typical user would perform with the movie database.

1. Write down the number of movies containing the word 'bacon' (e.g., a bacon, bacon and eggs) in the title.
2. Find the performer whose character name was Michelle Thomas.
3. Find the titles of the four most recent movies directed by Steven Spielberg.
4. Find the titles of all mystery movies from 1966 to 1967 produced in French.
5. Find the titles of all movies featuring both Robert De Niro and Meryl Streep.
6. Find the titles of all movies in which Billy Crystal acted in the 90's and that are over 2 hours long.

User-subjects wrote their responses to these tasks on a participant answer sheet. The computer screen was videotaped while they performed the tasks.

User instructions

User-subjects were informed in the Welcome window (Figure 6-4) that the goal of the study was to evaluate remotely the user interface of the Internet Movie Database, and was not to evaluate their own performance in any way. Expected time for completing all tasks was about 45 minutes, and user-subjects were not required to complete all the tasks at one sitting. They were instructed to take as much time as they needed, take breaks between tasks, or even do a few tasks per day, as long as they finished by a date predefined by the experimenter (user-subjects had a period of three weeks to complete the evaluation tasks).

The experimenter also provided additional guidelines on reporting critical incidents:

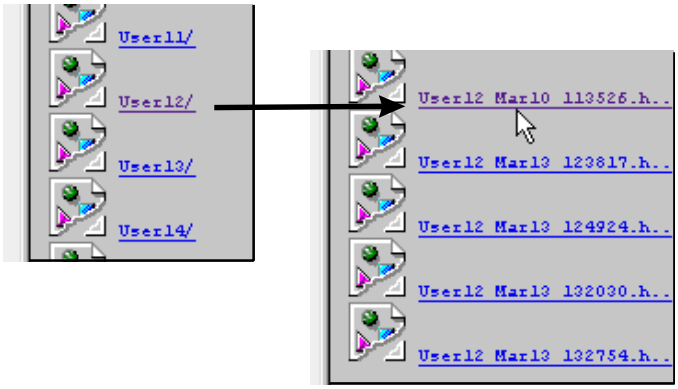
- depending on the time of day, access to the movie database could be slow (due to Internet traffic), and this was not to be reported as a critical incident;
- user-subjects could send more than one complete report for the same critical incident and/or task;
- user-subjects could send incomplete reports when necessary (e.g., only indicating the user task and a description of the problem encountered); and
- user-subjects could also use the report form to send other comments, feature requests, and suggestions for improving the Internet Movie Database.

User anonymity

The reporting system was designed to keep user reports anonymous to the experimenter (Waskul and Douglass, 1996). Even though user-subjects identified themselves before starting the tasks, critical incident reports were not associated with user-subject names because the system internally mapped each name to a user identification number. User-subjects were informed about this protection.

The experimenter created a separate directory for each user identification number to store the corresponding individual's critical incident reports anonymously in the experimenter's computer (Figure 6-13). When a user-subject clicked the *Report Incident* button, this identification number

was sent, along with the report, to the instrumentation routine, which stored the report in the proper directory. Additionally, the user identification number was used to create the filename of the report such that it included the user identification number, date, and time of submission in the name itself (e.g., User12_Mar11_113526.html), and the title in the heading of the critical incident report (e.g., Critical Incident Report #1 sent by User #12). Moreover, the instrumentation routine (i.e., CGI script) added HTML tags to the file (i.e., report), making it possible for evaluators to read the report using a Web browser. Thus, the Web browser, a commonly available tool, could be used for both submitting and retrieving critical incident reports.



Click figure to see enlarged image.

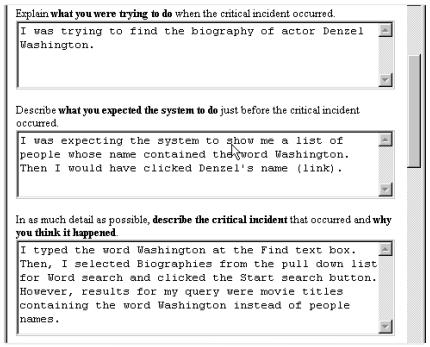
Figure 6-13. Providing anonymity for user-subject critical incident reports

6.3.5 Data collection

After receiving training to identify and report critical incidents, user-subjects performed the six representative tasks and reported critical incidents identified during task performance using the Remote Evaluation Report window. Carefully structured questions about each critical incident serve as content-based criteria for consistently gathering information so that evaluators can easily transform it into usability problem descriptions. To better understand the kind of information gathered via these questions, consider a real critical incident report made by User #X during the practice session of the training.

As illustrated in Figure 6-14, data gathered by the first three questions included:

- a description of the user task in progress when the critical incident occurred,
- expectations of the user about what the system was supposed to do when the critical incident occurred, and
- a detailed description of the critical incident (what happened and why user thought it happened).



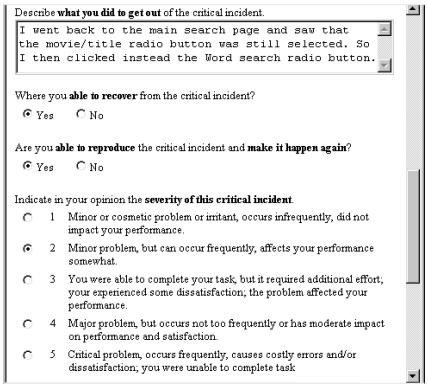
Click figure to see enlarged image.

Figure 6-14. Indication of user task and description of the critical incident

The pilot study revealed that, for some user-subjects, it was not obvious that the Report window contained more than three questions; thus, the experimenter instructed user-subjects not to write all the explanation of the critical incident in the answer to the first question and to use the scrollbar to find the remaining questions.

The next four questions (Figure 6-15) were used to gather the following data:

- an indication of whether the user could recover from the critical incident and, if so, a description of how the user did so,
- an indication of the user's ability to reproduce the critical incident, and
- a severity rating of the critical incident.

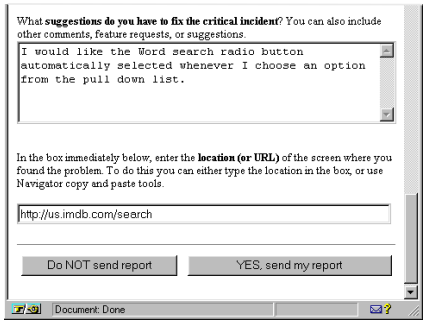


Click figure to see enlarged image.

Figure 6-15. Indication of how user got out of the situation, ability to recover and reproduce the critical incident, and severity of the critical incident.

With the last two questions, shown in Figure 6-16, user-subjects were asked to:

- send additional comments, suggestions, or possible solutions for fixing the problem, and
- enter the URL (or location) of the page where the user-subject found the critical incident.



Click figure to see enlarged image.

Figure 6-16. Suggestions for fixing the problem and location of the page with critical incident

After typing a critical incident report, user-subjects were able to send the report to the experimenter or to discard it (i.e., by clicking *Do Not send report* button). Following the evaluation session, all 24 user-subjects completed a satisfaction questionnaire (Appendix C.2) asking about their experience as remote users. During data analysis (Section 6.3.6), the experimenter found that he needed more information from user-subjects, so he administered a second questionnaire (Appendix C.3), but only 18 user-subjects completed it.

6.3.6 Data analysis

After all data gathering was completed, the experimenter reviewed the 24 one-hour videotapes twice, tagging and coding critical incident data, to determine:

- the specific instant at which user-subjects identified the critical incidents (before or after completing a task),
- the quantity and severity ratings of critical incidents missed by user-subjects,
- the relevance of audio for the method, and
- any other useful information relevant to help identify critical incidents.

Additionally, the experimenter produced contextualized critical incident packages, for evaluator-subjects in Phase II to transform into usability problem descriptions, by:

- randomly selecting critical incident reports and videotapes from six user-subjects who participated in both the videotape training and practice session of the training,
- carefully selecting one good (i.e., complete and precise) report for each of the tasks (one report per user-subject), and

- using a video mixer and controller to edit each videotape and create a new tape containing six screen-sequenced clips of fixed length (three minutes long) mapping to the critical incident reports.

Regarding the latter point, the earlier case study (Hartson et al., 1996) used two-minute clips successfully. Three-minute clips were used in this thesis study to determine if the extra length would provide more useful information. From feedback of evaluator-subjects, discussed later, two minutes probably would have been enough.

6.4 PHASE II: TRANSFORMATION OF CRITICAL INCIDENT DATA INTO USABILITY PROBLEM DESCRIPTIONS

6.4.1 Participants

Four volunteer participants (two graduate students from Computer Science and two from Industrial and Systems Engineering all trained in usability methods) served as evaluator-subjects. Their role was to analyze critical incident reports sent by user-subjects and convert them into usability problem descriptions.

6.4.2 Location, equipment, and materials

Since evaluator-subjects did not require any special equipment to analyze critical incident data, they were able to do the analysis at their place of preference (e.g., home, office). The only equipment used by two of the evaluator-subjects (the other two did not need it, as discussed below) was a VCR and a video monitor to watch video clips containing critical incident data.

Critical incident reports were presented to all evaluator subjects on paper. Because critical incident reports were accessed as hypertext documents, it was essential to ensure that these reports were easy to read with a minimum of scrolling. Figure 6-17 illustrates how critical incident information was organized for the report sent by User #X discussed in Section 6.3.5.

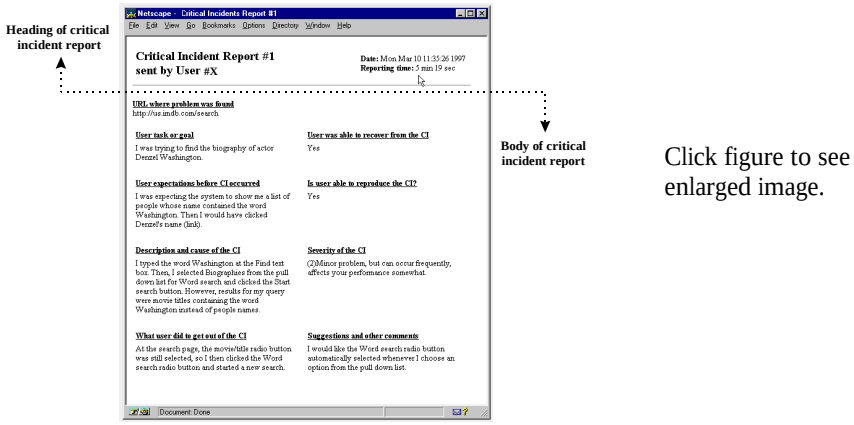


Figure 6-17. Critical incident report from User #X

A critical incident report consists of two parts: the heading and the body. The heading includes the number or sequence of the report (e.g., Report #1), the sender (e.g., User #X), the date and time of submission (e.g., Mon Mar 10 11:35:26 1997), and the amount of time the user-subject spent typing the critical incident report (e.g., 5 minutes 29 seconds). The body contains content information, answers to the questions, an specific information about the critical incident.

6.4.3 Protocol and data collection

The experimenter reviewed all 74 critical incident reports to select six reports from different user-subjects, each report mapping to one of the tasks and containing information of good quality (considering the completeness and accuracy of the report). After carefully selecting these reports, the experimenter edited the videotape corresponding to each of the reports to create a short 3-minute clip for each critical incident. Each clip was manually determined by the experimenter to be the 3-minute interval most useful in identifying the usability problem associated with the critical incident. The end product was a VHS tape containing six critical incident clips.

Two independent evaluator-subjects each analyzed six contextualized critical incident packages (i.e., a paper copy of each critical incident report and the tape containing the critical incident clips) and created a list of usability problem descriptions. Two other independent evaluator-subjects analyzed only the critical incident reports (no video clips) to create a list of usability problem descriptions. In addition, all four evaluator-subjects completed a satisfaction questionnaire (Appendices C.3 and C.4) asking about their experience as evaluators in a remote usability situation.

6.4.4 Data analysis

The experimenter compared the four lists of usability problem descriptions created by evaluator-subjects and analyzed their answers to the questionnaire to determine:

- the feasibility (i.e., time and effort) of transforming remotely-reported critical incident data into usability problem descriptions;
- the level of agreement about severity ratings of critical incidents between user-subjects and evaluator-subjects;
- the quality of the content of the critical incident reports; and
- the role of video, text, and audio during analysis of critical incident data.

CHAPTER 7:
EXPECTATIONS, DISCUSSION, LESSONS LEARNED

Section 1.3.2, Steps and objectives, contains a summary of the objectives of the exploratory study of the user-reported critical incident method. Each of these objectives, respectively, maps to a research question (Table 7-1), discussed in detail here.

Table 7-1. Objectives and research questions of the study

Objective	Type of question	Question
1	User-related	Can users report their own critical incidents and how well can they do it?
2	Evaluator-related	Can evaluators use critical incident data to produce usability problem descriptions and how well can they do it?
3	Method- and study-related	What are the variables and values that make the method work best?

7.1 USER-RELATED RESEARCH QUESTION: CAN USERS REPORT THEIR OWN CRITICAL INCIDENTS AND HOW WELL CAN THEY DO IT?

Objective 1 of Section 1.2.2 translates to the following research question: Can users report their own critical incidents and how well can they do it? In this study, Objective 1 is divided into the following sub-objectives:

1. Explore issues about user-subject performance in identifying and reporting critical incidents:
    - User-subject ability to identify and report critical incidents during task performance
    - User-subject activity sequencing and timing in reporting critical incidents
    - Level of time and effort required to report critical incidents
    - User-subject ability to rate severity of critical incidents
    - User-subject ability to identify high severity critical incidents as well as low and medium severity critical incidents

2. Obtain subjective data about user-subject perceptions, preferences, and attitudes towards remotely-reporting critical incidents:
    - User-subject attitudes towards remotely-reporting critical incidents
    - User-subject preferences with respect to reporting critical incidents anonymously
    - User-subject perceptions with respect to interference with user tasks
    - User-subject preferences relating to reporting negative and positive critical incidents

Each of these sub-objectives is discussed here.

7.1.1 Issues about user-subject performance in identifying and reporting critical incidents

User-subject ability to identify and report critical incidents during task performance

Expectation #1: User-subjects will be able to report their own critical incidents during task performance.

For each user-subject, the experimenter watched full videotapes to identify the number of critical incidents that each participant encountered and reported during task performance (Figure 7-1).

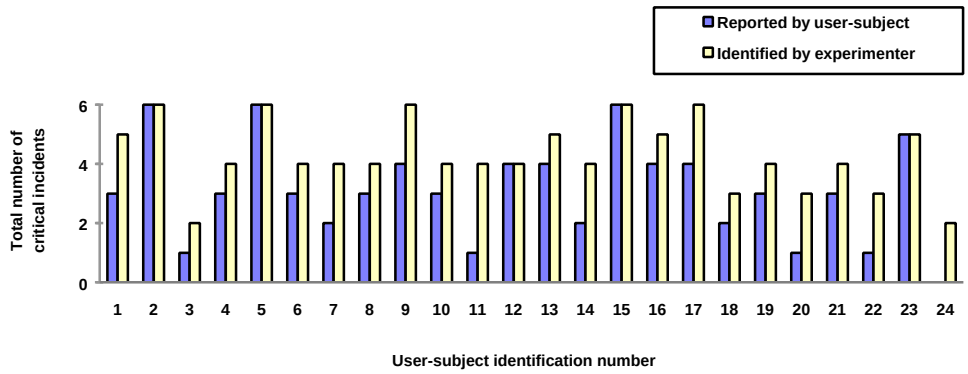


Figure 7-1. Number of critical incidents reported by all 24 user-subjects

Across all user-subjects, the experimenter found 97 critical incidents (Figure 7-2): 66 reported by both experimenter and user-subjects and 31 identified only by the experimenter (mostly of low severity – see Figure 7-15). User-subjects sent a total of 74 critical incident reports (mean: 3.1 reports per user-subject, standard deviation: 1.7). Interestingly, 8 low severity critical incidents were reported by user-subjects in cases where the experimenter did not recognize from review tapes that user-subjects were experiencing a critical incident. The experimenter did not, however, consider these as gratuitous reports sent to please the experimenter, concluding that these critical incidents were known in the minds of the user-subjects but not evident visually in the videotapes. Nevertheless, these 8 reports were not considered during data analysis.

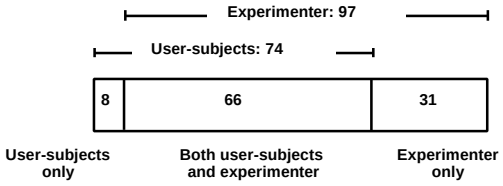


Figure 7-2. Number of critical incidents identified by user-subjects and experimenter

Figure 7-3 illustrates a characterization of the number of critical incidents reported by user-subjects: 9 user-subjects reported more than 3 critical incidents, 7 user-subjects reported 3 critical incidents, and 8 user-subjects reported less than 3 incidents.

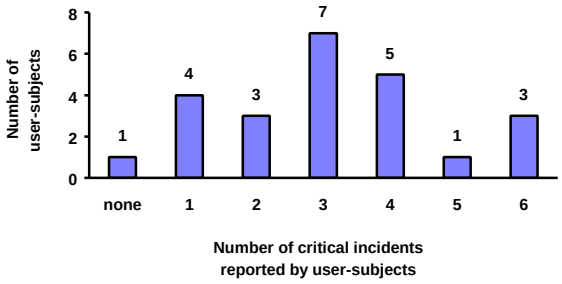


Figure 7-3. Number of critical incidents reported by user-subjects

Four user-subjects reported all experimenter-identified critical incidents, 10 user-subjects missed 1 critical incident, 9 user-subjects missed 2 critical incidents, and 1 user-subject missed 3 critical incidents (Figure 7-4). Although one user-subject (see user-subject #24 in Figure 7-1) did not report any critical incident, the experimenter identified two critical incidents (of low and medium severity) from watching the videotape for that user-subject.

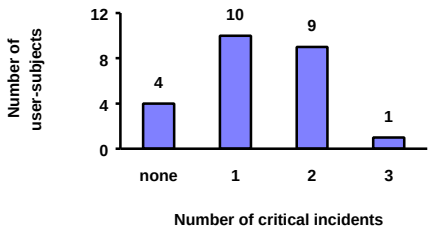


Figure 7-4. Number of critical incidents missed by user-subjects

Results indicated that users, even when working in their daily job environment and lacking interaction with evaluators, are capable of self-reporting critical incidents encountered during task performance.

Expectation #2: Only a few user-subjects may call the experimenter for assistance in solving a problem and/or task.

In real life situations, users located in their natural work setting lack the means to contact evaluators (or application developers) when critical incidents occur. To simulate this lack of interaction in the study, user-subjects were isolated and told they were on their own during the evaluation session. The experimenter was located in a room adjacent to that of the user-subjects. An intercom system was installed in both rooms as a safety net and user-subjects were instructed to use it only when they experienced hardware or software problems that prevented them from continuing with the tasks. Regardless of this restriction, it was expected that a few user-subjects might still call the experimenter for assistance in solving a problem and/or task.

During the sessions, two user-subjects (out of 24) experienced sufficient dissatisfaction while working on the final search task to leave the user-subject room and ask the experimenter for assistance in solving the task. Another user-subject left the testing room and asked the experimenter how to reinstate the *Remote Evaluation Control* window (containing the *Report Incident* button) which had been accidentally closed. Only one user-subject used the intercom system to ask the experimenter for assistance, and this occurred when the Web browser crashed.

User-subject activity sequencing and timing in reporting critical incidents

Expectation #3: User-subject activity for reporting a critical incident will be somewhat structured (i.e., following the sequence of going to the Remote Evaluation Report window and answering all questions, returning to the application window only to copy the URL of the Web page with the usability problem).

Contrary to expectations, user-subjects exhibited considerable variation in the ways they reported critical incidents. Following are some examples of this:

- placing the *Remote Evaluation Report* window on top of the application, answering all questions of the report window, and clicking back the application window only to copy the URL of the Web page where the critical incident was found,
- placing the *Remote Evaluation Report* window and the application window side-by-side to click back and forth between the windows and complete the critical incident report,
- placing the *Remote Evaluation Report* window and the application window side-by-side and clicking back and forth between the windows to continue working on the task and also answering the questions of the critical incident report,
- minimizing the *Remote Evaluation Report* window, recreating the steps followed when the critical incident occurred with the application, and reporting each step in detail,
- sending incomplete reports (i.e., not answering all questions on the *Remote Evaluation Report* window), and
- sending consecutive reports (i.e., two reports with no intervening task performance).

Expectation #4: User-subjects will send one report per critical incident.

The expectation for one report per critical incident was not met by all user-subjects. Only two user-subjects sent one report per critical incident. On six occasions, user-subjects sent consecutive critical incident reports (i.e., two reports with no intervening task performance):

- to report multiple critical incidents encountered while working on the same task,
- to add extra information a user may have forgotten to include in a previous critical incident report (about the same or different task), and
- to report critical incidents that occurred while working on <u>different</u> tasks.

In an example of this latter point, one user-subject sent two consecutive critical incident reports about two different tasks. The second report was about an incident that occurred 20 minutes earlier while working on a task that the user-subject never completed. Apparently something in the first of the two reports lead the user to remember the earlier incident, or perhaps the user-subject was "batching" reports.

Expectation #5: Implicit in the questions of the Remote Evaluation Report *window is the assumption that a single reporting format fits all reporting needs.*

Contrary to expectations, user-subjects indicated the need for more than one reporting format. For example, 3 user-subjects indicated that a few situations seemed too insignificant to merit sending a complete critical incident report. Other user-subjects sent incomplete critical incident

reports because they could not answer some of the questions (e.g., a user-subject never completed a task and, therefore, could not answer the question about how to get out of the situation). Further, some user-subjects suggested the need for sending quick questions or comments about the application, when the detailed questions in the formal critical incident report were not appropriate or warranted. This outcome led to redesign of the critical incident reporting tool. A more detailed explanation of this topic is presented in the lessons learned for Expectation #6.

Expectation #6: The flow of the reports during task performance will be somewhat structured (i.e., following the sequence of task performance, critical incident, and report)

The expectation for a structured flow of reports during task performance (e.g., task performance, critical incident identification, and reporting) was not met by all user-subjects. Not surprisingly, high severity critical incidents had the most disruptive impact on task performance and flow of activities. Eleven user-subjects sent high severity critical incident reports for tasks they never completed. Sometimes when encountering a critical incident, user-subjects gave up and continued to the next task without any effort to complete the current task. Sometimes they jumped to the next task but later came back to work on the troublesome task (with or without success).

For example, one user-subject could not find the answer to corresponding to a particular retrieval task and continued with the next task. After completing the last task, the user-subject came back to work on a previous unfinished task but became frustrated (user-subject comments: "Forget it. I give up!") and sent a critical incident report, still without completing the task. Another user-subject could not complete the last three tasks and worked with them on and off at random times instead of one after the other which lead to unordered critical incident reports.

Expectation #7: Most users-subjects will report critical incidents immediately after they occur.

Substantial variation was observed in user-subject behavior with regard to timing of critical incident reports. User-subjects reported critical incidents during task performance, immediately after the task ended, at a later time, or never did report the critical incident. Although user-subjects were directed to send a report immediately after encountering a critical incident, they sent 52 (or 70%) of all 74 critical incidents reports (Figure 7-5) *after* the task ended:

- with successful completion,
- with incorrect results (unknown to user-subject), or
- as an abandoned task with no results.

