Evaluating the Collaborative Critique Method

Tamara Babaian
CIS Department
Bentley University
Waltham, MA, 02452, USA
tbabaian@bentley.edu

Wendy Lucas
CIS Department
Bentley University
Waltham, MA, 02452, USA
wlucas@bentley.edu

Mari-Klara Oja
IPM Department
Bentley University
Waltham, MA, 02452, USA
oja_mari@bentley.edu

ABSTRACT

We introduce a new usability walkthrough method called Collaborative Critique (CC), which is inspired by the human-computer collaboration paradigm of system-user interaction. This method applies a "collaboration lens" to assessing the system's behavior and its impact on the user's efforts in the context of the task being performed. We present findings from a laboratory evaluation of the CC method with usability practitioners, in which the results of the CC walkthrough were compared to a benchmark set of problems collected via user testing with two experimental Enterprise Resource Planning (ERP) system tasks.The development of this new usability evaluation method was driven by the need for an approach that assesses the adequacy of the system's support for reducing the user's cognitive and physical effort in the context of the interaction.

Author Keywords

Usability inspection methods, human-computer collaboration, complex systems, ERP

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces-Evaluation/methodology

INTRODUCTION

Despite the plethora of available usability methods, designing and evaluating information systems for usability remains a challenge due to the inherent complexity of designing for a broad range of users performing a variety of tasks [2]. In this paper, we present the Collaborative Critique (CC) - a new usability walkthrough method that is based on the step-by-step evaluation of how well the system supports a user working on a task. Findings from our initial experimental evaluation in the domain of enterprise information systems are encouraging, with walkthrough teams predicting a majority of the usability issues identified by the user study. Coverage of issues associated with error situations, which are explicitly addressed by the CC method, are particularly strong.

Usability inspection methods, such as Heuristic Evaluation (HE), Heuristic Walkthrough (HW) and Cognitive Walkthrough (CW), remain a "key technique" [10], although researchers and practitioners have acknowledged their deficiencies. These deficiencies include limiting the usability analysis according to a set of heuristics or specific user difficulties; the limited consideration of the context of the system-user interaction; and little analysis of problem causes, leading to difficulty in generating appropriate fixes [9, 24]. The CC method aims to mitigate some of these problems by explicitly invoking the context of the interaction in the evaluation, including the task context and the user's experience and role. It considers the total effect of the interaction on the user's cognitive and physical efforts and evaluates specific components of the system's behavior.

Like a Cognitive Walkthrough, a Collaborative Critique involves evaluators going through a step-by-step scenario of using the system to perform a task while answering a set of questions. In formulating the CC questions, we have employed the human-computer collaboration paradigm [21] because it provides a useful framework for assessing the effectiveness and efficiency of system-user interactions while potentially addressing several limitations of the popular usability inspection methods mentioned above. Human-computer collaboration does not require the system to emulate human behavior, since computers and people have very different and sometimes opposite strengths. Instead, in accordance with theory of collaborative behavior (e.g. [12, 5]), it emphasizes the need for the system to behave in a goal- and context-aware way, provide effective means of communication and information sharing to facilitate efficiency of joint user-system operation, and help the user resolve problems that arise during the interaction instead of merely reporting the