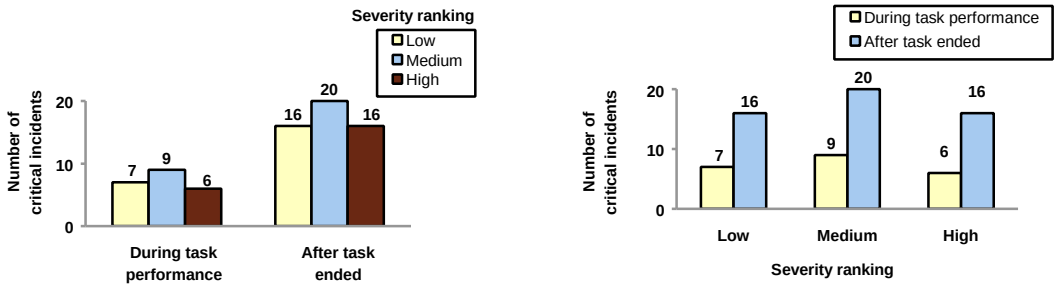


Figure 7-5. Most critical incident reports were sent after task completion

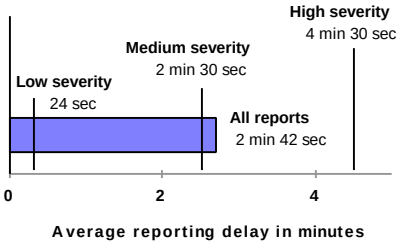
Possible reasons for waiting until completing the task to report a critical incident include the natural desire for closure on one task before doing another, the desire to wait until enough information was acquired to make a complete report, and the desire to avoid interference of the report with task performance. The following comments from user-subjects support some of these explanations:

- "I'll try to see if I can fix the critical incident on my own before I waste someone else's time."
- "[I reported critical incidents after completing my tasks] because I wanted to see if I could still get the job done."
- "This way [(i.e., reporting after completing my tasks)] I have a better change of figuring out the problem on my own."
- "I preferred to report the critical incidents when I had completed or attempted to complete my task. I did this because I believed that my job was to complete my task if possible and (if any) report incidents. I guess I felt that if I stopped at each critical incident I would loose track of the task and get on a tangent."
- "Then [(i.e., waiting to report until completing my task)] I could find out what was wrong and [I] can explain how to fix it in my report."
- "If it's not a major incident [(i.e., high severity),] I would prefer to wait [to report] so I could get my task done."
- "[I waited to report] because after I finished my task I have already figured out how to fix it and what went wrong. That way I can give a better response on how to fix it."
- "Reporting incidents after I tried to solve them gave me a better idea of how to suggest to solve it."
- "I preferred to wait until I finished my task to report my critical incidents because I felt that I had a fuller understanding of the nature of the incident, including possible better ways to do some things. Also, some of the questions on the critical incident report did not apply until I had completed the task (for example: "were you able to complete the task?"

- "[I report critical incidents after completing my task to] see if anything happens along the way."
- "Then I can see what impact the incident had on my task."

Two user-subjects, however, indicated preference for reporting each critical incident immediately after its occurrence so they do not forget to report the incident (user-subject comments: "[I prefer to report a critical incident immediately after it occurs] so [the situation] is fresh in my mind, otherwise I would probably forget to report it").

In an attempt to learn more about the nature of the timing issues, the experimenter watched all 24 videotapes yet again. For each critical incident report, the experimenter determined the point in time when it was first evident that a critical incident had occurred. As illustrated in Figure 7-6, the average time interval between this time and the point of reporting was 2 minutes and 42 seconds (standard deviation of 5.5). The shortest delays occurred when reporting low severity critical incidents the longest for high severity ones.



Severity ranking	Mean (reporting time)	Standard deviation
Low	2 min. 42 sec.	5.5
Medium	24 sec.	0.5
High	4 min. 30 sec.	6.6

Figure 7-6. Average delay in reporting after clear onset of critical incidents

Delays roughly corresponded with severity of the critical incident. The presumption is that a more severe critical incident requires more information to report and, therefore, results in a larger delay before reporting.

Lessons learned about design

The observations in this part of the study indicated a need for much more flexibility in structure, quantity, format and timing of reports than was expected. A new design (Figure 7-7) for the critical incident reporting tool addresses all these concerns. Each of the problems observed is discussed below.

Click figure to see enlarged image.

Figure 7-7. New critical incident reporting tool

Need to de-couple critical incident identification from reporting

Perhaps the most significant observation in the study involved the delay in reporting critical incidents, as described above. As the reader may recall from Chapter 4, screen-sequence video clips accompany textual reports as visual context to obtain more information about the critical incident. Automatic capture of these clips requires some trigger mechanism to initiate the capture process. The clicking by the user on the *Report Incident* button was intended to be this trigger mechanism.

However, the results of this study show a highly variable, and often large, delay between the time when the video clip should begin at the clear onset of a critical incident and the time when user-subjects click the *Report Incident* button. The result is often video clips that contain user actions irrelevant to the critical incident. Thus, this study reveals that a different trigger mechanism, separate from the *Report Incident* button, is required. That trigger mechanism is the *Begin Incident Capture* button of the new design, as shown near the top of the window in Figure 7-7. The user clicks this button at the inception of a critical incident, when an occurrence of difficulty is beginning, even though information may yet be insufficient for effective reporting. Users who understand how the mechanism works can even "re-enact" a critical incident for capture. Knowing that this button triggers recording, users can do something on the screen that they want the evaluator to see and then click the button. Later, when ready, users can click the *Send Full Report* button to send a detailed report of the incident.

Need for short, quick reports

Users expressed a need for more than one kind of problem report. For situations where is not desirable to send a complete critical incident report (e.g., the critical incident is not seen as important enough), users asked for a "quick report" capability, which is accommodated in the new design via the *Send Quick Comments* button.

Need to browse and review previous reports

Users indicated a need for support in situations where they have identified a critical incident but are not sure whether they have reported that particular incident earlier. This problem is solved in the new design by way of a new feature in the critical incident reporting tool. A list (at the bottom of the window in the figure) is used to keep track of reports previously sent by a user. Reports related to a given task can be selected from this list to be browsed in read-only mode.

Need of more functionality and structure for critical incident reporting window

Results from this study suggested that the *Remote Evaluation Report* window should allows users to:

- *preview* a critical incident report before sending it to evaluators,
- *send* a critical incident report (asynchronously) to evaluators, and
- *cancel* or discard a critical incident report without sending it to evaluators.

Each question on the *Remote Evaluation Report* will be numbered and organized in two sections: Task-related questions and Problem-related questions. Numbering the questions and providing an estimated time of completion for a full report may give a better sense to users of how much time and effort is required to answer the questions. This might help a user decide for completing a full and detailed report of an incident or sending quick comments.

Level of time and effort required to report critical incidents

Expectation #8: Based on results of the feasibility case study, user-subjects will spend an average of about 3 minutes typing critical incident reports.

Although it was expected that three minutes would be sufficient time to type critical incident reports, user-subjects took significantly longer time to make critical incident reports, spending an average of 5 minutes and 24 seconds (standard deviation of 2.3) per report.

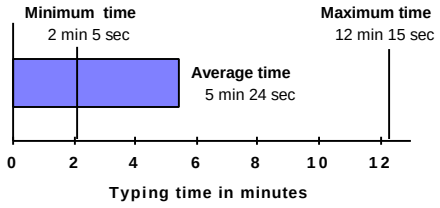
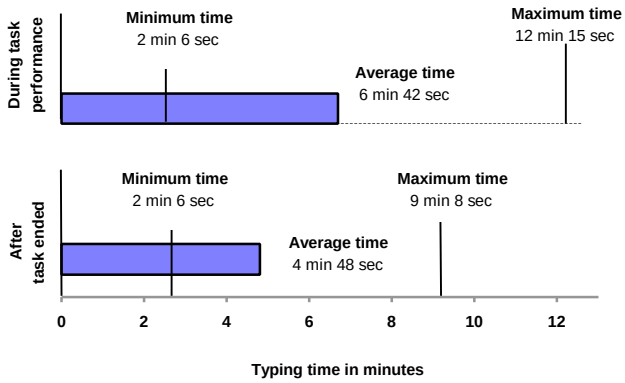


Figure 7-8. Average typing time for critical incident reports

User-subjects who reported critical incidents during task performance (Figure 7-9) spent more time typing critical incident reports than those user-subjects who waited until the task ended. Further, eight user-subjects who identified a critical incident and later worked on both task and report spent about 8 minutes typing critical incident reports.

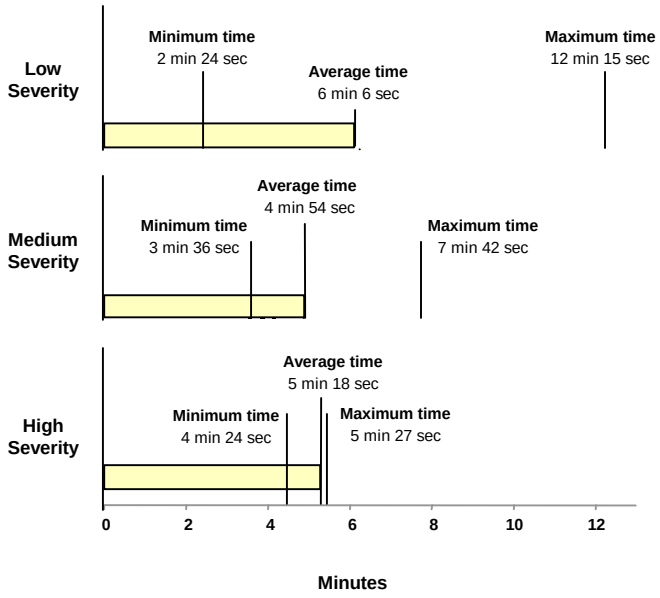


Instant of reporting	Mean (typing time)	Standard deviation
During task performance	6 min. 42 sec.	2.7
After task ended	4 min. 48 sec.	1.6

Figure 7-9. Average typing time by instant of occurrence (during or after task performance)

Expectation #9: User-subjects will spend longer time typing reports for high severity critical incidents than for low or medium severity critical incidents.

Contrary to expectations, user-subjects spent more time reporting low severity critical incidents (Figure 7-10) than medium severity incidents or high severity critical incidents. The experimenter was unable to explain this effect.



Severity ranking	Mean (typing time)	Standard deviation
Low	6 min. 6 sec.	2.8
Medium	4 min. 54 sec.	1.8
High	5 min. 18 sec.	1.9

Figure 7-10. Average typing time by severity ranking

It was noticed that user-subjects appeared to take less time to report critical incidents as they gained some experience with the reporting process. During pre-testing, two pilot subjects (trained in human-computer interaction and usability methods) spent longer time reporting their first critical incident than subsequent critical incidents. This also happened during the experimental session to 13 out of 24 (or 54%) user-subjects who sent more than one critical incident report.

User-subject ability to rate severity of critical incidents

User-subjects made severity ratings on a scale of one through five, with one being the lowest severity and five the highest. As an abstraction, the experimenter converted the ratings to severity rankings, where the low severity rank corresponds to ratings one and two, medium severity rank corresponds to rating three, and high severity rank corresponds to ratings four and five.

Expectation #10: For most cases, the experimenter will agree with severity rankings made by user-subjects.

Across all 24 user-subjects, the experimenter's rankings agreed with those of users-subjects for 55 out of 66 (83%) of the critical incidents reported by both user-subjects and evaluator-subjects. The others were balanced, with six reports being lower severity than the experimenter and five higher (Figure 7-11). Thus, results indicated that users can rate self-reported critical incidents with reasonable accuracy compared to an expert evaluator.

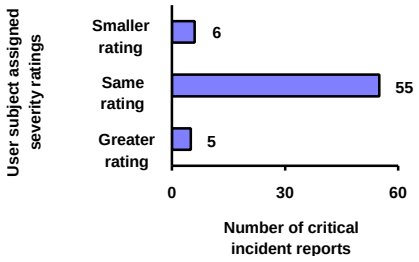


Figure 7-11. User-subject severity ratings compared to the experimenter's rating

Expectation #11: User-subjects will find it easy to rate the severity of critical incidents.

As illustrated in Figure 7-12, 22 (or 92%) of all 24 user-subjects agreed that it was somewhat easy to determine the severity of critical incidents encountered during the evaluation session.

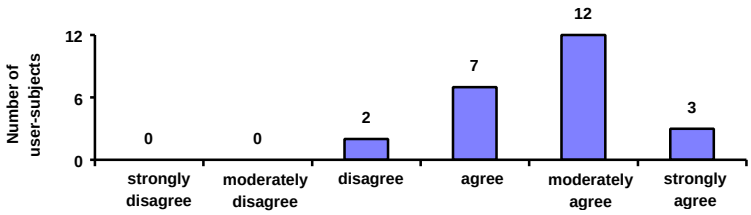


Figure 7-12. Indication from user-subjects that assigning severity ratings to critical incidents was easy to do

However, some user-subjects still had difficulties rating critical incident severity. The following may be why this happened:

- uncertainty about rating low critical incident properly (user-subject comments: "I'm used to try[ing] things four or five times in different ways to get something done, and if I make it work after a couple of tries, I might forget the details of the initial difficulties");
- unwillingness to read long descriptions for each severity rating option; and
- users' inclination to select the middle point of the scale (Figure 7-13) when uncertain about which option to choose (about 35% of critical incidents reported by user-subjects where medium severity ones).

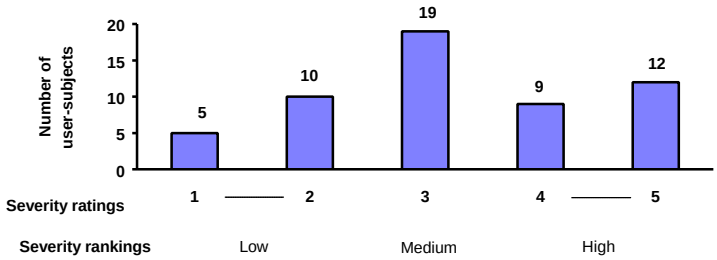


Figure 7-13. Characterization of user-assigned severity ratings by severity ranking

Lessons learned

Even though many user-subjects indicated that it was somewhat easy to rate the severity of critical incidents, the experimenter observed that a significant number of user-subjects actually found severity rating to be difficult. In retrospect, it is possible that lack of granularity of a five-point scale for identifying specific areas affected by the critical incident (e.g., satisfaction, effort) and verbose rating descriptions were the main factors contributing to this difficulty.

The odd number of rating levels probably contributed to the large number of choices in favor of the middle rating. A larger set of rating numbers might address the need for higher granularity. Previous researchers have used 6-point (Andersson and Nilsson, 1964) and 7-point scales del Galdo et al. (1986) to indicate critical incident severity.

In retrospect, it has become clear that severity rating is only part of what is needed to classify a critical incident within a usability development environment. Severity is only one part of the usability engineering concept of "importance to fix" (Hix and Hartson, 1993).

Rubin (1994) has shown that it is easier to rate severity or importance with the use of ratings that decompose into smaller and easier to judge ratings, which are then combined into an overall importance rating. Table 7-2 illustrates a possibly new approach for rating importance based on this idea of combining smaller ratings. Development of this scheme is research for future work (Section 9.9).

Table 7-2. Ranking of critical incident importance to fix

CRITICAL INCIDENT IMPORTANCE TO FIX						
Impact on satisfaction	❒ did not impact satisfaction	❒ minor impact on satisfaction	❒ moderate impact on satisfaction	❒ major impact on satisfaction	❒ experienced strong dissatisfaction	❒ don't know
Impact on effort	❒ completed task with no additional effort	❒ completed task with minor effort	❒ completed task with moderate effort	❒ completed task with major effort	❒ unable to complete task	❒ don't know
Time you spent trying to complete task	❒ normal expected time	❒ minor additional time beyond expected	❒ moderate additional time beyond expected	❒ major additional time beyond expected	❒ excessive additional time beyond expected	❒ don't know
Number of errors caused by this problem	❒ none	❒ small number of errors	❒ moderate number errors	❒ large number of errors	❒ excessive number of errors	❒ don't know
Severity of errors caused by this problem	❒ none	❒ low severity	❒ medium severity	❒ high severity	❒ extreme errors (very difficult or unable to recover)	❒ don't know
Number of people (across a mix of typical users) who will run into this problem	❒ very small number of people	❒ up to 25% of all users	❒ 25 - 50% of all users	❒ 50 - 75% of all users	❒ 75 - 100% of all users	❒ don't know
How often you expect to do this task when you are using this application?	❒ very small time	❒ up to 25% of the time	❒ 25 - 50% of the time	❒ 50 - 75% of the time	❒ 75 - 100% of the time	❒ don't know
What % of the time you perform this task would you expect to encounter this problem?	❒ very small % of time	❒ up to 25% of the time	❒ 25 - 50% of the time	❒ 50 - 75% of the time	❒ 75 - 100% of the time	❒ don't know
Criticality of task	❒ not critical at all	❒ low criticality	❒ medium criticality	❒ high criticality	❒ mission or safety critical (highest criticality)	❒ don't know

User-subject ability to identify high severity critical incidents as well as low and medium severity critical incidents

Expectation #12: User-subjects will identify the majority of critical incidents, at all severity levels, as identified by the experimenter.

User-subjects met expectations by reporting 21 out of 28 (75%) of the critical incidents identified by the experimenter as high severity (Figure 3), 19 out of 24 (79%) medium severity critical incidents, and 15 out of 45 (33%) low severity ones, as ranked by the experimenter (Figure 7-14).

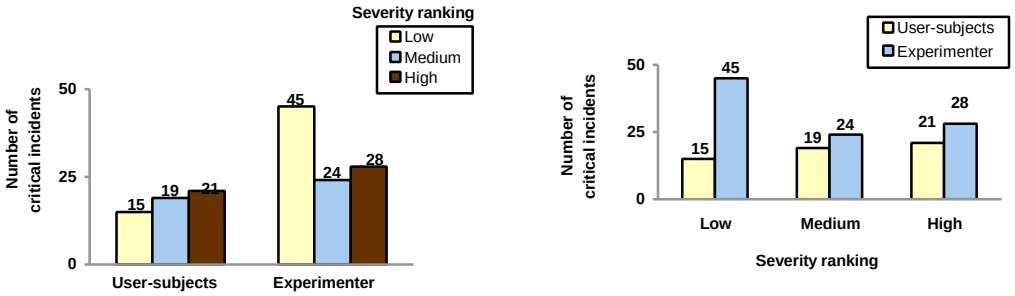


Figure 7-14. Number of reported critical incidents by severity ranking

User-subjects identified 40 out of 52 (77%) of the important (medium and high severity) critical incidents. Further, the experimenter found that 26 of the 31 critical incidents not reported by user-subjects were of low severity, the least important ones (Figure 7-15). In any usability evaluation setting, of course, the high severity critical incidents are the most important.

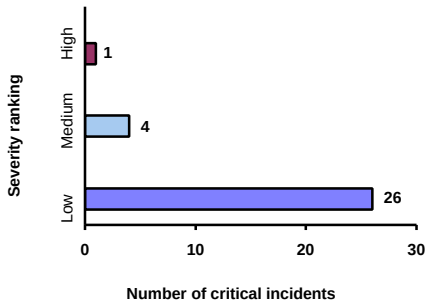


Figure 7-15. Distribution by severity ranking of the critical incidents reported only by experimenter

As severity became very low (e.g., cosmetic errors), critical incidents became more difficult to identify by both user-subjects and experimenter. There were many minor events observed by the experimenter in the videotapes that could have been called critical incidents, but user-subjects did not identify as such.

For purposes of cataloging and counting critical incidents missed by user-subjects, the experimenter:

- ignored isolated small critical incidents missed by user-subjects,
- lumped together clusters of related small critical incidents missed by user-subjects into a single critical incident "representative" of the cluster.

In sum, results indicated that users working in remote environments and lacking interaction with evaluators are capable of self-reporting critical incidents encountered during task performance. Specifically, the study indicated that users are generally capable of identifying high and medium severity critical incidents; and most of the critical incidents missed by users, but which might be identified by an expert, were of low severity.

7.1.2 Subjective data about user-subject perceptions, preferences, and attitudes towards remotely-reporting critical incidents

User-subject attitudes towards remotely-reporting critical incidents

Expectation #13: As regular users of software applications, user-subjects will want to be able to report critical incidents remotely to evaluators.

In a satisfaction questionnaire, all 24 user-subjects agreed that, as users, they want to be able to remotely report critical incident information to evaluators (Figure 7-16).

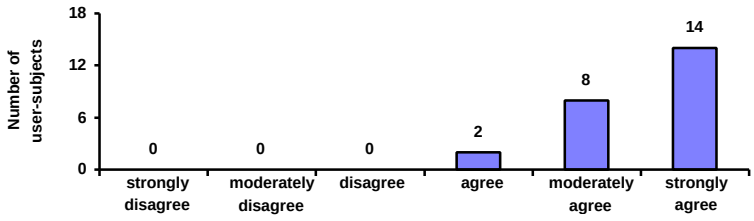


Figure 7-16. User-subject indication of their desire as normal users to report critical incidents during task performance to evaluators

The following statements made by user-subjects confirm this preference:

- "I really like the idea of being able to report critical incidents of software use."
- "I believe that the idea presented in this study is long over-due! Many times you have problems and resort to searching manuals and email addresses to [find information on

how to] fix them. It would be wonderful to let developers know what you have problems with and not what happens in testing. The real world is full of different people and problems that most certainly are not covered [with an evaluation session conducted in a usability laboratory]."

- "I like the idea of being able to immediately send reports of problems that you might have. This could be a good plan for big businesses like Microsoft if there was some way that they could guarantee a quick response."
- "It also allows me to feel better knowing that I told someone about the problem encountered, and I don't get as frustrated".
- "It would make me feel like filling out a [critical incident] report does really make a difference."

User-subject preferences with respect to reporting critical incident anonymously

Expectation #14: In general, user-subjects will want to report critical incidents anonymously.

Contrary to the expectations of this study, 17 (or 71%) of the 24 user-subjects disagreed that anonymity was important for them, mainly because the desire for feedback was incompatible with anonymity (Figure 7-17).

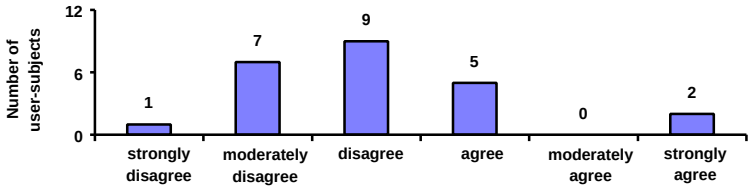


Figure 7-17. User-subjects generally did not prefer reporting critical incidents anonymously

Ten out of the 18 (or about 56%) user-subjects who completed the second questionnaire moderately or strongly agreed they would like to receive feedback (e.g., via email) in response to reports sent to evaluators (Figure 7-18).

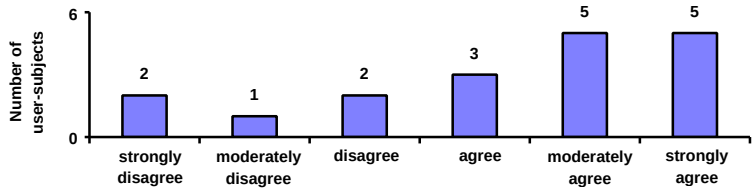


Figure 7-18. User-subject preference for receiving feedback from evaluators

More specifically, some user-subjects indicated that they expected a response from evaluators (i.e., via email) acknowledging receipts of their critical incident reports within 24 hours. Figure 7-19 shows that 9 out of 18 (or 50%) user-subjects moderately or strongly agreed that they expected evaluators to provide feedback about the critical incident report within 48 hours (however realistic this may be from the developer's point of view).

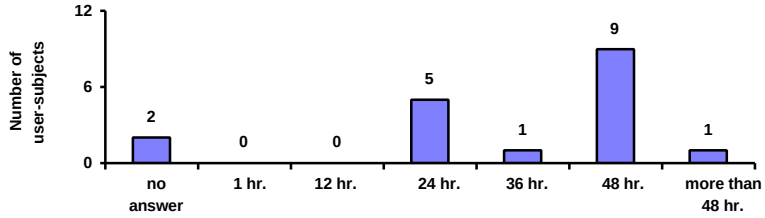


Figure 7-19. Expected time span for receiving feedback from evaluators, as indicated by user subjects

Finally, 13 out of 18 user-subjects (or 72%) indicated that they would like to be kept informed of the progress of developers in solving the reported problems (Figure 7-20).

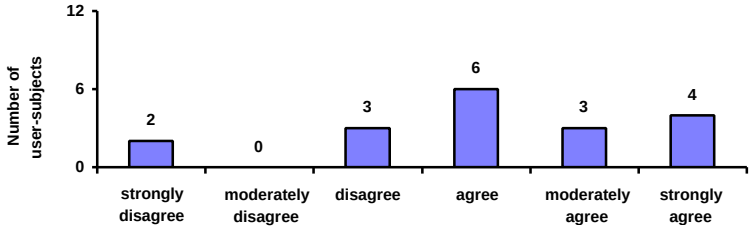


Figure 7-20. User-subject preference for being informed by evaluators of the progress in solving the problem reported

Here are some typical comments about receiving feedback from evaluators:

- "It would be great to get a quick response on the [critical incident] reports that we send to developers [or evaluators of the application being tested]. In all, it is much faster [to send them a report of my problems via email] than calling them over the phone and getting put on hold for hours."
- "[Receiving feedback from evaluators] would increase my faith and reliance on their product."
- "The faster [I receive feedback from evaluators], the more respect that would be associated [to the application]."

- "I can wait more than 48 hours [to receive feedback for my critical incident report. Nonetheless,] I would like to know if they [(the evaluators)] changed something because of my comments, rather than if they received my comments or not."
- "[How soon I want to receive feedback from evaluators] depends on the severity of the incident (and whether it was impeding my use of the application)."
- "Maybe a [message from evaluators saying] 'we received your report and will look into it right away' [is good enough], but no real response [is necessary] for a while."
- "[I would like to receive feedback from evaluators] only if [they have to tell me] something big [or important about the problem and/or its solution]. I don't want to know for 6 months what they are doing to fix it."
- "[I] would just rather have them [(developers)] fix it [(the problem)]."

Lessons learned

Given the interest of users in feedback from evaluators or developers (e.g., via email), the critical incident reporting tool should have an optional mechanism for gathering contact information. The first time the critical incident reporting tool is activated, it should open a configuration screen to store voluntary contact information (e.g., name, email, phone, fax) from users. Any other time the user activates the tool (i.e., user clicks *Report Incident* button), the system can automatically attach this contact information to the critical incident report.

As suggested by Elgin (1995), the best way to get users interested in reporting critical incidents is by getting immediate responses back to them. Responses to users could be sent via email by knowledgeable support personnel (e.g., developer who is "owner" of the screen containing usability problems) of the application being evaluated. Another medium to provide such responses could be a public area (e.g., Frequently Asked Question/Problems and Solutions Web page) where users and evaluators post messages about problems found with the application, and even possible suggestions and solutions to solve those problems.

Evaluators and application developers can encourage users to report critical incidents by rewarding those users who help them most (e.g., reporting largest amount of critical incidents, reporting incidents of highest severity).

User-subject perceptions with respect to interference with user tasks

Usage problem reporting by remote users working on real tasks for real work has considerable potential to interfere with task performance. As Elgin (1995) stated, "A big question [in remote evaluation] is [to determine] how to help them [(users)] remember this interaction [(e.g., by reporting critical incidents)] while not seeming to violate their own sense of intrusion."

Expectation #15: An indication from some user-subjects is expected that identifying and reporting critical incidents interfered with their tasks (i.e., at least in some cases of high severity critical incidents), and that longer reports would lead to a stronger perception of interference.

Contrary to expectations, 19 (or 79%) of the 24 user-subjects moderately or strongly agreed that identifying and reporting critical incidents was not intrusive and did not interfere with their tasks (Figure 7-21). This counter-intuitive result was reinforced by the fact that some of the best and most complete critical incident reports came from user-subjects who said that they felt that reporting did not interfere much. Perhaps the feeling of interference was offset by the satisfaction of being able to identify and report problems. One explanation is that user-subjects (i.e., university students) for this experimental session were not in a real work setting, where they would have to get something done, and therefore did not feel any intrusion to their tasks.

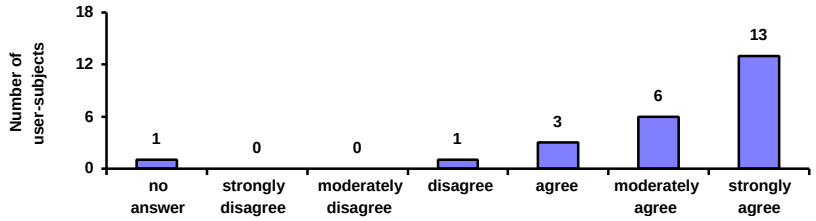


Figure 7-21. User-subject indication that identifying and reporting critical incidents did not interfere with task performance

User-subject preferences relating to reporting negative and positive critical incidents

As discussed in Section 6.3.4, all user-subjects were trained to identify and report critical incidents during task performance. Although user-subjects learned to identify and report both positive and negative critical incidents, they were encouraged to report the negative incidents because these reflect usability problems.

Expectation #16: User-subjects will be more motivated to identify negative critical incidents than positive critical incidents.

During the experimental sessions, user-subjects experienced both positive and negative critical incidents, but nevertheless reported only the negative ones. In the questionnaire, 16 out of 18 (or 89%) user-subjects moderately or strongly agreed they were motivated to report negative critical incidents (Figure 7-22), and only 7 out of 18 (or 39%) user-subjects agreed to be

motivated to report positive critical incidents (Figure 7-23). Typical comments from user-subjects include the following:

- "I usually only think of reporting the mistakes, not the successes."
- "If I was slowed down or confused by a critical incident, I was highly motivated to report it. [However,] I was not as motivated to report positive critical incidents."
- "It seems [that reporting] negative critical incidents would be more beneficial for the developers to [find usability problems and] improve [the user interface of the application], but after using the movie database I have found that there are several good things that I could report too - so perhaps [I could have sent] positive [critical incident] reports".
- "If I get feedback [from developers], it would motivate me to report both negative and positive [critical] incidents."
- "Those [(negative critical incidents)] are the ones that I notice and are unexpected. When what I want to happen actually happens, I don't take the extra note."

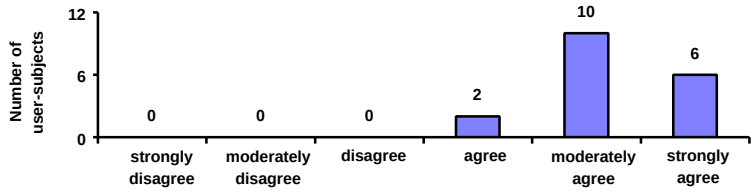


Figure 7-22. User-subject preference for reporting negative critical incidents

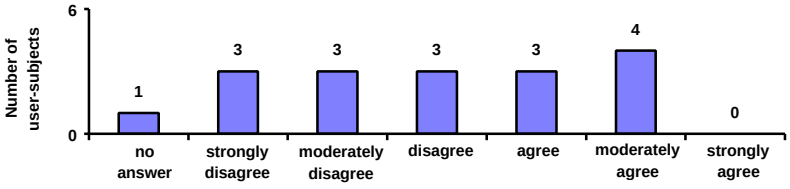


Figure 7-23. User-subject preference for reporting positive critical incidents

7.2 EVALUATOR-RELATED RESEARCH QUESTION: CAN EVALUATORS USE CRITICAL INCIDENT DATA TO PRODUCE USABILITY PROBLEM DESCRIPTIONS AND HOW WELL CAN THEY DO IT?

For data analysis in Phase II of the study (Section 6.4), two evaluator-subjects (report-only evaluator-subjects), working independently of each other, analyzed six contextualized critical incident reports to create a list of usability problem descriptions. Two other evaluator-subjects (clip-and-report evaluator-subjects), again working independently, analyzed six critical incident reports and six video clips to create a list of usability problem descriptions. In addition, all four evaluator-subjects completed a satisfaction questionnaire about their experience as evaluators in a remote usability situation.

As a general matter, the small number of evaluator-subjects and individual differences among them strongly colored the results of this phase of the study, implying a low significance of results. Therefore, the "results" obtained here are only hypotheses and not conclusions, and will serve as expectations for a future study.

Phase II of this study was conducted to investigate Objective 2 in Section 1.2.2, which translates to the following research question: Can evaluators use critical incident data to produce usability problem descriptions and how well can they do it? In this study, Objective 2 is divided into the following:

- Ability of evaluator-subjects to analyze critical incident data
- Role of textual reports in data analysis
- Role of video in data analysis
- Role of audio in data analysis
- Time and effort required to analyze critical incident data
- Level of agreement with user-subject critical incident severity ratings

Each of these sub-objectives is discussed here.

7.2.1 Ability of evaluator-subjects to analyze critical incident data

Generally, all evaluator-subjects were capable of analyzing critical incident data to produce usability problem descriptions. For example, report-only evaluator-subjects reported similar or related usability problem descriptions for five out of six critical incident reports. The exception to this was the usability problem encountered by a particular user-subject while working on task #3 (i.e., Find the titles of the four most recent movies directed by Steven Spielberg):

> Report-only evaluator-subject #1:
>
> "[The] usability problem [for this task was] caused by network speed. [The Web] page [with information about Steven Spielberg] would have come up with the director's films listed in chronological order, starting at the top of the page, but because the network was slow and the logical top of the page was really the middle of a larger page, other information

was shown while the page was loading. [To fix this problem, developers] need to make the top of the page more clear for users that unexpectedly see it first."

Report-only evaluator-subject #2:

- "[I found various usability problems in this task:]
- External links and internal links (to within the current page) are not differentiated, giving no spatial cues for a flat information space.
- Extraneous information not relevant to task [is] displayed, cluttering [the] screen and the user's information model, requiring the user to perform unwanted [information] filtering.
- Multiple information types [are] displayed in same information space. They should be separated."

Clip-and-report evaluator-subjects showed similar results in their ability to produce usability problem descriptions. However, differences in the lists of usability problem descriptions among the 4 evaluator-subjects made it difficult to compare them (e.g., one evaluator-subject's list described usability problems found for each task while the other list described usability problems found in the user interface as a whole).

7.2.2 Role of textual reports in data analysis

Expectation #17: All evaluator-subjects might agree that analyzing critical incident reports into usability problem descriptions is easy to do.

Report-only evaluator-subjects somewhat or strongly agreed that it was indeed easy to analyze critical incident reports and create a list of usability problems descriptions. However, there was no concurrence among clip-and-report evaluator-subjects about the ease of analyzing critical incident reports without having videotape clips to create a list of usability problems descriptions. The experimenter was unable to explain this difference.

Expectation #18: All evaluator-subjects may agree that the content of critical incident reports is of high quality (e.g., completeness, accuracy).

Report-only evaluator-subjects moderately or strongly agree that it was easy to understand the content of critical incident reports. There was no concurrence among clip-and-report evaluator-subjects about this question. Interestingly, clip-and-report evaluator-subjects frequently read the report before watching the video clip to get some context about the task and critical incident.

Expectation #19: All evaluator-subjects may prefer to read the online version of critical incident reports instead of paper copy.

Evaluator-subjects from both groups indicated a preference in reading the critical incident reports on paper instead of an online version.

7.2.3 Role of video in data analysis

Expectation #20: Clip-and-report evaluator-subjects might indicate that it is easy to create usability problems descriptions based on analysis of textual reports and video clips.

Clip-and-report evaluator-subjects somewhat or strongly disagreed that it was easy to create a list of usability problem descriptions after analyzing both the critical incident reports and videotape clips (evaluator-subject comment: "It was somewhat difficult to match the two together.").

Expectation #21: Clip-and-report evaluator-subjects might prefer to determine usability problem descriptions analyzing only the videotape clips (i.e., video and audio, no textual reports).

Clip-and-report evaluator-subjects did not concur on a preference for determining usability problem descriptions by analyzing only the videotape clips, but both considered textual reports to be essential. The following are comments from evaluator-subjects about this matter:

- "No way! …it was difficult/impossible to tell the problem without the [paper] reports."
- "[Task #1:] I had no idea what was occurring on the screen when I started. In addition I could not read the text very well."
- "[Task #2:] I read the report first to get some context. It was difficult to tell [just from watching the clip] if the user was waiting for the system, reading, thinking, or what."
- "[Task #5:] I would have had no idea the user was looking for movies with Meryl Streep from the video clip."
- "[Task #6:] I could not tell from the video that the user was searching for 'time'. It was difficult … to see if the user was getting the information needed."

Expectation #22: Clip-and-report evaluator-subjects might prefer to determine usability problem descriptions without using videotape clips (i.e., only with critical incident reports).

Clip-and-report evaluator-subjects did not concur on a preference for determining usability problem descriptions without using the videotape clips (only with critical incident reports). One evaluator-subject indicated the following in the questionnaire: "[I'm] not sure how I would have liked just reading the critical incident reports."

Expectation #23: Clip-and-report evaluator-subjects might indicate that the videotape clips play an important role in understanding critical incidents.

There was no concurrence among clip-and-report evaluator-subjects about the videotape clips playing an important role in understanding critical incidents. However, these evaluator-subjects mentioned that:

- " [The video clips] helped [me] clarify [the] order of events…"
- "…much of user strategy [while doing the evaluation tasks] (e.g., searching menus) would have been lost [without the clips]."

Expectation #24: It was expected that three-minute video clips would show enough information about the critical incidents.

There was no agreement among clip-and-report evaluator-subjects about whether 3 minutes is the appropriate length to show enough information for determining usability problem descriptions:

> Clip-and-report evaluator-subject #1:
>
> "In most cases, less [than three minutes] would be fine. Of course, it always depends on the situation and the nature of the critical incident. [Actually,] two minutes would be fine."
>
> Clip-and-report evaluator-subject #2:
>
> "Some clips started in the middle of the task (e.g., after initial search). [To store enough information, the clips need] the amount of time necessary to capture the entire task for which the critical incident occurred."

Expectation #25: Report-only evaluator-subjects might indicate that video clips (with audio) of screen action, in addition to the critical incident reports, could have helped them better determine the usability problem descriptions.

Report-only evaluator-subjects moderately or strongly agreed that they believed video clips (with audio) of screen action, in addition to the critical incident reports, would have helped them create usability problem descriptions.

Lessons learned

When an evaluator-subject mentioned that it was somewhat difficult to match a critical incident report with the videotape clip, the experimenter went back to review each tape and experienced this reality. In general, it was expected that videotape clips would be a necessary supplement to written reports to understand the task context leading up to the critical incident but, frankly, the experimenter was later inclined to agree that video might not be as useful as expected. However, this small amount of data was not sufficient to rule out the usefulness of video and it should be a high priority to explore this in a future study. If it is found that video is not very useful to understand critical incident data, that would certainly make the user-reported critical incident method even less expensive.

7.2.4 Role of audio in data analysis

Expectation #26: All evaluator-subjects might prefer listening to verbal critical incident reports (i.e., audio with user's voice) rather than reading textual reports.

Clip-and-report evaluator-subjects strongly agreed on a preference for listening to verbal critical incident reports rather than reading textual reports (evaluator-subject comments: "Yes! I thought exactly this while reading the critical incident reports."). However, report-only evaluator-subjects somewhat or strongly disagreed with a preference for verbal critical incident reporting. The experimenter does not have enough data to explain this disagreement.

7.2.5 Time and effort required to analyze critical incident data

Expectation #27: Because of the additional data, it was expected more analysis effort required of clip-and-report evaluator-subjects than of report-only evaluator-subjects.

Contrary to expectations, clip-and-report evaluator-subjects took somewhat less analysis time on average than report-only evaluator-subjects. Report-only evaluator-subjects spent an average of 1 hour and five minutes analyzing all six critical incident reports into usability problem descriptions (average 10 minutes and 50 seconds analyzing each report). On the other hand, clip-and-report evaluator-subjects spent an average of 50 minutes analyzing critical incident data (textual reports and videotape clips) and creating a list of usability problem descriptions (average of 8 minutes and 20 seconds analyzing each pair of critical incident reports and videotape clips). It is possible that report-only evaluator-subjects had to spend more time analyzing critical incident data because they had less information, and therefore needed more time to understand the critical incidents.

7.2.6 Level of agreement with user-subject critical incident severity ratings

For the most part, it was expected that evaluator-subjects and user-subjects would agree about severity ratings. Clip-and-report evaluator-subjects somewhat disagreed with user-subject ratings (evaluator-subject comment: "I usually rated critical incidents less severe than user-subjects."), but there was no concurrence among report-only evaluator-subjects about this matter.

7.3 METHOD- AND STUDY-RELATED RESEARCH QUESTION: WHAT ARE THE VARIABLES AND VALUES THAT MAKE THE METHOD WORK BEST?

Objective 3 of Section 1.2.2 translates to the following research question: What are the variables and values that make the method work best? In this study, Objective 3 is divided into the following:

- Preferred location for the critical incident reporting tool
- Using the Remote Evaluation Report window
- Role of training in reporting critical incidents
- Verbal protocol issues
- Role of audio in reporting critical incidents
- Role of video in the study
- Packaging critical incident data
- Issues relating to the proximity of a critical incident and its cause

Each of these sub-objectives is discussed here.

7.3.1 Preferred location for the critical incident reporting tool

For this study the critical incident reporting tool was implemented as an independent piece of software, separate from the application. It also could have been built-in as part of the Internet Movie Database software application. Building it in requires modification of the Internet Movie Database software, not a practicable alternative in the study, since the experimenter did not have access to the source code.

Expectation #28: User-subjects will want to have a Report Incident button to report critical incidents during task performance. Further, user-subjects might prefer having the Report Incident button and critical incident reporting tool built into the application being evaluated.

All 24 user-subjects agreed to have a *Report Incident* button because they want to be able to report critical incidents to evaluators during task performance (Figure 7-24).

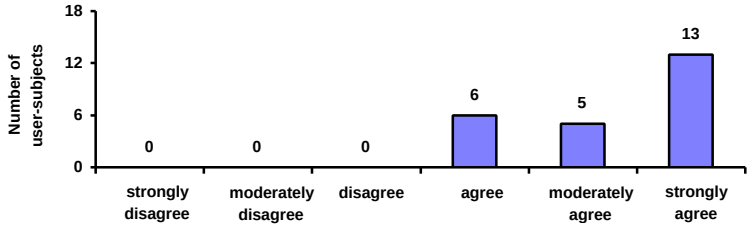


Figure 7-24. User-subject preference of being able to click on a *Report Incident* button to report critical incidents during task performance to evaluators

For reasons of convenience and logical clarity, 16 out of 18 (or 89%) users-subjects expressed a preference for having the *Report Incident* button built into the application being evaluated, rather than in a separate window (Figure 7-25). In a questionnaire, user-subjects mentioned that:

- "I found it to be a hassle to have to switch to a different window to [click the *Report Incident* button and] report an incident. A button [located in the user interface itself] would make it all much easier to make a report."
- "[Clicking the *Report Incident* button in] the separate window was tedious."
- "[I] somewhat disagree [to have the *Report Incident* button outside the application] because it is sometimes difficult to click back and forth between windows."

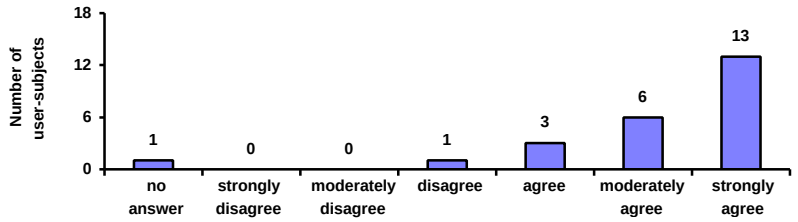


Figure 7-25. User-subject preference for placing the critical incident reporting tool built into the application

As shown in Figure 7-26, only 2 out of 18 (or 11%) user-subjects agreed of having the *Report Incident* button placed in a separate window. Some comments from user-subjects are the following:

- "[Placing the Report Incident button in the application may] clutter [the] interface."
- "[I would have the Report Incident button built into the user interface of the application] only for testing. I wouldn't want to be at a [Web] site and have that button there if I'm just looking for, say, travel information at Delta's site.".

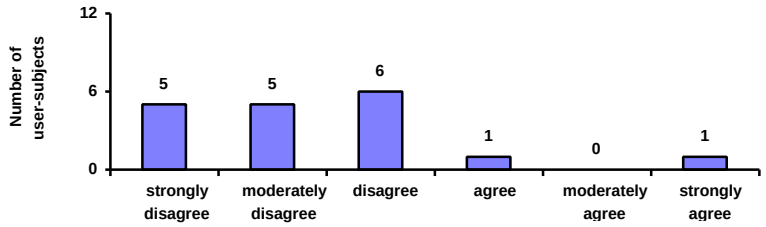


Figure 7-26. User-subject preference for placing the critical incident reporting tool in a window separate from the application

Expectation #29: Problems may arise due if user-subjects accidentally close the window containing the Report Incident button.

Two user-subjects experienced dissatisfaction when they accidentally closed the window containing the *Report Incident* button. One user-subject found how to re-open this window without help. The second user-subject, intending only to minimize the window, closed the window accidentally by clicking the **X** or *Close* button (Figure 7-27) located at the upper right corner. It was not until the next critical incident occurred that this user-subject recognized the *Report Incident* button was missing from the desktop ("I lost it [(*Report Incident* button)]. I think I found another critical incident but I can't find the *Report* [*Incident*] button"). In the questionnaire, this user-subject also indicated that the *Remote Evaluation Control* window should be smaller than the application window and should be accessible from the Web browser by selecting a bookmark.

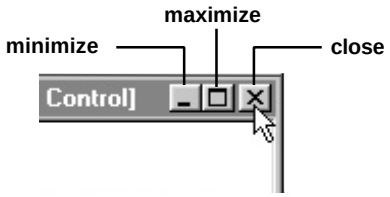


Figure 7-27. Default buttons for manipulating windows in Windows95™

Two other user-subjects accidentally clicked the Netscape Navigator mail icon at the lower right corner of the browser while intending to open the *Remote Evaluation Control* window, which they minimized earlier while working on the tasks. None of the 24 user-subjects lost the *Remote Evaluation Control* window from accidentally exiting the Web browser.

7.3.2 Using the Remote Evaluation Report window

The *Remote Evaluation Report* window contained a form that user-subjects used to enter precise information about the critical incident. This report form consisted of the following eight questions:

1. Explain what you were trying to do when the critical incident occurred.
2. Describe what you expected the system to do just before the critical incident occurred.
3. In as much detail as possible, describe the critical incident that occurred and why you think it happened.
4. Describe what you did to get out of the critical incident.
5. Where you able to recover from the critical incident?
6. Are you able to reproduce the critical incident and make it happen again?
7. Indicate in your opinion the severity of this critical incident.
8. What suggestions do you have to fix the critical incident? You can also include other comments, feature requests, or suggestions.

Expectation #30: The majority of user-subjects will find it easy to report critical incidents using the Remote Evaluation Report window.

As expected in the study, 23 of the 24 user-subjects agreed that it was easy to report critical incidents using the *Remote Evaluation Report* window (Figure 7-28). User-subjects liked that this window was smaller in size and separate from the application, allowing them to click back and forth between the windows.

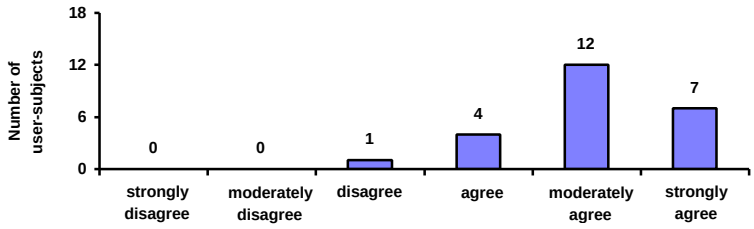


Figure 7-28. User-subject indication that it was easy to report critical incidents using the *Remote Evaluation Report* window

Two user-subjects indicated that the questions "what you expected the system to do" and "what you did to get out of the situation" were confusing. Another user-subject mentioned that those questions could be eliminated from the report form because, in user-subject's opinion, "were unnecessary and could take too long to answer for a common user".

Expectation #31: User-subjects might accidentally close the Remote Evaluation Report window accidentally before sending the report.

Although some user-subjects closed the *Remote Evaluation Report* window as a successful way to discontinue a report, one user-subject did inadvertently close the window and lost a report. This user-subject answered the first few questions and then tried to minimize this window to go back to the application and read an error message that resulted from a search query made earlier. But instead this user-subject accidentally clicked the *Close* button (see Figure 7-27) on the upper right corner of the window. This was the only critical incident report lost during the entire evaluation session. The user-subject experienced frustration but did not want to call the experimenter for assistance for a second time — this is the same user-subject who accidentally closed the window containing the *Report Incident* button (Section 7.3.1). In the questionnaire, this user-subject mentioned that *Remote Evaluation Report* window should fit in the screen side by side with the application window. Finally, none of the user-subjects lost the *Remote Evaluation Report* window from accidentally exiting the Web browser.

In general, user-subjects found it easy to report critical incidents using the *Remote Evaluation Report* window. However, some of them spent longer time typing answers for the following questions:

1. describe the critical incident and why you think it happened,
2. what you expected the system to do before the critical incident occurred, and
3. how you got out of the situation.

The answer to the first question is essential because it yields context information of the critical incident and could probably reveal the cause of that problem. Novice users (novice at working with the application being evaluated) may take longer time than experienced users to report critical incidents and probably would be more difficult for them to answer the last two questions. Thus, these type of users can be allowed to send a quick comment or question (i.e., by clicking

the *Send Quick Comments* button in Figure 7-7) instead of completing a formal critical incident report.

Users prefer the *Remote Evaluation Report* window separate and smaller in size than the application window. This allows them to click back to the two windows, for example, if a user wants to read an error message in the application window and later clicked on the report window to explain the situation. Further, the data collection tool should automatically capture the name, title, and/or location (e.g., URL for Web-based applications) of screen where user encountered critical incidents.

7.3.3 Role of training in reporting critical incidents

Minimal instruction principles were applied to design critical incident training for user-subjects. The critical incident training consisted of two parts (video presentation and practice session), presented to user-subjects separately. To investigate the role of training (Section 6.3.4) for users to identify and report critical incidents effectively, user-subjects were randomly assigned to two separate groups, twelve people in each group. Group 1 watched a training videotape with information about identifying critical incidents, but Group 2 did not. Both groups received a brief explanation (five minutes) about both identifying and reporting critical incidents and immediately after, in a practice session, identified and reported critical incidents while performing a representative task using the Internet Movie Database.

Expectation #32: User-subjects in Group 1 (with video presentation) may report a larger number of critical incidents than users-subjects in Group 2.

Across all participants of each group, user-subjects of Group 1 (with video presentation) reported 30 critical incidents and user-subject of Group 2 reported 25 incidents (Figure 7-29).

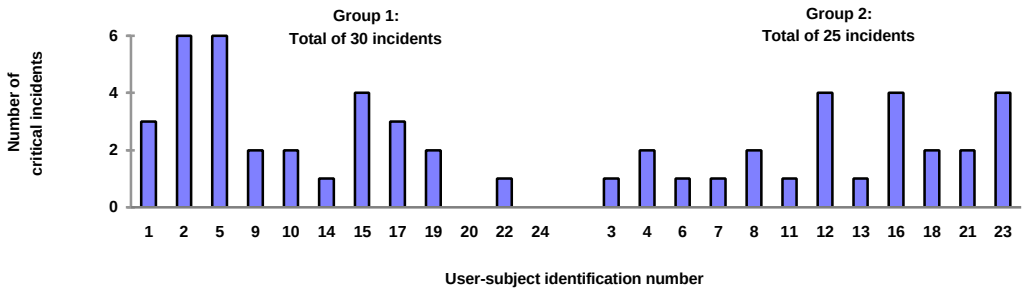


Figure 7-29. Number of critical incidents reported by user-subject of each group

User-subjects of Group 1 reported 8 low severity incidents, 11 of medium severity, and 11 of high severity, finding 31% of the critical incidents identified by the experimenter across all user-subjects (Figure 7-30). Similarly, user-subjects of Group 2 reported 7 low severity critical incidents, 8 of medium severity, and 10 of high severity, finding 26% of the critical incidents identified by the experimenter.

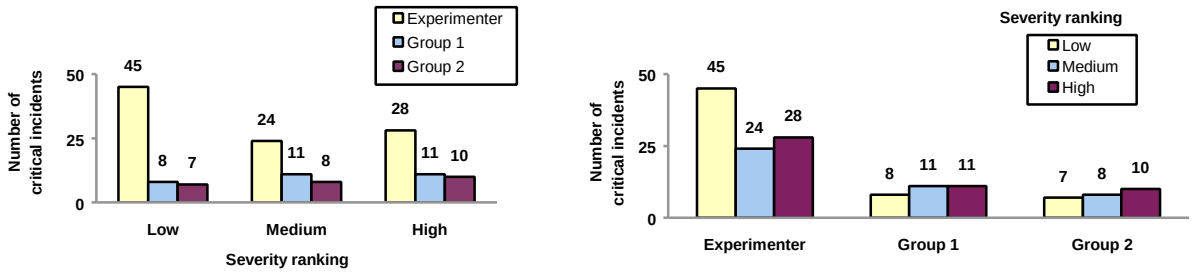


Figure 7-30. Effect of training in reporting critical incidents

Contrary to expectations, user-subjects from both groups reported a similar number of critical incidents for the various severity ratings. An interesting observation is that one user-subject of Group 2 mentioned feeling qualified to identify and report critical incidents after only participating in the practice session: "Although I received the less thorough training, I feel that it was more than adequate to allow me to complete the tasks and report the incidents". To further investigate this matter, a future study will be conducted to determine the best way to train users (training only about reporting critical incidents versus training about both identifying and reporting critical incidents) and the effect of training in the number of critical incidents reported by users.

Expectation #33: User-subjects in Group 1 (with video presentation) may produce reports of better quality (e.g., accuracy and completeness) than user-subjects in Group 2.

To investigate the quality of critical incident reports, the experimenter randomly selected one critical incident report from each user-subject. Two pilot subjects (researchers trained in human-computer interaction and usability methods) carefully examined the accuracy and completeness of these reports and agreed that there was no significant difference in the quality of the reports. Thus, it seems that the video presentation did not have a significant effect on the quality of critical incidents reported by user-subjects. However, some of those user-subjects who watched the videotape indicated that they felt better prepared to identify and report critical incidents after watching the tape (user-subject comments: "Although the critical incident examples [shown on the tape] were fairly common sense, I found the video useful in finding incidents that I might normally not report and just struggle through time after time.)" To investigate further this matter, a future study will be conducted to determine the effect of training in the quality of critical incident reports.

Other expectations, results, and discussion

As illustrated in Figure 7-31, 17 out of 18 (or 94%) user-subjects agreed that it would be useful in training to practice both identifying and reporting a critical incident using the same application as the one being evaluated during the session.

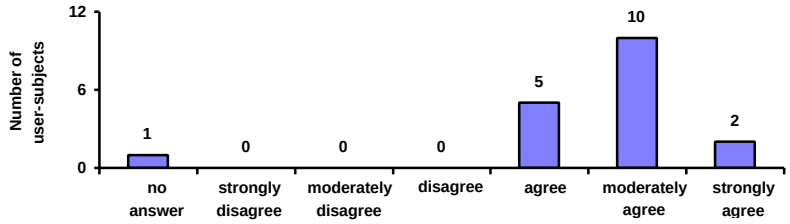


Figure 7-31. User-subjects liked the idea of practicing, in training, both identification and reporting a critical incident with the same application being evaluated

Some user-subjects indicated that it was confusing having on the videotape only one person playing the role of both user and narrator (i.e., usability expert). One user-subject suggested that the video training should include both explanation and practice on identifying as well as on reporting critical incidents.

Across both groups, 20 (or 83%) of all 24 user-subjects indicated that the training helped them learn to identify critical incidents (Figure 7-32).

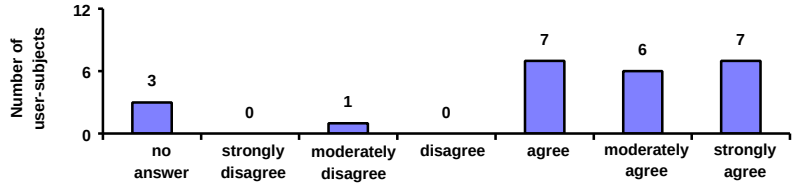


Figure 7-32. User-subject indication that the training helped them learn to recognize critical incidents

Across both groups, 22 (or 92%) of all 24 user-subjects indicated that the training provided enough information (Figure 7-33).

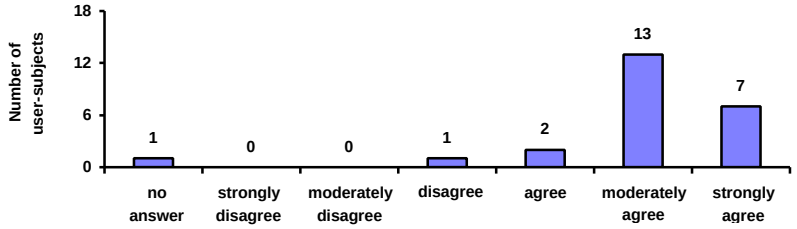


Figure 7-33. Indication from user-subjects that the training provided enough information

Across both groups, 23 (or 96%) of the 24 user-subjects indicated that the training was easy to follow (Figure 7-34).

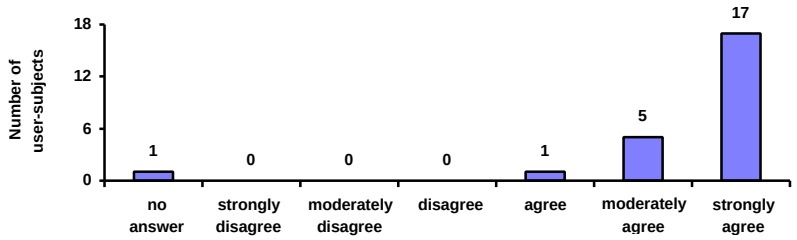


Figure 7-34. User-subject indication that the training was easy to follow

Lessons learned

The main advantage of using videotape as the medium for conveying training material is that all users receive exactly the same presentation. This training material can be made more modular (e.g., one video clip per type of usability problem) and more broadly available by placing video clips on the Web, allowing users to obtain self-paced training by downloading the clips.

As Wiedenbeck, Zila, and McConnell (1995) point out, hands-on practice during training may contribute to better user task performance. This motivates a future study to investigate further the effect of the practice part of the training (i.e., practice identifying and reporting a critical incident, preferably using the same application being evaluated) in user task performance.

Possible ways to improve the training include the following:

- the training videotape should include examples of a user both identifying and reporting critical incidents during task performance (instead of only identifying critical incidents), two different people should play the roles of user and narrator respectively, and
- training should more strongly emphasize that users should click the *Begin Incident Capture* button (Figure 7-7) immediately after a critical incident occurs in order to record useful critical incident data.

7.3.4 Verbal protocol issues

One problem exposed in the feasibility case study (Section 5.7) was the need to prompt users continually for verbal protocol to establish task context for the critical incidents. For the most part, verbal protocol was essential for the expert-subjects to establish for each critical incident what the task was, what the use was trying to do, and why the user was having trouble.

In the exploratory study, however, because it is very difficult to prompt users for verbal protocol in real world remote conditions and because critical incident reports capture information about the task context, user-subjects were not prompted for verbal protocol. Nonetheless, a microphone was installed to record the user's voice (in the audio portion of the videotape) during the evaluation session in case of spontaneous verbal comments.

Expectation #34: Evaluator-subjects will not consider verbal comments essential in analyzing critical incident reports.

Two users-subjects made comments that showed signs of frustration (i.e., tapping fingers on table, "Oh my God!", "Ugh!") and confusion (e.g., "It wasn't supposed to do that"; "I still don't know what I'm doing"). Verbal explanations from another participant helped the experimenter better understand a critical incident rather than just from watching the videotape without audio. The user-subject, who was searching for a particular movie title on the database, typed the movie name at a *Find* text box. Next, the user-subject had to choose between two radio buttons (*substring* and *fuzzy search*) to determine a specific search method (user-subject comment: "Hum!"). The user-subject selected the *fuzzy search* radio button (user-subject comments "Well, I don't know what fuzzy search means.") and clicked the *Search* button. The result from that search was a system message remarking that the title was not found on the movie database. After reading the system message, the user-subject clicked a *tips* link to read more information about searching hints but did not find it very useful (user-subject comment: "That didn't help me either!").

Clip-and-report evaluator-subjects did not agree about the usefulness of verbal protocol in analyzing critical incident reports:

> Clip-and-report evaluator-subject #1:
>
> "Not having verbal protocol made it difficult [to understand the critical incidents]".

Clip-and-report evaluator-subject #2:

"[The audio on the tape was] totally useless; there wasn't any [verbal input], except for typing and mumbling."

Verbal protocol did not play a vital role in determining critical incidents in this study. User-subjects were not particularly encouraged to talk aloud during task performance and only three user-subjects gave spontaneous comments. Verbal protocol might still be considered valuable in identifying critical incidents and future work should include finding an effective technique for prompting users to provide verbal explanations of the critical incidents encountered during task performance.

7.3.5 Role of audio in reporting critical incidents

Verbal descriptions of critical incidents are an alternative for entering text in a dialogue box (e.g., *Remote Evaluation Report* window). Verbal descriptions (Ericsson and Jones, 1990), which use a non-visual output channel, might not interfere with task performance as much as typing does. Further, reporting critical incidents verbally would normally take less time than typing. Disadvantages of verbal reporting include the cost of adding adequate audio equipment (e.g., microphone) for recording verbal comments and the possibility of interfering with tasks of other users located in quiet environments (e.g., library, cubicles).

Expectation #35: For the most part, user-subjects may indicate a preference to report~~ing~~ critical incidents verbally rather than typing.

In the feasibility case study (Section 5.5), user-subjects gave a verbal description of the critical incident that was captured on an audio track of the videotape. Those user-subjects indicated that providing verbal descriptions did not interfere with their tasks. Consequently, it was expected in this study that the majority of user-subjects would like the idea of reporting critical incidents verbally (e.g., recording comments using a microphone) rather than typing them. However, only 11 (or 46%) of all 24 user-subjects agreed that they preferred to provide verbal critical incident reports (Figure 7-35). Despite these results, a future study should explore use of audio in the user-reported critical incident method with other kinds of users, like telephone operators (Muller et al., 1995), who would normally use a non-visual output channel for communicating during their everyday work and possibly would benefit more from reporting critical incidents verbally.

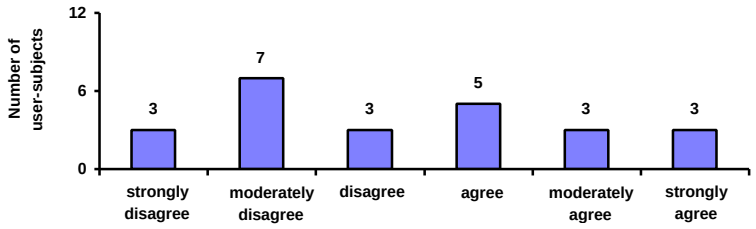


Figure 7-35. User-subject preference for reporting critical incidents verbally

7.3.6 Role of video in the study

For obtaining video clips in contextualized critical incident reports, the experimenter considered using UCDCam (Section 2.2.5) to record screen-sequence video clips containing critical incident data. But, as learned from the feasibility case study, to provide the maximum raw data in experimental studies, it is more useful to use normal continuous video to capture user actions during the entire evaluation session. For that reason, for each user-subject in the study, the experimenter recorded all the screen activity via a scan converter and videotape deck.

Expectation #36: Videotaping of user-subject screen activity during the entire session will be helpful in obtaining relevant information (e.g., severity of critical incidents missed by user) that could have been missed by only observing screen-sequence clips.

Reviewing videotapes of the entire evaluation session was essential to gather the necessary data about the critical incidents encountered during the evaluation session. It was necessary for the experimenter to review all videotapes (24 one-hour tapes) to tag and code all critical incidents, and brief, selected video clips would have been grossly inadequate for this task.

Watching the videotapes was not enough to understand the exact problems that some user-subjects experienced. In two cases, critical incidents were observed by the experimenter and reported by user-subjects, but user-subjects reported the critical incident to be something else than what the experimenter had expected. In one situation the experimenter thought numerous network delay messages were disturbing a user-subject, but what the user-subject reported was wanting to hit the "Return" or "Enter" key to activate a search query rather than clicking the "Search" button on the interface. In eight other cases, user-subjects reported low severity critical incidents, but the experimenter did not perceive them to be critical incidents from watching the tapes. In such cases laboratory-based usability evaluation has the advantage that the evaluator can engage them in dialogue and elicit more understanding about the situation. However, these cases are mostly limited to low severity problems.

As mentioned by one evaluator-subject who watched video clips to analyze critical incident data, one drawback of using the scan converter was that the computer's screen resolution had to be lowered to 640x480 pixels to satisfy the maximum available resolution for the scan converter: "The entire screen was not visible to me because of the scan converter clipping the image. I would have missed this critical incident if not for the report (e.g., scrollbar movement was not visible)." Lowering the screen resolution increased the size of all objects (e.g., windows, text), making some objects too long to fit on the screen and making it difficult for experimenter to determine what some user-subjects where typing, clicking, or pointing at.

In sum, the overall data obtained from the videotapes gave good insight about the user-reported critical incident method. The study yielded a wide variety of useful data that would have been missed by only observing screen-sequence video clips. Thus, for experiments it is essential to use continuous video for recording the entire usability evaluation session as a baseline for comparison, rather than only storing screen-sequence video clips for each critical incident.

For deployment within the user-reported critical incident method in remote user locations, it is more cost-effective (for both data collection and analysis) to use software tools (e.g., UCDCam) for capturing screen usage and collecting information from users. Instead of having real

evaluators spend long hours videotaping users, analyzing the data, and editing the observational tapes into video clips, it is more effective for them to receive automatically recorded clips focused on critical incidents.

Adding a video camera to the setup of the user-reported critical incident method for recording the user's face would not be cost-effective because it would make the data gathering process more costly and complex (e.g., adding hardware and software, larger network traffic caused by increased size and number of video clips, longer time to analyze critical incident information), while adding less useful information.

7.3.7 Storing and communicating critical incident data

The normal mode of operation for the remote-reporting mechanism is having users connected online while using the application being evaluated. When the user is not connected online, the critical incident reporting tool can store contextualized critical incident reports (i.e., textual reports and screen-sequence video clips) locally in the user's computer. When the user next goes online, the tool can automatically download reports and video clips to the evaluator's computer or send them as email attachments. Further, the reporting tool can also wait until network delay peak hours are over (e.g., midnight) to send the reports.

7.3.8 Issues relating to the proximity of a critical incident and its cause

Koenemann-Belliveau et al. (1994) claim that the cause of some critical incidents can be far from the occurrence (in time and task thread steps). This claim would seem to work against the user-reported critical incident method, implying that contextualized critical incident data does not always reveal the cause of a critical incident. However, the objective of the method is only to identify usability problems of the user interface, and is the responsibility of the evaluator (e.g., usability expert) to find what causes them. Future work will be conducted on approaches to capture the contextualized critical incident data in a way that facilitates analysis and deduction of the causes of usability problems.

7.4 SUMMARY OF RESULTS AND LESSONS LEARNED

Results indicated that users, even when working in their daily job environment and lacking interaction with evaluators, are capable of self-reporting high, medium, and low severity critical incidents encountered during task performance. Results also showed that users can rate self-reported critical incidents with reasonable accuracy compared to an expert evaluator.

The user-reported critical incident method includes automatic and continuous scan-converted (or digital) video capture of screen activity during task performance. From this, the system extracts contextualized critical incidents, short video clips of screen activity just moments preceding the point at which a critical incident is reported. It was expected that these clips, when reviewed by an evaluator in conjunction with reading the reports, to help explain the critical incidents. However, because of the reporting delay just described, video clips often were irrelevant to the critical incident. A solution to this problem is to de-couple the time of critical

incident occurrence from the time of reporting, and to associate the video clip with the occurrence itself to ensure that the clip contains data relevant to the critical incident. In this approach, users would click a button (e.g., *Begin Incident Capture*) when they believe they are beginning to experience a critical incident. The retrospective video clip would be captured at this point. Users would subsequently click on a *Report Incident* button when ready to complete a detailed report of the critical incident.

Another expectation was that users would want to report critical incidents anonymously, but they indicated they did not mind being identified with their reports, if, in a real-world setting, it meant they could receive acknowledgment of receipt of their reports, plus feedback from evaluators.

Interestingly, the experimenter discovered by manual inspection of data that some of the users who gave the longest and most detailed reports also said they felt that reporting critical incidents did not interfere much with performing the tasks. The experimenter had feared that self-reporting might be perceived as burdensome to users.

Results indicated, in sum, that users could, in fact, with minimal training, recognize and report critical incidents effectively, that they could rank their severity reasonably, and that they did not find this self-reporting to interfere with getting real work done.

CHAPTER 8:
SUMMARY

8.1 BACKGROUND

Although existing lab-based formative usability evaluation is frequently and effectively applied to improving usability of software user interfaces, it has limitations. Project teams want higher quality, more relevant, usability data – more representative of real world usage. The ever-increasing incidence of users at remote and distributed locations (often on the network) precludes direct observation of usage. Further, transporting users or developers can be very costly. As the network itself and the remote work setting have become intrinsic parts of usage patterns, the users' work context is difficult or impossible to reproduce in a laboratory setting. These constraints and requirements led to extending usability evaluation beyond the laboratory to the concept of remote usability evaluation, typically using the network itself as a bridge to take interface evaluation to a broad range of users in their natural work settings.

Perhaps the most significant impetus for remote usability evaluation methods, however, is the need for a project team to continue formative evaluation downstream, after implementation and deployment. Most software applications have a life cycle extending well beyond the first release. The need for usability improvement does not end with deployment, and neither does the value of lab-based usability evaluation, although it does remain limited to tasks that developers believe to represent real usage. Fortunately, deployment of an application creates an additional source of real-usage usability data. However, these post deployment usage data are not available to be captured locally in the usability lab. Motivated by this problem, this work was based on developing a method for employing users to capture these detailed data (i.e., critical incident data) remotely during task performance – the kind of data required for formative usability evaluation.

8.2 DEFINITION OF REMOTE USABILITY EVALUATION

Remote evaluation is defined as usability evaluation where evaluators are separated in space and/or time from users (Hartson et al., 1996). For consistency of terminology throughout this thesis, the term *remote*, used in the context of remote usability evaluation, is relative to the developers and refers to users not at the location of developers. Similarly, the term *local* refers to location of the developers. Sometimes developers hire outside contractors to do some usability evaluation in a usability laboratory at the contractor's site. Neither term (local or remote) per the above definitions, applies very well to these third-party consultants, but they (as surrogate developers) could have remote users.

8.3 GOAL OF THIS WORK

Because of the vital importance of critical incident data and the opportunity for users to capture it, the over-arching goal of this work is to develop and evaluate a remote usability evaluation method for capturing critical incident data and satisfying the following criteria:

- tasks are performed by real users,
- users are located in normal working environments,
- users self-report own critical incidents,
- data are captured in day-to-day task situations,
- no direct interaction is needed between user and evaluator during an evaluation session,
- data capture is cost-effective, and
- data are high quality and therefore relatively easy to convert into usability problems.

Several methods have been developed for conducting usability evaluation without direct observation of a user by an evaluator (see Section 3.2 entitled "Types of remote evaluation methods"). However, none of these existing remote evaluation methods (nor even traditional laboratory-based evaluation) meets all the above criteria. The result of working toward this goal is the user-reported critical incident method, described in this thesis.

8.4 APPROACH

The over-all goal of developing and evaluating a new method for remote usability evaluation is comprised of several steps, each representing a substantial project on its own:

1. feasibility case study to explore relevant issues, develop the operative research questions;
2. development of the user-reported critical incident method;
3. extensive exploratory study of the method to gain understanding and insight about the method;
4. controlled laboratory-based experiments validating research hypotheses; and
5. field studies conducted in real work environments, accounting for work context factors (e.g., noise, interruptions, multi-thread task performance).

Steps 1, 2, and 3 comprise the work completed for this thesis. The first step, the case study described in Chapter 5, was reported in Hartson et al. (1996), where the method was called semi-instrumented critical incident gathering. The objective of this step was to judge feasibility of the method — determining if this new method could provide approximately the same quantity and quality of qualitative data that can be obtained from laboratory-based formative evaluation, and determining if this is possible in a way that was cost-effective for both evaluator (e.g., minimal resources for data collection and analysis) and user (e.g., minimal interference with work). Based on the insights gained from the case study, in the second step a new method for conducting remote usability evaluation was developed called the user-reported critical

incident method. Step 3 is an in-depth exploratory study reported in Chapters 4, 6, and 7. Steps 4 and 5 are reserved for future work.

8.5 DESCRIPTION OF THE USER-REPORTED CRITICAL INCIDENT METHOD

The user-reported critical incident method is a usability evaluation method that involves real users located in their own working environment, doing everyday tasks, and reporting critical incidents (after receiving minimal training) without direct interaction with evaluators. Critical incident reports are augmented with task context in the form of screen-sequence video clips and evaluators analyze these contextualized critical incident reports, transforming them into usability problem descriptions.

8.6 EXPLORATORY STUDY

8.6.1 Objectives

As mentioned in Section 8.4, the exploratory study in step 3 was performed. The study is described as exploratory because, although quantitative data was obtained, it was not the kind of summative study that uses statistically significant results to prove or refute an experimental hypothesis. Rather, it was an exploratory study to gain insight and understanding, under practical operating conditions, about the strengths and weaknesses of the method. In particular, the objectives were:

Objective 1: Investigate feasibility a nd effectiveness of employing users to identify and report their own critical incidents during usage.

Objective 2: Investigate feasibility and effectiveness of transforming remotely gathered critical incident data into usability problem descriptions.

Objective 3 Gain insight into various parameters associated with the user-reported critical incident method.

Objective 1 translates to the following research question: Can users report their own critical incidents and how well can they do it? In this study, Objective 1 is divided into the following sub-objectives:

1. Explore issues about user-subject performance in identifying and reporting critical incidents:
    - User-subject ability to identify and report critical incidents during task performance
    - User-subject activity sequencing and timing in reporting critical incidents
    - Level of time and effort required to report critical incidents
    - User-subject ability to rate severity of critical incidents

- User-subject ability to identify high severity critical incidents as well as low and medium severity critical incidents

2. Obtain subjective data about user-subject perceptions, preferences, and attitudes towards remotely-reporting critical incidents:
    - User-subject attitudes towards remotely-reporting critical incidents
    - User-subject preferences with respect to reporting critical incidents anonymously
    - User-subject perceptions with respect to interference with user tasks
    - User-subject preferences relating to reporting negative and positive critical incidents

Objective 2 translates to the following research question: Can evaluators use critical incident data to produce usability problem descriptions and how well can they do it? In this study, Objective 2 is divided into the following sub-objectives:

- Ability of evaluator-subjects to analyze critical incident data
- Role of textual reports in data analysis
- Role of video in data analysis
- Role of audio in data analysis
- Time and effort required to analyze critical incident data
- Level of agreement with user-subject critical incident severity ratings

Objective 3 translates to the following research question: What are the variables and values that make the method work best? In this study, Objective 3 is divided into the following sub-objectives:

- Preferred location for the critical incident reporting tool
- Using the *Remote Evaluation Report* window
- Role of training in reporting critical incidents
- Verbal protocol issues
- Role of audio in reporting critical incidents
- Role of video in the study
- Packaging critical incident data
- Issues relating to the proximity of a critical incident and its cause

8.6.2 Design, results, and lessons learned

The exploratory study was divided in two phases: critical incident gathering (Phase I in Section 6.3) and transformation of critical incident data into usability problem descriptions (Phase II in Section 6.4). Results from Phase I (Section 7.1) revealed that users, with only brief training can identify, report, and rate the severity level of their own critical incidents. Observations also indicated a need for much more flexibility in structure, quantity, format, and timing of reports than expected.

Screen-sequence video clips accompany textual reports as visual context to obtain more information about each critical incident. Automatic capture of these clips requires some trigger mechanism to initiate the capture process. The clicking by the user on the *Report Incident* button was intended to be this trigger mechanism. However, the most significant observation in the study was a highly variable, and often large, delay between the time when the video clip should begin (at the clear onset of a critical incident) and the time when user-subjects click the *Report Incident* button. The result is often video clips that contain user actions irrelevant to the critical incident. Thus, this study reveals that a different trigger mechanism, separate from the *Report Incident* button, is required.

User-subjects who voluntarily participated in Phase I of the study were university students and not real application users working on real tasks in their normal working environment. To find user-subjects representative of a large population of users, the experimenter administered a background questionnaire to students. All 24 user-subjects came from a variety of academic disciplines and had minimum knowledge of Web browsing and Web-based information retrieval. No experience was required with the Internet Movie Database, the application evaluated in the study. User-subjects performed six search tasks predefined by the experimenter. The experimenter selected these tasks considering what he considered representative usage (i.e., search tasks) of the Internet Movie Database by real users. We find students to be credible representative users of a movie database, and we have no reason to believe that either their behavior or their performance would be significantly different from other, non-student, users.

In real life situations, users located in their natural work setting lack the means to contact evaluators (or application developers) when critical incidents occur. To simulate this lack of interaction in the study, user-subjects were isolated and told they were on their own during the evaluation session. Despite the fact that the study was not conducted in a real remote evaluation situation with real users, real tasks, and a real remote working environment, the experiment considers that results identified in this thesis are valid and that user-subjects did not provide gratuitous critical incident reports sent to please the experimenter.

As a general matter, the small number of evaluator-subjects and individual differences among them strongly colored the results of Phase II of the study, presumably contributing to the low significance of its results. Therefore, the “results” obtained are hypotheses and not conclusions, and will serve as expectations for a future study.

8.7 REMOTE VERSUS LOCAL USABILITY EVALUATION

Will remote evaluation ever replace the traditional lab-based usability evaluation? Remote usability evaluation methods are not a direct replacement for lab-based evaluation. Interactive software development groups have large investments in usability labs, employing established and effective evaluation process. The face-to-face contact between users and evaluators in a traditional usability evaluation setting is crucial to help identify critical incidents that would otherwise be missed by only watching video clips or reading reports. It is also essential in eliciting verbal protocol data. However, these high quality data are produced at a high cost.

Additionally, lab-based evaluation is the only realistic alternative for early prototypes, before a working version is available remotely.

However, once the earliest version of a software system is deployed, remote evaluation methods pick up where lab-based evaluation often leaves off. Perhaps, 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. Most software applications have a life cycle extending well beyond the first release. The need for usability improvement does not end with deployment, and neither does the value of lab-based usability evaluation, although it does remain limited to tasks that developers believe to represent real usage. Deployment of an application creates an additional source of real-usage usability data, however, these post deployment usage data are not available to be captured locally in the usability lab. Fortunately, the user-reported critical incident method can be employed to capture these data and at a lower cost than lab-based evaluation.

Further, as many of the basic and more obvious usability problems have been removed via the usability lab process and software becomes deployed, project teams seek higher quality usability data, more relevant to, and more representative of, real world usage. The ever-increasing incidence of users at remote and distributed locations (often on the network) precludes direct observation of usage. As the network itself and the remote work setting have become intrinsic parts of usage patterns, the users' work context is difficult or impossible to reproduce in a laboratory setting. Further, transporting users or developers to remote locations can be very costly. These barriers led to extending usability evaluation beyond the laboratory to the concept of remote usability evaluation, typically using the network itself as a bridge to take interface evaluation to a broad range of users in their natural work settings.

Finally, remote evaluation methods have the potential to vastly improve other post-deployment activities such as field support and customer help lines. While these activities have a significant impact on overall usability in the long term, possibly because they are not perceived as glamorous as the up-front design and evaluation activities, they have not received much attention from the HCI community. Remote usability evaluation can provide a remedy to this, tying these activities back to the usability development cycle.

CHAPTER 9:
FUTURE WORK

The topics for future work are introduced in approximate order of chronology.

9.1 REFINING THE METHOD

The future work activities in the following subsections are partly to follow-up on study results, partly to extend method.

9.1.1 Role of video for contextual data in critical incident reports

In general, it was expected that evaluator-subjects would consider video clips to be a necessary supplement to written reports in understanding the task context leading up to the critical incident. Although the results of this study do not support this expectation, the experimenter strongly felt that this small amount of data was not sufficient to rule out those expectations, and this issue should be of high priority to explore in a future study. Therefore, a formal study will be conducted to investigate the usefulness of video clips in analyzing critical incident data.

This study will require two kinds of participants: user-subjects (not trained in usability methods) and evaluator-subjects (trained in human-computer interaction and usability methods). User-subjects will perform representative tasks and report critical incidents during task performance. Evaluator-subjects will analyze contextualized critical incident data in the form of textual critical incident reports and video clips to provide subjective feedback about:

- whether evaluator-subjects prefer having both textual reports and video clips for data analysis, over having only one of these two;
- effort required to map a video clip to the corresponding critical incident report;
- time and effort required to analyze both video clips and critical incident reports to create a list of usability problem descriptions; and
- usefulness of video in helping to understand the critical incident.

9.1.2 Evaluating the new video clip trigger redesign

Assuming that video clips are useful to convey critical incident data, a formal study will be conducted to evaluate the video clip trigger mechanism (i.e., *Begin Incident Capture* button) of the critical incident reporting tool (Figure 7-7). This study will require two kinds of participants: user-subjects and evaluator-subjects.

During the evaluation sessions, the computer screen will be recorded using continuous video while user-subjects perform representative tasks. Users will click the *Begin Incident Capture*

button at the inception of each critical incident (i.e., when an occurrence of difficulty is beginning), even though information may yet be insufficient for effective reporting. Users who understand how the mechanism works can even "re-enact" a critical incident for capture. Knowing that this button triggers recording of screen-sequence video clips, users can do something on the screen that they want the evaluator to see on the clip and then click the button. Later, when ready, users will click the *Send Full Report* button to send a detailed report of the incident. Following the evaluation session, user-subjects will complete a questionnaire to describe their experience as remote users, including information about their perceived ability to recognize (i.e., clicking *Begin Incident Capture* button) and report critical incidents (i.e., clicking *Send Full Report* button).

For each user-subject, evaluator-subjects will watch a full videotape with task performance to determine the exact time of reporting critical incidents (i.e., during or after task performance). After watching all videotapes, evaluator-subjects will then watch screen-sequence video clips for each user-subject to provide subjective feedback about the relevance of critical incident data contained on the clips and their usefulness for creating usability problem descriptions.

9.1.3 Determining the optimal length and starting point for a critical incident video clip

A formal case study will be conducted to determine the optimal length and starting point for producing critical incident video clips of good quality (i.e., containing useful information about the critical incident). This study will require two different types of participants: user-subjects and expert-subjects (trained in human-computer interaction and usability methods). After receiving critical incident training, user-subjects will perform a set or representative tasks and remotely-report critical incidents. The computer screen for each user-subject will be videotaped during the entire evaluation session (e.g., 45 min. of videotape per user-subject), and after all user-subjects are videotaped, a panel of expert-subjects will review the tapes to determine a good approximation for the best length and starting point for producing critical incident video clips. After deciding the optimal length and starting point for a video clip, a study will be conducted to determine a way to automate capture of screen-sequence video clips (e.g., using UCDCam).

9.1.4 Critical incident training

Success of the user-reported critical incident method depends on the ability of typical users to recognize and report critical incidents effectively, but there is no reason to believe that all users have this ability naturally. This study was designed with the assumption that some training would be required — 12 user-subjects received extensive explanation on identifying critical incidents, all 24 user-subjects received a brief explanation on identifying and reporting critical incidents, and a practice on reporting critical incidents.

Results in Section 7.3.3 seem to indicate that there was no significant difference in the quality and quantity of critical incident reports among all user-subjects, however, further research is necessary to investigate whether users need to be trained to identify and report critical incidents effectively. If results of this research reveal the need for training, a study can be conducted to:

- determine the appropriate content and amount of critical incident training (i.e., provide information about both identifying and reporting critical incidents or only about identifying critical incidents, emphasize that users should click the *Begin Incident Capture* button immediately after a critical incident occurs in order to record useful critical incident data), and
- investigate further the effect of the practice part of the training (i.e., practice identifying and reporting a critical incident, preferably using the same application being evaluated) in user task performance.

Videotape has proven to be a consistent medium for conveying training to users, employing two different people playing the roles of user and narrator respectively. In the future, critical incident training can also be developed as a series of video clips that a broader range of users can access via a Web browser. With this kind of self-paced training users will be capable of taking training at their own speed and downloading clips for different types of usability problems.

9.1.5 Real users doing real tasks at their normal working environment

Once the new video clip trigger mechanism is evaluated and critical incident training is redesigned, the next step is to conduct a formal study to confirm results found in this thesis in a real remote usability evaluation setting. This study will require user-subjects and evaluator-subjects. User-subjects will be real application users, working on day-to-day tasks in their normal working environment. During the study, user-subjects will be asked to identify and report critical incidents during task performance using a *Remote Evaluation* window. Critical incident reports will be sent to evaluator-subjects, whose role is to analyze these reports into usability problem descriptions and compare results found with those reported by this thesis.

9.1.6 Comparison of verbal and textual critical incident reporting

A formal study will be conducted to investigate user preference in verbal or textual critical incident reporting. This study will require two kinds of participants: user-subjects and evaluator-subjects. First, user-subjects will perform a set of representative tasks and report critical incident using a textual form. The same group of user-subjects (i.e., within-subjects design) will also perform a similar set of tasks but this time reporting critical incidents verbally. During task performance, the experimenter will determine the time and effort of user-subjects in reporting critical incidents under both conditions. After the experimental session is conducted, each user-subject will complete a questionnaire to indicate his or her preference in reporting critical incidents.

A group of evaluator-subjects will receive both kinds of critical incident reports (verbal and textual) and analyze them to create a list of usability problem descriptions for each condition. Finally, evaluator-subjects will complete a questionnaire to indicate:

- the time an effort required to analyze both kinds of data,
- their preference about analyzing textual or verbal critical incident reports, and/or

- the usefulness of verbal and textual data (i.e., which condition provides data of higher quality and accuracy?).

9.2 COMPARISONS WITH OTHER METHODS

9.2.1 Comparison of the user-reported critical incident method with othe r remote evaluation techniques

Empirical studies can be similarly conducted to compare the user-reported critical incident method with the various remote evaluation techniques described in Section 3.2. The purpose of each study will be to investigate advantages, disadvantages, similarities, and differences between the user-reported critical incident method and the corresponding remote evaluation technique. Every study will require three types of participants: user-subjects (performing representative tasks), evaluator-subjects (creating usability problem descriptions), and expert-subjects (using usability problem descriptions and other kinds of data to compare the user-reported critical incident method with the respective remote evaluation technique).

9.2.2 Comparison of the user-reported critical incident method with traditional laboratory-based usability evaluation

An empirical study will be conducted to compare the quality and correctness of usability problem descriptions obtained from the user-reported critical incident method with usability problems descriptions obtained from laboratory-based evaluation, the traditional yardstick for comparison with new usability methods. This study will involve three types of participants: user-subjects, evaluator-subjects, and expert-subjects.

User-subjects and evaluator-subjects will be divided into two groups (i.e., remote and laboratory-based condition). In the remote condition, a group of user-subjects will identify and report critical incidents during task performance. Evaluator-subjects for the remote condition will independently analyze contextualized critical incident reports sent by user-subjects to create a list of usability problem descriptions. In the local evaluation condition, another group of user-subjects will talk-aloud aloud during task performance, while different evaluator-subjects independently create a list of critical incidents. These evaluator-subjects will later analyze critical incident data obtained during the evaluation sessions (e.g., videotape, evaluator notes) to create a list of usability problem descriptions. As a final step, a panel of expert-subjects will compare the lists of usability problem descriptions obtained from evaluator-subjects of both conditions to determine similarities and differences among them.

9.3 FURTHER FUTURE WORK

9.3.1 The user-reported critical incident method in the software life cycle

A qualitative study will be conducted to investigate the place of the user-reported critical incident method in the overall software and user interface life cycles and determine which types of

applications are more suitable to be evaluated remotely. A survey will be sent to a number of usability specialists, human-factor practitioners, software engineers, and managers to obtain subjective data about this subject matter.

In general, the user-reported critical incident method could be applied in the following stages, to which results produced by the method will be different from one software stage to another:

- early formative evaluation;
- alpha, beta, and other field usability evaluation;
- post-deployment;
- field and customer support; and
- marketing strategies.

One complete untapped area that could benefit from remote usability evaluation is customer support. In this regard, a new whole domain of applications of the user-reported critical incident method can be explored, for example, to allow customers to talk about very specific points of usability problems of software, instead of just generalities. It also has potential to get feedback from evaluators back to the users.

9.3.2 Usability Problem Classifier

The Usability Problem Classifier (van Rens, 1997), a method for classifying usability problems, provides evaluators with structured guidance in creating usability problem descriptions for problem reporting. This guidance ensures that descriptions are complete and well-structured, thereby improving their quality. A study will be conducted to explore the possible integration of the Usability Problem Classifier with the user-reported critical incident method. This study will require two types of participants: user-subjects and evaluator-subjects. User-subjects will perform representative tasks and report critical incidents encountered during task performance. Evaluator-subjects, on the other hand, will analyze critical incident data and use the Usability Problem Classifier to create usability problem descriptions. Results of this study will also help determine what additional information is needed from the user-reported critical incident method to support high quality usability problem reporting and complete classification.

9.3.3 Severity rating

In this study (Section 7.1.8), many user-subjects indicated that it was somewhat easy to rate the severity of critical incidents. However, some user-subjects still found various difficulties in rating critical incidents using a 5-point scale. Previous researchers used 6-point and 7-point scales to indicate critical incident severity, but this study showed that it is quite difficult to make a comprehensive and reliable scale. In fact, instead of rating critical incidents by severity, the ultimate goal is to find the importance to fix of critical incidents, which is influenced by several factors (e.g., impact on satisfaction, impact on performance, severity of errors caused by the critical incident). For that reason, a study will be conducted to create a comprehensive way to determine the importance to fix of critical incidents by allowing users to indicate the degree to

which each factor influenced their task performance (e.g., did not impact satisfaction, unable to complete task, high severity critical incidents).

REFERENCES

Abelow, D. (1993). Automating Feedback on Software Product Use. *CASE Trends*, 15-17.

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

Ashlund, S., and Hix, D. (1992). *IDEAL: A Tool to Enable User-Centered Design.* Proceedings of the CHI Conference on Human Factors in Computing Systems (Posters and Short Talks Supplement to Proceedings), 119-120.

Bowers, V.A., and Snyder, H.L. (1990). *Concurrent Versus Retrospective Verbal Protocol for Comparing Window Usability.* Proceedings of the 34th Annual Meeting of the Human Factors Society, 1270-1274.

Carroll, J.M. (1984). Minimalist Training. *Datamation, 30*(18), 125-136.

Carroll, J.M. (1990). *An Overview of Minimalist Instruction.* Proceedings of the 23rd Annual Meeting of the Human Factors Society. (IEEE Computer Society Press Reprint, December 1994, 210-219).

Carroll, J.M., Koenemann-Belliveau, J., Rosson, M.B., and Singley, M.K. (1993). *Critical Incidents and Critical Themes in Empirical Usability Evaluation.* Proceedings of the HCI-93 Conference, 279-292.

Compaq Corporation. (1997). *Compaq Carbon Copy*. Internet WWW page at: http://www.microcom.com/products/sas/cc/index.html [Accessed:7/1/97].

del Galdo, E.M., Williges, R.C., Williges, B.H., and Wixon, D.R. (1986). *An Evaluation of Critical Incidents for Software Documentation Design.* Proceedings of the 30th Annual Human Factors Society Conference, 19-23.

Dzida, W., Wiethoff, M., and Arnold, A.G. (1993). ERGOGuide: The Quality Assurance Guide to Ergonomic Software . Joint internal technical report of GMD (Germany) and Delft University of Technology (The Netherlands).

Elgin, B. (1995). Subjective Usability Feedback from the Field over a Network. *SIGCHI Bulletin, 27*(4), 43-44.

ErgoLight™ Usability Software. (1997). *ErgoLight™ Usability Software Home Page*. Internet WWW page at: http://www.ergolight.co.il/ [Accessed:7/1/97].

Ericsson, K.A., and Jones, R.E. (1990). Verbal reports as data. *Psychological Review, 87*, 215-251.

Fitts, P.M., and Jones, R.E. (1947). Psychological Aspects of Instrument Display: Analysis of Factors Contributing to 460 "Pilot Error" Experiences in Operating Aircraft Controls. In

H.W. Sinaiko (Ed.), *Selected Papers on Human Factors in the Design and Use of Control Systems (1961)* (pp. 332-358). New York: Dover Publications, Inc.

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

Fleming, S.T., Kilgour, A.C., and Smith, C. (1993). *Computer Support for Evaluation Studies.* Proceedings of the INTERCHI'93 Conference on Human Factors in Computing Systems, 201-202.

Goodman, D. (1996). *JavaScript Handbook*. California: IDG Books Worldwide, Inc.

Hameluck, D.E., and Velocci, V.V. (1996). Buffered screen capturing software tool for usability testing of computer applications . Submitted by IBM Canada Limited to Canadian Patent Office (CA9-96-016).

Hammontree, M., Weiler, P., and Nayak, N. (1994). Remote Usability Testing. *Interactions*, 21-25.

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

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

Hix, D., and Hartson, H.R. (1994). *IDEAL: An Environment for User-Centered Development of User Interfaces.* Proceedings of the EWHCI'94: Fourth East-West International Conference on Human-Computer Interaction, 195-211.

Johansen, R. (1988). *Groupware: Computer Support for Business Systems*. New York: The Free Press.

Koenemann-Belliveau, J., Carroll, J.M., Rosson, M.B., and Singley, M.K. (1994). *Comparative Usability Evaluation: Critical Incidents and Critical Threads.* Proceedings of the CHI Conference on Human Factors in Computing Systems, 245-251.

Lewis, C. (1982). *Using the "Thinking-aloud" Method in Cognitive Interface Design* (Research Report RC 9265 (#40713)). Yorktown Heights: IBM Thomas J. Watson Research Center.

Meister, D. (1985). *Behavioral Analysis and Measurement Methods*. New York: John Wiley & Sons, Inc.

Microsoft Corporation. (1997). *Microsoft NetMeeting: Overview*. Internet WWW page at: http://www.microsoft.com/netmeeting/ [Accessed:7/1/97].

Muller, M.J., Carr, R., Ashworth, C., Diekmann, B., Wharton, C., Eickstaedt, C., and Clonts, J. (1995). *Telephone Operators as Knowledge Workers: Consultants Who Meet Customer Needs.* Proceedings of the CHI Conference on Human Factors in Computing Systems, 130-137.

Nielsen, J. (1992). Evaluating the Thinking-Aloud Technique for Use by Computer Scientists. In H.R. Hartsonand H. Hix (Eds.), *Advances in Human-Computer Interaction* (Vol. 3, pp. 69-82). New Jersey: Ablex Publishing Corporation.

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

Ohnemus, K.R., and Biers, D.W. (1993). *Retrospective versus Concurrent Thinking-Out-Loud in Usability Testing.* Proceedings of the 37th Annual Meeting of the Human Factors Society, 1127-1131.

Rubin, J. (1994). *Handbook of Usability Testing*. New York: John Wiley & Sons, Inc.

Shattuck, L.W., and Woods, D.D. (1994). *The Critical Incident Technique: 40 Years Later.* Proceedings of the 38th Annual Meeting of the Human Factors Society, 1080-1084.

Siochi, A.C., and Ehrich, R.W. (1991). Computer Analysis of User Interfaces Based on Repetition in Transcripts of User Sessions. *ACM Transactions on Information Systems, 9*(4), 309-335.

Sun Microsystems. (1997). *ShowMe Product Overview*. Internet WWW page at: http://www.sun.com/products-n-solutions/sw/ShowMe/ [Accessed:7/1/97].

TeamWave Software Ltd. (1997). *TeamWave Workplace Overview*. Internet WWW page at: http://www.teamwave.com/overview.html [Accessed:7/1/97].

van der Meij, H., and Carroll, J.M. (1995, Second Quarter). Principles and Heuristics for Designing Minimalist Instruction. *Technical Communication,* 243-261.

van Rens, L.S. (1997). *Usability Problem Classifier.* Unpublished Master's Thesis, Information Systems, Tilburg University, Tilburg, The Netherlands (work done at the Department of Computer Science, Virginia Tech, Blacksburg, VA, USA).

Vertical Research, I. (1997). *About our Remote Inspection Service*. Internet WWW page at: http://www.vrix.com/serv/rem_insp.htm [Accessed:7/1/97].

Waskul, D., and Douglass, M. (1996). Considering the Electronic Participant: Some Polemical Observations on the Ethics of On-line Research. *Information Society, 12*(2), 129-139.

Whiteside, J., Bennett, J., and Holtzblatt, K. (1988). Usability Engineering: Our Experience and Evolution. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 791-817). Amsterdam: Elsevier North-Holland.

Wiedenbeck, S., Zila, P.L., and McConnell, D.S. (1995). *End-User Training: An Empirical Study Comparing On-line Practice Methods.* Proceedings of the CHI Conference on Human Factors in Computing Systems, 74-81.

WinWhatWhere Corporation. (1997). *Welcome to WinWhat Where: What does W3 do?* Internet WWW page at: http://www.winwhatwhere.com/bsiwww1.htm [Accessed:7/1/97].

Wright, P.C., and Monk, A.F. (1991, December). The use of Thinking-Aloud Evaluation Methods in Design. *SIGCHI Bulletin, 23,* 55-57.

APPENDICES

APPENDIX A: INFORMED CONSENT FORMS

A.1 Informed consent form for user-subjects

I. PURPOSE OF THIS RESEARCH

You are invited to participate in a study that investigates a method for evaluating software remotely. The study involves experimentation for the purpose of evaluating and improving the user interface of the Internet Movie Database. The Internet Movie Database is an international organization whose objective is to provide useful and up to date movie information freely available on-line, across as many systems and platforms as possible. It currently covers over 75,000 movies with over 1,000,000 filmography entries and is expanding continuously. Advertising and sponsorship support their service.

II. PROCEDURES

You will be asked to perform a set of tasks using the Internet Movie Database. These tasks consist of various procedures for searching movie titles and character names on the database. Your role in this test is that of evaluator of the software. We are not evaluating you or your performance in any way; you are helping us to evaluate our system. All information that you help us attain will remain anonymous.

You DO NOT need to complete all the tasks at one sitting. You can take as much time as you need, take breaks between tasks, or even do a few tasks per day. However, please make sure that you finish the tasks by *Friday, March 21, 1997*. After the usability test you will be asked to complete a satisfaction questionnaire about your usage of the system.

The session will last about one hour. There are no risks to you. The tasks are not very tiring, but you are welcome to take rest breaks as needed. If you prefer, the session may be divided into two shorter sessions.

III. RISKS

There are no known risks to the subjects of this study.

IV. BENEFITS OF THIS PROJECT

Your participation in this project will provide information that may be used to improve the usability of the Internet Movie Database. No guarantee of benefits has been made to encourage you to participate. You may receive a synopsis summarizing this research when completed. Please leave a self-addressed envelope with the experimenter and a copy of the results will be sent to you.

You are requested to refrain from discussing the evaluation with other people who might be in the candidate pool from which other participants might be drawn.

V. EXTENT OF ANONYMITY AND CONFIDENTIALITY

The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a subject number will identify you during analyses and any written reports of the research.

The screen actions of your usage will be videotaped. The tapes will be stored securely, viewed only by the experimenters (Dr. H. Rex Hartson, José C. Castillo, Denis Neale, Jonathan Kies), and erased after 3 months. If the

experimenters wish to use a portion of your videotape for any other purpose, they will get your written permission before using it. Your signature on this form does not give them permission to show your videotape to anyone else.

VI. COMPENSATION

Your participation is voluntary and unpaid.

VII. FREEDOM TO WITHDRAW

You are free to withdraw from this study at any time for any reason.

VIII. APPROVAL OF RESEARCH

This research has been approved, as required, by the Institutional Review Board for projects involving human subjects at Virginia Polytechnic Institute and State University, and by the Department of Computer Science.

IX. SUBJECTS RESPONSIBILITIES AND PERMISSION

I voluntarily agree to participate in this study, and I know of no reason I cannot participate. I have read and understand the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project

A.2 Informed consent form for evaluator-subjects

I. PURPOSE OF THIS RESEARCH

You are invited to participate in a study that investigates a method for evaluating software remotely. The study involves experimentation for the purpose of evaluating and improving the user interface of the Internet Movie Database. The Internet Movie Database is an international organization whose objective is to provide useful and up to date movie information freely available on-line, across as many systems and platforms as possible. It currently covers over 75,000 movies with over 1,000,000 filmography entries and is expanding continuously. Advertising and sponsorship support their service.

II. PROCEDURES

You will be asked to study a series of critical incident reports sent by user-subjects. User-subjects made these reports when performing a set of tasks using the Internet Movie Database. The tasks consisted of different procedures of searching for movie titles and character names on the database. You will be asked to analyze the critical incident reports and create a usability problem list. You can take as much time as you want to create the usability problem list, as long as you finish by February 21, 1997. You will also be asked to complete a questionnaire relating to your experience analyzing the critical incident reports.

Your role in this test is that of evaluator of the software. We are not evaluating you or your performance in any way; you are helping us to evaluate our system. All information that you help us attain will remain anonymous.

III. RISKS

There are no known risks to the subjects of this study.

IV. BENEFITS OF THIS PROJECT

Your participation in this project will provide information that may be used to improve the usability of the Internet Movie Database. No guarantee of benefits has been made to encourage you to participate. You may receive a synopsis summarizing this research when completed. Please leave a self-addressed envelope with the experimenter and a copy of the results will be sent to you.

You are requested to refrain from discussing the evaluation with other people who might be in the candidate pool from which other participants might be drawn.

V. EXTENT OF ANONYMITY AND CONFIDENTIALITY

The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a subject number will identify you during analyses and any written reports of the research.

The experiment may be videotaped. If it is taped, the tapes will be stored securely, viewed only by the experimenters (Dr. H. Rex Hartson, José C. Castillo), and erased after 3 months. If the experimenters wish to use a portion of your videotape for any other purpose, they will get your written permission before using it. Your signature on this form does not give them permission to show your videotape to anyone else.

VI. COMPENSATION

Your participation is voluntary and unpaid.

VII. FREEDOM TO WITHDRAW

You are free to withdraw from this study at any time for any reason.

VIII. APPROVAL OF RESEARCH

This research has been approved, as required, by the Institutional Review Board for projects involving human subjects at Virginia Polytechnic Institute and State University, and by the Department of Computer Science.

IX. SUBJECTS RESPONSIBILITIES AND PERMISSION

I voluntarily agree to participate in this study, and I know of no reason I cannot participate. I have read and understand the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project.

APPENDIX B: TASK-RELATED DOCUMENTS

B.1 Search tasks for user-subjects

1. Write down the number of movies whose titles contain the word 'bacon' (e.g., the bacon, bacon and eggs).
2. Find the performer whose character name was Michelle Thomas.
3. Find the titles of the four most recent movies directed by Steven Spielberg.
4. Find the titles of all mystery movies from 1966 to 1967 produced in French.
5. Find the titles of all movies featuring both Robert De Niro and Meryl Streep.
6. Find the titles of all movies in which Billy Crystal acted in the 90's and that are over 2 hours long.

B.2 Participant answer sheet

Instructions:
For each task that you complete, please write your answers at the *Your Answer* column. Remember that we are not evaluating you or your performance in any way; you are helping us to evaluate our system. Don't be concerned if you can not finish a task or get any answers. Please return this sheet to the secretary of the Computer Science department (McBryde 660) with attention to José C. Castillo.

TASK	YOUR ANSWER
1 Write down the number of movies whose titles contain the word 'bacon' (e.g., a bacon, bacon and eggs).	*Number of movies whose titles contain the word 'bacon':*
2 Find the performer whose character name was Michelle Thomas.	*Performer's name:*
3 Find the titles of the four most recent movies directed by Steven Spielberg.	*Movie titles:*
4 Find the titles of all mystery movies from 1966 to 1967 produced in French.	*Titles of all French mystery movies from 1966 to 1967:*
5 Find the titles of all movies featuring both Robert De Niro and Meryl Streep.	*Titles of movies featuring both Robert De Niro and Meryl Streep:*
6 Find the titles of all movies in which Billy Crystal acted in the 90's and that are over 2 hours long.	*Movie titles:*

APPENDIX C: QUESTIONNAIRES

C.1 Background questionnaire for user-subjects

Education

1. What is your academic level at Virginia Tech?

   ❑ Undergraduate student ❑ Graduate student (Masters or Ph.D.)

2. List your major area of study. _____

Computer experience

1. In a typical week, how often do you use a Macintosh computer?

   ❑ Every day ❑ A few times a week ❑ A few times a month ❑ Never

2. In a typical week, how often do you use an IBM style computer or PC?

   ❑ Every day ❑ A few times a week ❑ A few times a month ❑ Never

3. Are you experienced with PCs in a Windows 3.11 environment?

   ❑ Very experienced ❑ Moderate experience ❑ Occasional usage ❑ No experience

4. Are you experienced with PCs in a Windows 95 environment?

   ❑ Very experienced ❑ Moderate experience ❑ Occasional usage ❑ No experience

5. How often do you use Netscape Navigator?

   ❑ Every day ❑ A few times a week ❑ A few times a month ❑ Never

6. Have you ever used any search tool to find any topic of interest on the Web?

   ❑ Every day ❑ Sometimes ❑ Just once or twice ❑ Never

7. Have you ever visited the Internet Movie Database home page?

   ❑ Yes ❑ No

C.2 Post-test questionnaire #1 for user-subjects

The critical incident training provided enough information.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

The training was easy to follow.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

The training helped me learn to recognize critical incidents.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I like the idea of remote-reporting critical incident information to developers.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

Reporting critical incidents anonymously is important for me.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

It was easy to report critical incidents using the *Remote Evaluation Report* window.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

It was easy to determine the severity of critical incidents that I encountered.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I would like to have a *Report Incident* button built into the software applications that I use.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

If possible, I prefer to report incidents verbally (e.g., recording comments using a microphone) rather than typing.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

Reporting critical incidents did not interfere with my tasks.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

Additional comments about your experience as a remote user.

C.3 Post-test questionnaire #2 for user-subjects

Read each statement carefully and indicate **how strongly you agree or disagree** with the statement by putting a mark in the number that applies. If a statement does not apply to you, mark N/A. Thank you!

During the training, I found it useful to practice identifying and reporting a critical incident while using the movie database.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I felt better prepared to identify critical incidents after watching the video training.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I would like to have the *Report Incident* button built into the application that I use (instead of in a separate window).

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I would like the *Report Incident* button placed in a separate window (independent from the application that I use) floating on the computer's desktop.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I was motivated to report positive critical incidents.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I was motivated to report negative critical incidents.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

As a user, I would like to get feedback from developers about the critical incident reports that I send.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

How soon (realistically) would you expect feedback from developers?

❒ 1 hour ❒ 12 hours ❒ 24 hours ❒ 36 hours ❒ 48 hours

I would like developers to keep me informed of progress in solving the problem I reported.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I prefer to report critical incidents:

❒ immediately I encounter them ❒ after completing my task

If you report critical incidents **after completing your tasks**, explain why.

C.4 Post-test questionnaire for evaluator-subjects who only analyzed critical incident reports

Indicate the time you spent analyzing the critical incident reports and video clips into a list of usability problems.
____ hours _____ min.

I generally agreed with the severity ratings user-subjects gave to critical incident reports.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

It was easy to understand the content of the critical incident reports.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

It was easy to create a list of usability problems using critical incident reports (without any supporting videotape clips).

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I think that video clips (with audio) of screen action, in addition to the critical incident reports, would have helped me determine usability problem descriptions.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I would prefer to listen to verbal critical incident reports (i.e., audio with user's voice) rather than reading textual reports.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

I would prefer to read the online version of the critical incident reports instead of paper copy.

Strongly disagree	❒ 1	❒ 2	❒ 3	❒ 4	❒ 5	❒ 6	Strongly agree	❒ N/A

C.5 Post-test questionnaire for evaluator-subjects who analyzed both video clips and critical incident reports

Indicate the time you spent analyzing the critical incident reports and video clips into a list of usability problems.
____ hours _____ min.

I generally agreed with the severity ratings user-subjects gave to critical incident reports.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

It was easy to understand the content of the critical incident reports.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

It was easy to create a list of usability problems using the critical incident reports and the videotape clips.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

The videotape clips played an important role in understanding critical incidents.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

The audio portion of the videotape clips played an important role in understanding critical incidents.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

I *could have determined* the usability problems without using the videotape clips (only with the critical incident reports).

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

I *would prefer* to determine the usability problem descriptions without using the videotape clips (only with the critical incident reports).

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

I would prefer to listen to verbal critical incident reports (i.e., audio with user's voice) rather than reading textual reports.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

I would prefer to determine the usability problem descriptions only using the videotape clips (i.e., video and audio, no textual reports).

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

I would prefer to read the online version of the critical incident reports instead of paper copy.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

The length of the videotape clips (3 min.) showed enough information to help me determine usability problems.

Strongly disagree ❒ 1 ❒ 2 ❒ 3 ❒ 4 ❒ 5 ❒ 6 Strongly agree ❒ N/A

If 3 min. was not the best length for you, what is your desired length? _____ min.

APPENDIX D: SCRIPT OF TRAINING VIDEOTAPE

D.1 Introduction

"The *training lecture* conveys explanation content about critical incidents. You will be watching a videotape of a user who experiences usability problems while performing tasks with various software applications. Your job is to listen what he has to say about the situations that he encounters and why these are considered as critical incidents. Do you have any questions before we play the tape?"

D.2 Deleting a document from a database

Hi, I'm Rex Hartson in the Virginia Tech Usability Methods Lab. In this scene I play the role of a user working on a personal document retrieval system. I'm browsing through some documents here and I see some information on the screen for a document that I already discarded in the paper version, so now I want to delete the record from the database.

User remarks

[In Browse, go to document #1235.]

It looks pretty easy here.

I see a button that says Delete Document.

I click on that… [A dialog box comes up.]

Oh, wait a minute!

I got a dialog box that says "Are you sure you want to delete?".

OK, that makes sense to me as a user… to be cautious.

But what is this funny looking stick of dynamite here. I'm not sure what it means, except that probably is some kind of warning. Looks too dangerous; scary.

Explanation of critical incident

This is a minor critical incident. The icon is inappropriate for the situation. OK, lets continue on.

User remarks

[Still at same dialog box.]

But, the next sentence confuses me. It says "This marks the record for deletion."

Well, I didn't ask the system to mark it. I asked the system to delete it.

And there is another button here that also says "delete". So I guess I should click on that to see what happens.

[Dismiss dialog box.]

Well, nothing happened. The record is still there.

OK, yeah. Up here it says marked for delete.

Explanation of critical incident

This usability problem is of intermediate importance. There is no good feedback and the message is not visible enough. OK, now back to the user.

User remarks

Now, I'm really confused even more. I don't know whether I deleted it or not.

I still see the document here, but maybe if I go backwards by one, and then come back, maybe it will be gone.

No, it's still there. It still says marked for delete.

So, I don't really understand what it takes to actually delete it. It looks like it only marked it for delete. And I really don't know what that means.

[Pause and click the "Undelete" button].

Explanation of critical incident

To explain, this one is a major critical incident. The user is confused about the model of how the system works for deletion.

The interface doesn't help the user understand that model.

The way it really works is, when you click on this button to delete a document, it actually just marks it for deletion.

In the next dialog box, it alludes to this but doesn't really explain it. That leads to confusion. If fact, you can still undelete it by clicking on the undelete button, any time you want to up until the time that you use a function called "Permanently delete from the database". And that actually deletes all of the records that are marked for deletion and permanently removes them from the database.

At that point, they are unrecoverable.

There is a second somewhat minor usability issue related to these two buttons. Instead of having two buttons, we could combine the two into a single button and toggle back and forth between the two labels, for Delete and Undelete.

OK, let's assume that we have just told the user all about how deletion works and pick up that task from here to go ahead and get rid of the record. Again, I'll play the part of the user.

User remarks

Now that I know how this works, I'll select "Permanently Delete from Database".

And now it says that it will permanently delete records that are marked.

OK, so following what was explained to me, then this makes more sense.

Now, here is a sentence though that says: "After deletion is done, delete button will be disabled."

What does that mean I should do? I don't get it.

Explanation of critical incident

To explain this critical incident, that's useless information. Designers should get rid of it. Is not user centered. The result is further confusion on the part of the user.

Now back to the user.

User remarks

I'm not really sure where I am or what I want to do now.

I feel like if click on Delete now it could do some damage that I can't recover from.

I guess I need to go and find out more about how this works.

So, that means for the moment I do not want to delete. I want to change my mind and back out of this.

But, there is not way to do that.

There is no Cancel button, or Escape button; there is no way to get out of this.

So know I really don't know what to do. I'm stuck.

Explanation of critical incident

It needs another button. Probably not cancel, but just "Return to MADAM database", or something like that.

This is a case of a missing navigational feature. Designers should always allow a way to back out without committing to do the delete.

D.3 Counting the number of penalties called on your favorite football team

In this example I play the part of a person who is watching on TV one of the final football games of the season. I'm disappointed because my favorite team is getting called for many penalties during the game. I know they are capable of winning, but the penalties are holding them back. To take my mind of this dilemma, I decide to count the team's penalties for myself. So at half time of the game, I quickly programmed with HTML and JavaScript (Goodman, 1996) a tool to count the number of penalties.

User remarks

So here we have a penalty counter. You can see that if you click the plus ["+"] button, it puts one in the penalty count. Hitting the minus ["-"] button subtracts and plus adds. I can also reset the counter by clicking on the *Reset* button. So, now let's get on with the game.

So here we go, I'm sitting back enjoying the game and out of the sudden, "Oh no!, my team gets a penalty". So I run up here, click the "+" and we have a penalty of one. So, I'm relaxed again watching again, and after some chips and soda I realized, "Ugh!, another penalty, too bad, but at least I'm counting them now.

And this goes on as we go through the game, until one time I'm too excited and I clicked up here [*Reset* button], "Oh no!, what I have done!". I missed the "+" button over here and actually clicked the *Reset* button by accident. That means that the penalty counter now is reset to zero, and I don't remember what it was. So, I really made a user error on my little interface.

Explanation of critical incident

The usability problem here is that the "+" button, which is the most frequently used button, ought to be featured more prominently and not, in fact, stuck graphically closely in here between these other two buttons [(i.e., *Reset* and "-"buttons)]. So, if the "+" button is the more frequently used, we want to make it bigger for one thing, and we want to get it away from this close proximity to these other two buttons.

So, I make a second design which looks like this. In this new design, I now have the "+" button way out here on the left side, not buried between two buttons as they were in the first example. Now I have it in left side here and much larger size. It is also far away enough from the *Reset* button so that error that I made is extremely unlikely.

D.4 Formatting and labeling a diskette in Microsoft DOS format

In this scene I am a user who is working on a presentation on my Macintosh computer at home. I need to take that presentation to work tomorrow, but the computer at my office uses only DOS and can't read Macintosh diskettes. So I want to reformat one of my diskettes to DOS format, save the document on it, and take that diskette to my office tomorrow.

User remarks

So, I put the diskette in, and a dialog box comes up here.

So I choose the format here to be the DOS format.

And now I want to assign this a name since this is for my presentation, so I am going to type in the title "Presentation 1997".

[Start typing. After typing "Presentatio", a beep sounds.]

Ugh! It beeped there in "Presentatio".

I'm not sure why it didn't…

[Type another letter and a beep sounds.]

Oh. It beeped again.

What I want to do is type this in…

[Type another letter and a beep sounds.]

[Dialog box appears on screen.]

Ha! At last, a message here: "Cannot add more characters to the name, because DOS disk names are limited to 11 characters in length".

Yes, that's right. I did forget DOS uses shorter names.

OK, well. I'll get rid of that.

And go back here and call it "Presn97" to shorten it up.

Explanation of critical incident

To explain, the usability problem was that the length of this box indicates that I can put a lot more than 11 characters in here, even though is selected as a DOS format down here.

[Highlight extra space of text box].

This extra space is a misleading visual cue. The long highlighted text field makes me think I can type more in, but I really can't.

The solution for this, which would have facilitated my task and help eliminate errors, would have been to change the size of this text box to just 11 characters when I selected the DOS format here. Then I could have seen that I am running out of space for the name when I ran into the end of the box, and I would have known immediately why it was beeping.

D.5 Changing the Auto-save parameter of Microsoft Word to 1 ½ or 1.5 minutes

In this scene I am a user who is writing a paper using Microsoft Word on a Macintosh computer. There have been electrical storms going on, and the power sort of has been flickering. But I really have a deadline and the work is going very slowly because it's hard to think what I want to write when I'm worried that maybe the power might fail and I'll loose my document. So what I want to do is to use "Autosave" and I want to set it so that is very short time between saves. So I'm going to pick something arbitrary like 1 ½ minutes or something like that.

User remarks

[A Word document is already open.]

So, I'm in Word right now.

[User is looking at different menu options.]

I want to do "Autosave" here, so I'm looking at all these different things here.

[User is looking at "File" menu option.]

Looks like it has to do with this menu up here because it really is about files.

So let me look and see if there is anything about "Autosave" .

No… there is "Save", "Save as…", all that kind of stuff, but there is nothing about "Autosave".

Is not about editing, I don't think. No…

Not "View", "Insert", "Format".

Well, I don't really think of it as a "Tool", but ….

I guess …..

Yeah, that's the one right there because it is really about a preference; setting a preference.

I bet that's it!

I never would have guessed that it was under "Tools", but….

Oh yeah.. "Open and Save". It's sure about that.

Yeah, there we are.

Explanation of critical incident

To explain, there is a somewhat minor usability problem in providing easy access or affordance in locating the "Autosave" command. With the large number of Word commands, designers can't put them all on the menus. But maybe the names could be labeled better, or bring out "Preferences" on its own menu. Now lets continue with the user.

User remarks

OK, I want to click on "Autosave"; save every reminder.

It looks like 15 minutes is the default, but that's way too long for this electrical storm.

So I'm going to put 1.5 minutes; about 90 seconds.

OK. Let's dismiss that box and we can go back to typing.

[A beep sounds and an error message box appears on screen.]

Oh! OK. Wait a minute here.

When I was dismissing the box, I get an error message that says: "Not a valid number", but at that point I wasn't dealing with a number. I was just trying to get rid of the dialog box.

The only thing I can think of is that this must refer back to the thing that I did the step before.

So the timing is not very good for that message.

Also… hang on here, the message is telling me what's wrong, but doesn't tell me what I should do.

[Click OK button].

So I click on OK and get another message that says: "Number must be between 1 and 120".

Well, 1.5 is between 1 and 120!

So, I still don't understand exactly what's wrong.

Anyway, I guess it must be because of the decimal point.

So, I better put something like 2 there.

[Dismiss box.]

Ha! It takes that. OK.

[Pause and bring "Preferences" dialog box again.]

Explanation of critical incident

All right. In this scenario, we ran into several usability problems.

First, there was the difficulty of finding the "Autosave" settings.

Now, we got here and if you remember I put 1.5 here, and nothing happened, but when I tried to leave the message box, then I didn't get any immediate feedback, but a delayed feedback for the previous action.

So the timing there was not so good, because really you want an error message closely associated in time with the action that caused the error.

[Try to close dialog box. First error message box appears on screen].

And this first error message is simply not very informative.

It says: "Not a valid number".

But. A good error message should be constructive and say what is a valid number, and what you could do to get one.

[Click "OK" button and second error message box appears on screen.]

So, I get rid of that anyway, and now it tries to give me a little bit of information: "Number must be between 1 and 120".

And, in fact, 1.5 is in this range, but the problem is not that 1.5 is not on that range.

But, of course, as we found the real problem was it has to be an integer or a whole number between 1 and 120.

It doesn't say that.

The problem is that it is in the wrong type.

VITA

José Carlos Castillo, also called Charlie by his family and friends, was born to José and Dalila Castillo on September 24, 1972 in San Juan, Puerto Rico. In his senior year of high school, he held the position of battalion commander and the highest rank among all cadets from Antilles Military Academy, where he graduated in May 1989. In August of the same year, he entered the Computer Engineering program at University of Puerto Rico in Mayagüez, Puerto Rico. There he became interested in the area of human-computer interaction and worked as research assistant in several projects related to this field. He also worked during four summers as a Software Engineer for Motorola Inc. in Schaumburg, Illinois. He graduated in May 1994 with a Bachelor of Science degree in Computer Engineering.

Castillo continued in pursuit of a Master of Science degree in Computer Science at Virginia Polytechnic Institute and State University, with a concentration in human-computer interaction. Upon completion of this degree in July 1997, he will start working full-time as a Human Factors Engineer for U S WEST Information Technologies in Denver, Colorado. He also plans to continue research in human-computer interaction, Web technologies, and remote usability evaluation.

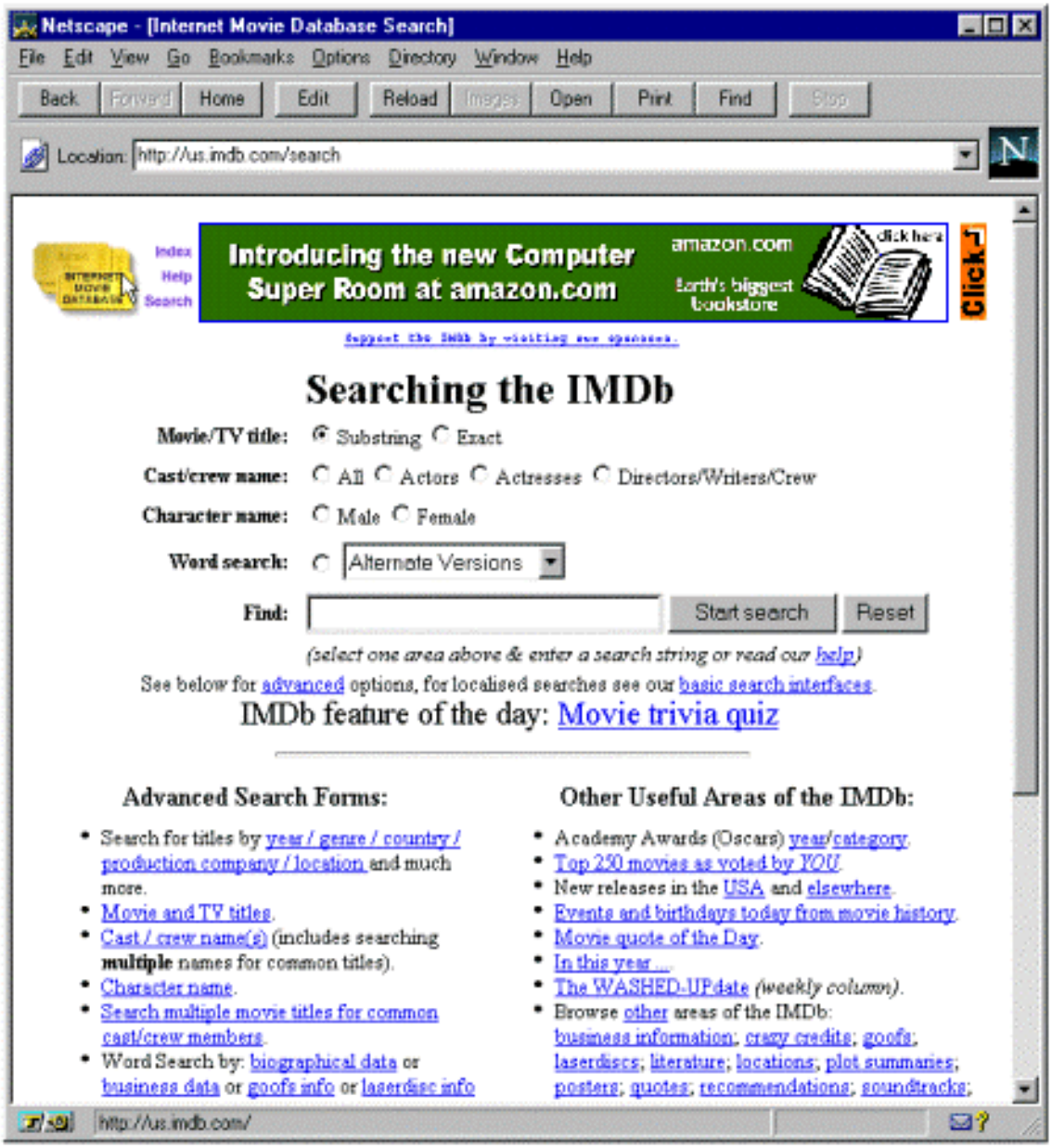


Figure 6-2. User interface of main search page of the Internet Movie Database

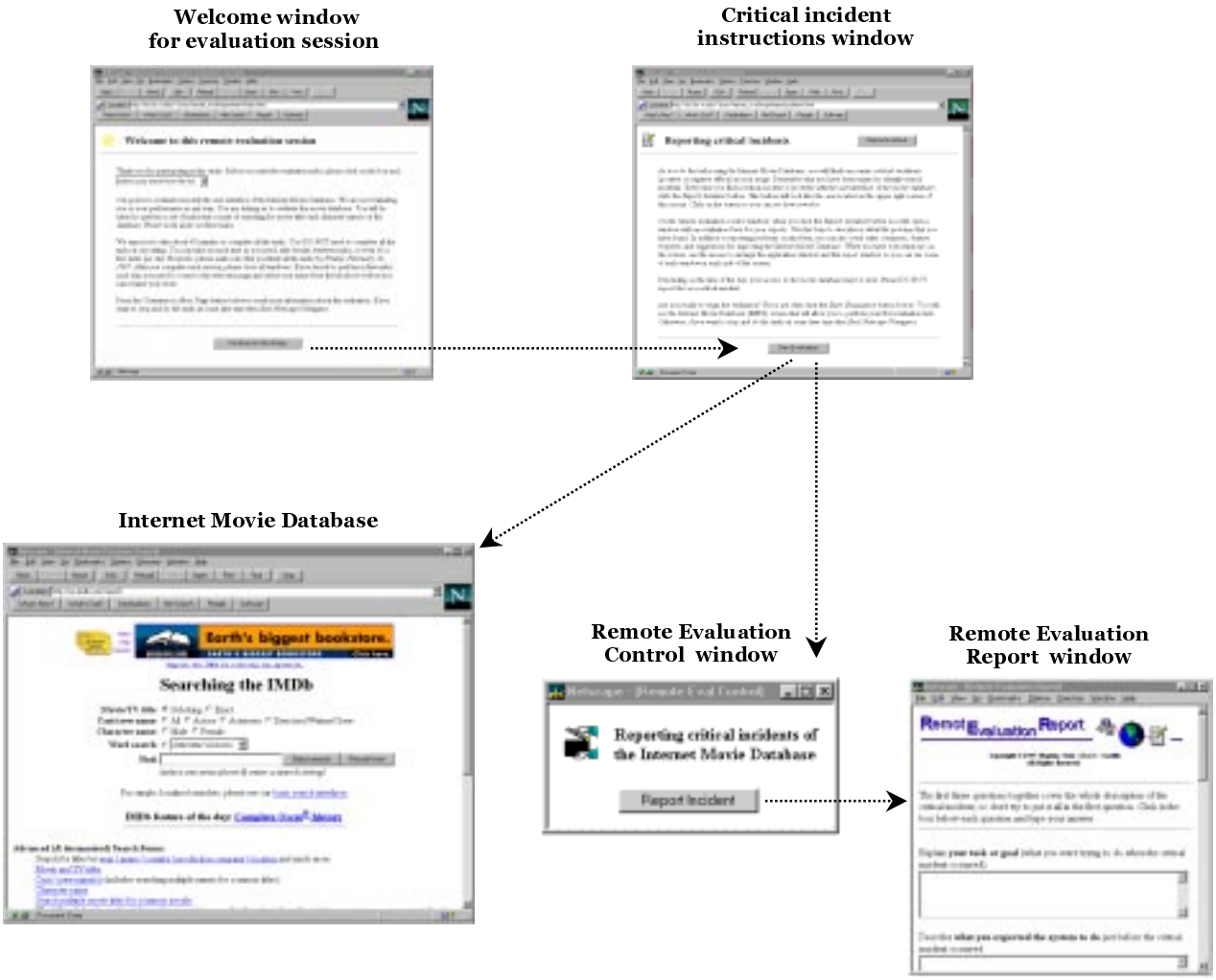


Figure 6-3. Sequencing relationships among the screens for Phase I

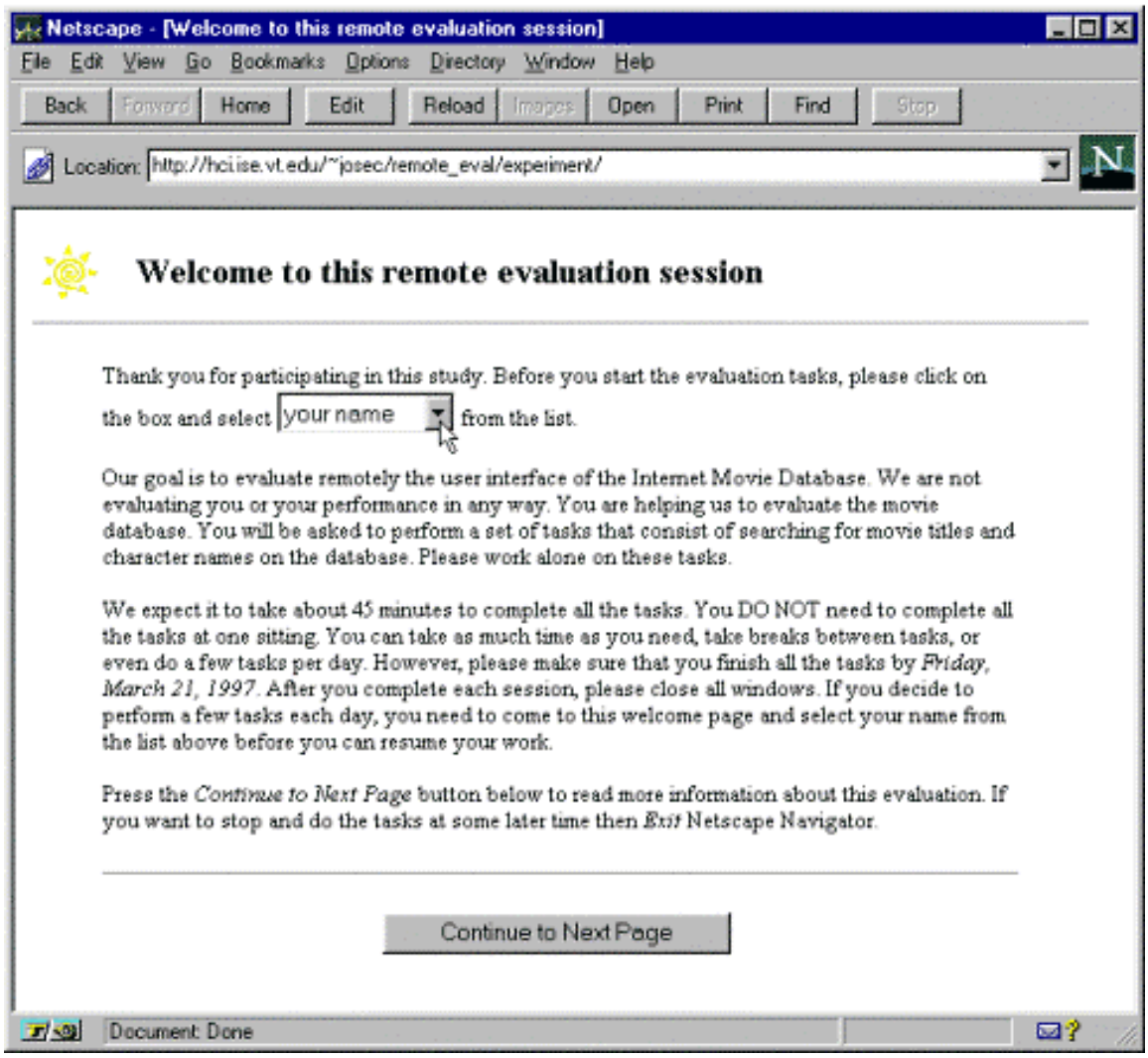


Figure 6-4. Welcome window for remote evaluation study

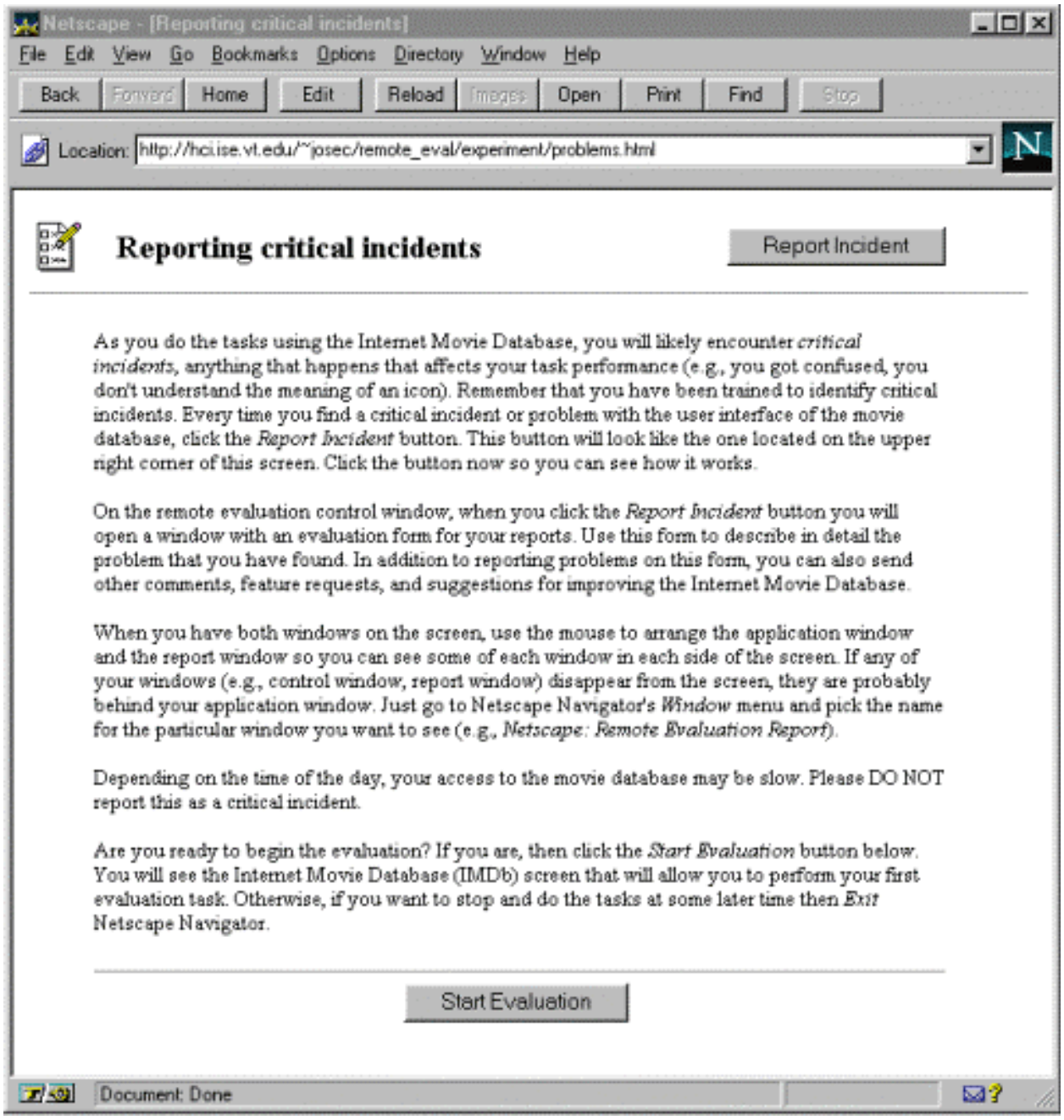


Figure 6-6. Critical Incident Instructions window

Figure 6-7. Remote Evaluation Control window

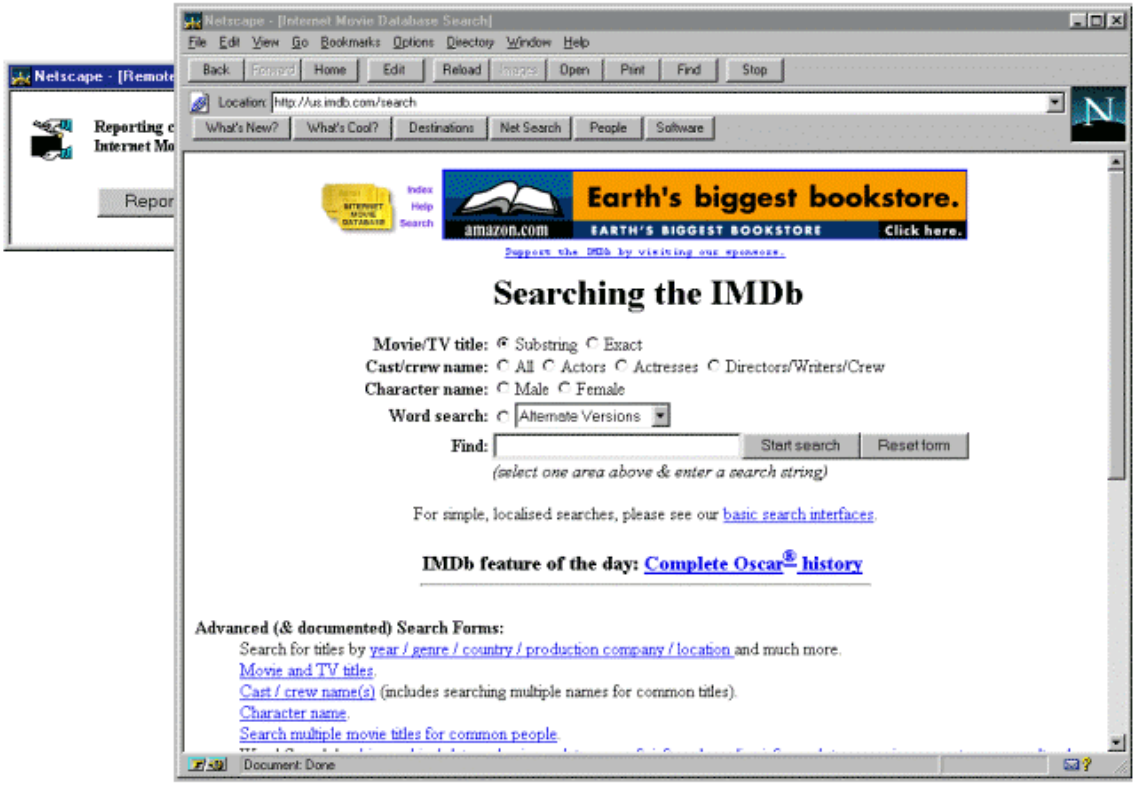


Figure 6-8. Positioning of the Remote Evaluation Control window and the application window

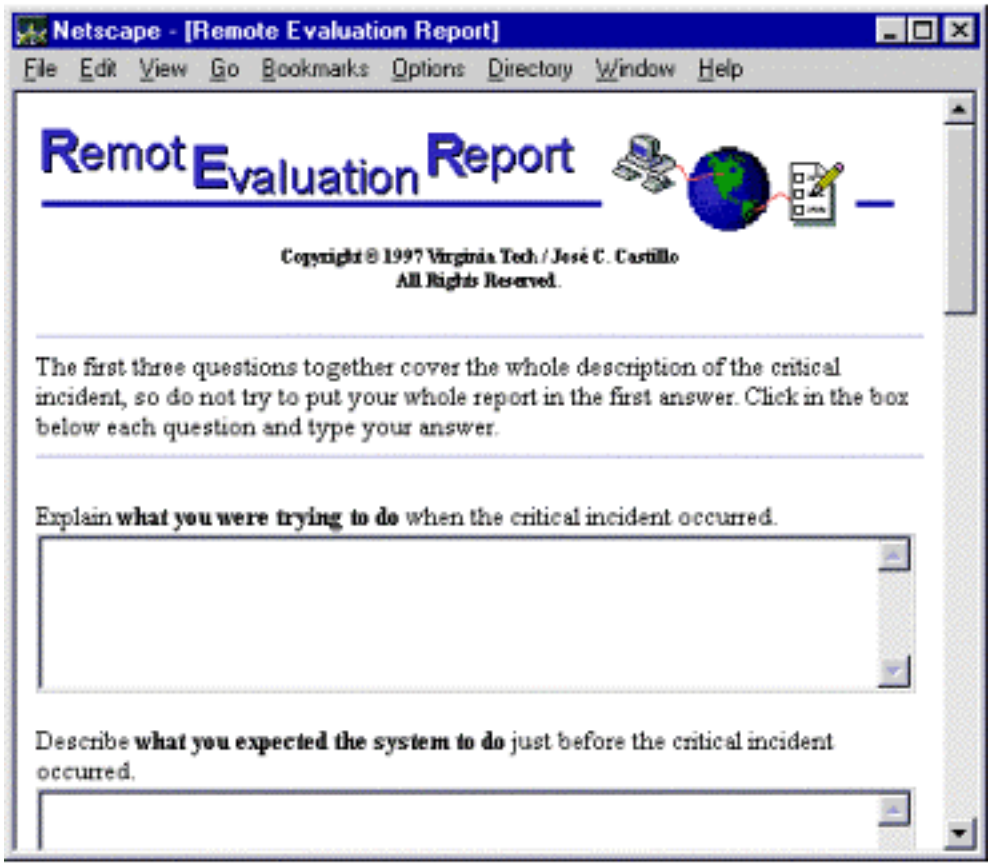


Figure 6-10. Remote Evaluation Report window

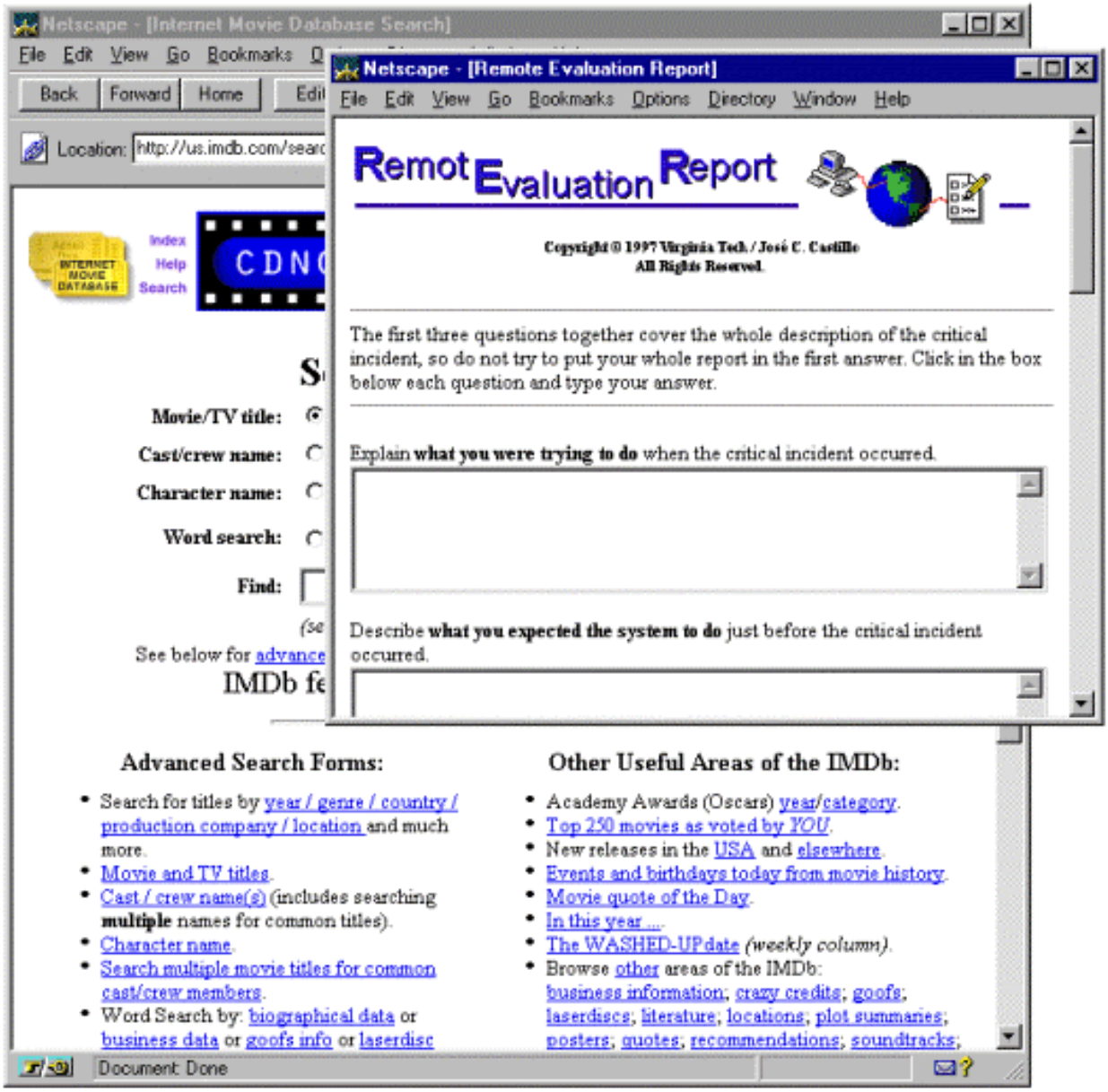


Figure 6-11. Positioning of the Remote Evaluation Report window and the application window

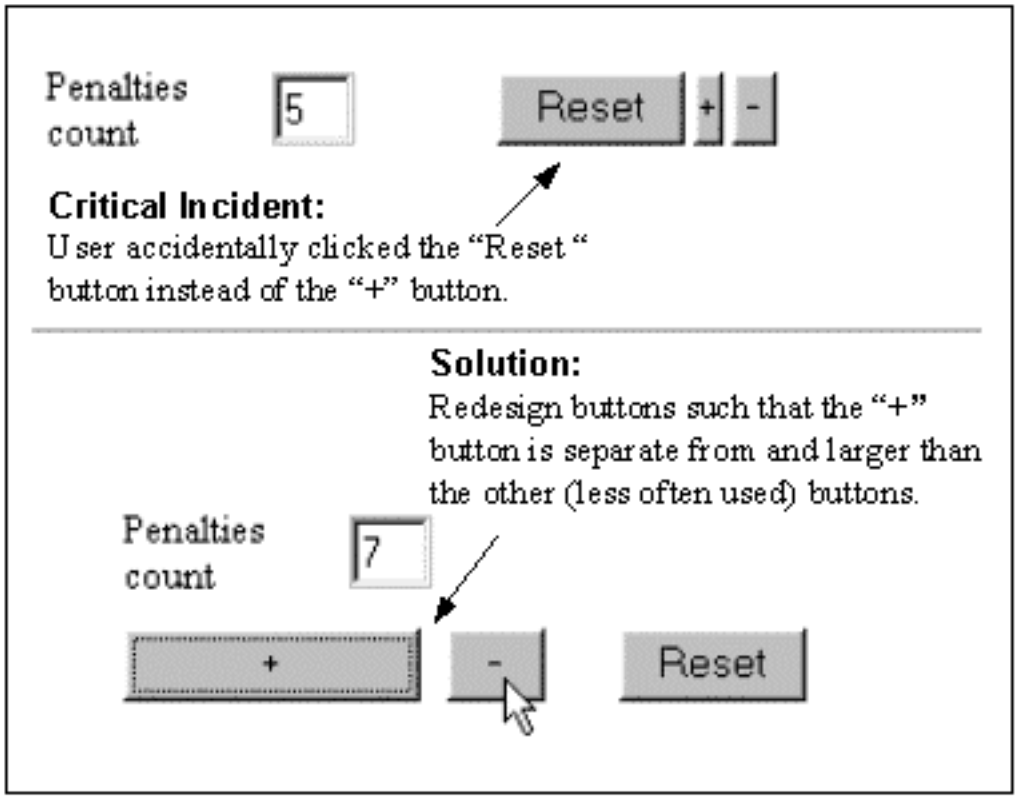


Figure 6-12. Critical incident found on a Web-based counter

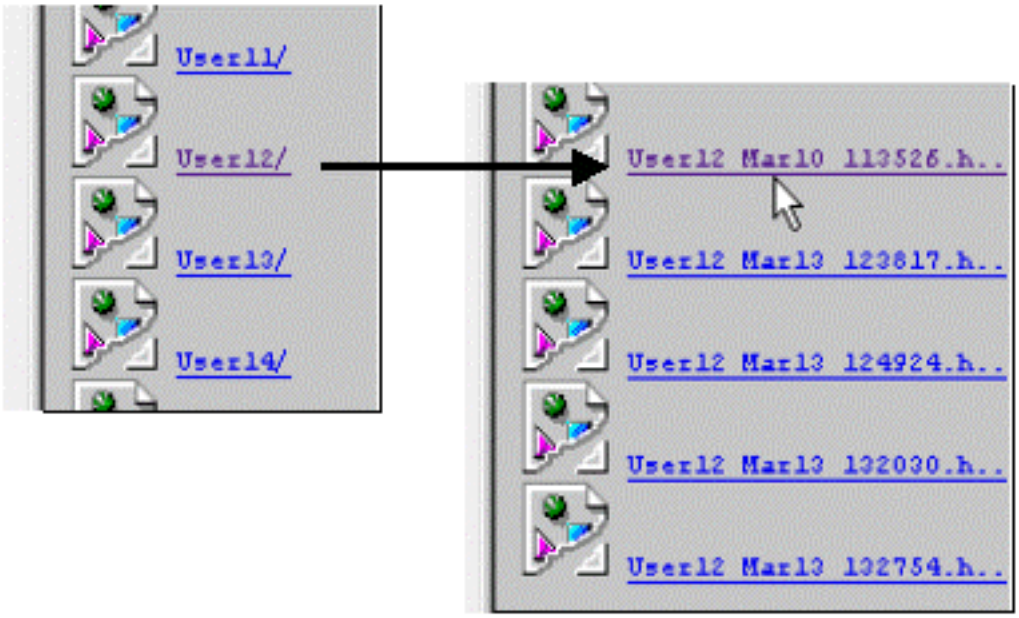


Figure 6-13. Providing anonymity for user-subject critical incident reports

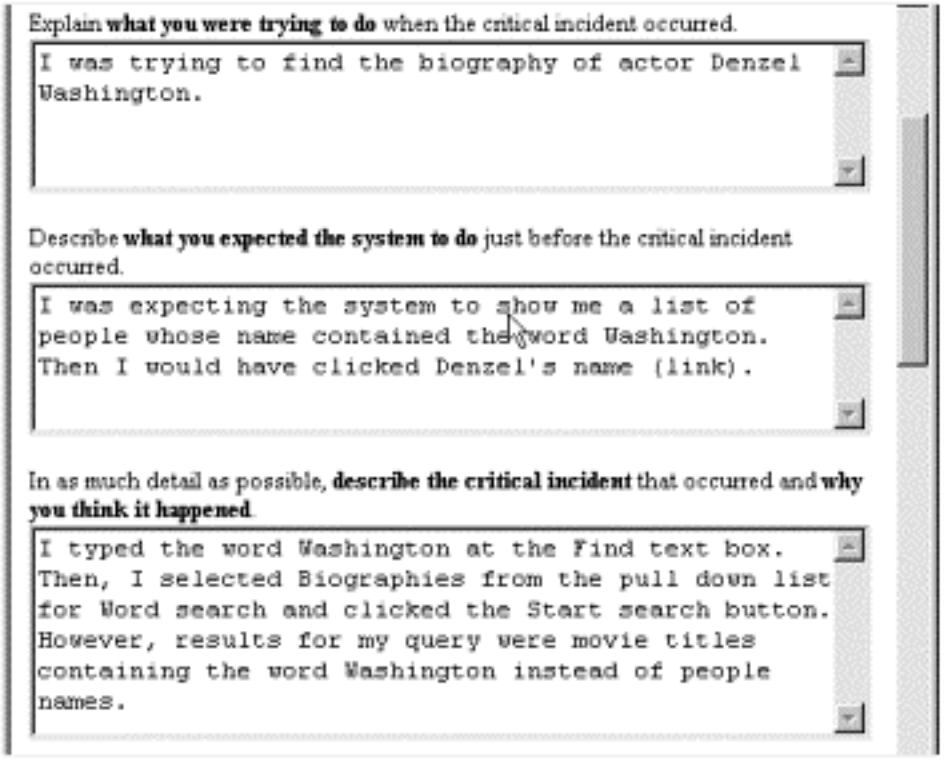


Figure 6-14. Indication of user task and description of the critical incident

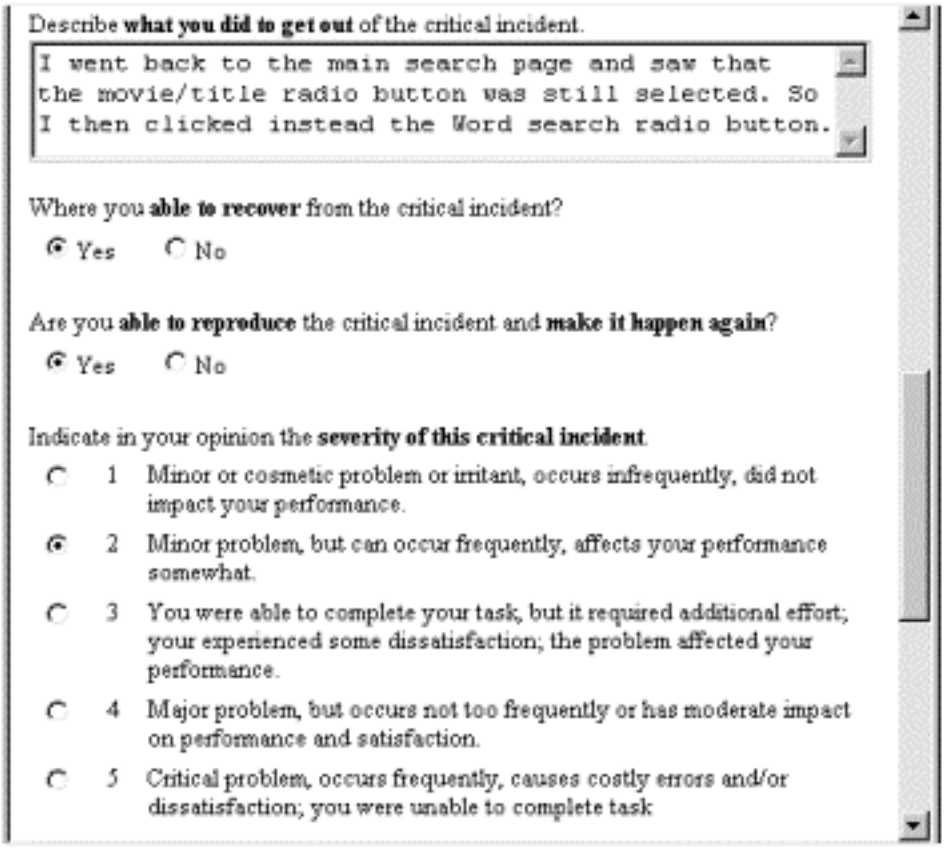


Figure 6-15. Indication of how user got out of the situation, ability to recover and reproduce the critical incident, and severity of the critical incident.

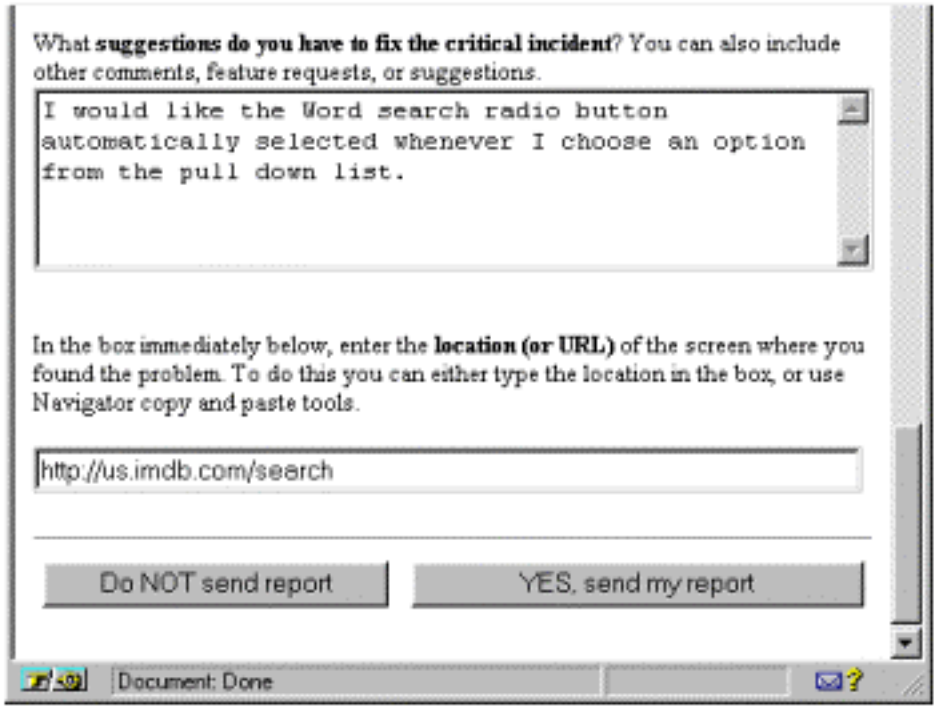


Figure 6-16. Suggestions for fixing the problem and location of the page with critical incident

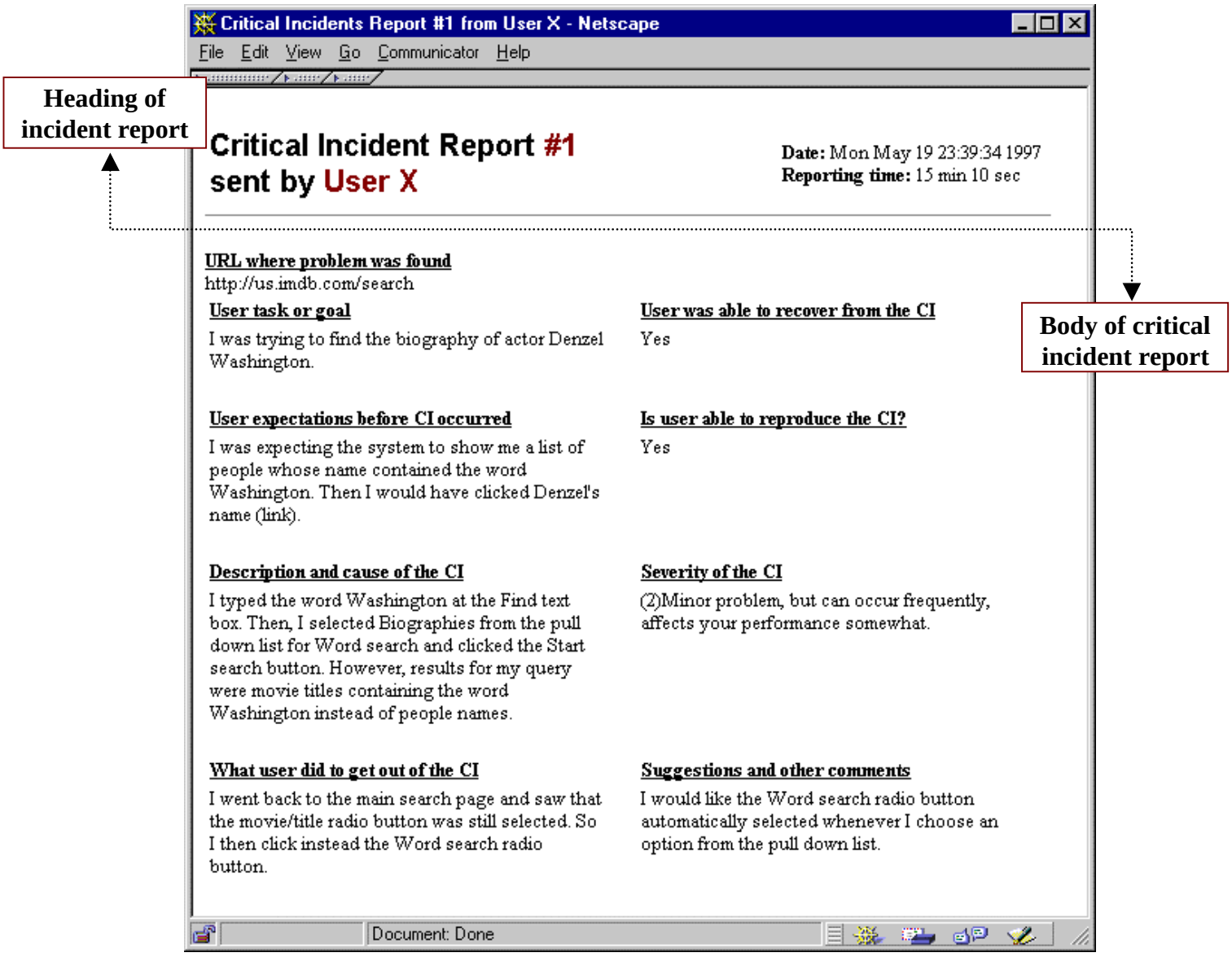


Figure 6-17. Critical incident report from User #X

When users identify that a situation is a critical incident, they click this button to activate a screen capture tool that records a screen-sequence clip showing all screen actions that occurred before the button was clicked.

Users click this button to send quick questions, comments, and/or solutions about the critical incident.

Users click this button to send a complete and detailed report of the critical incident.

Users can select a previously sent critical incident report from this list to browse it in read-only mode.

Figure 7-7. New critical incident reporting tool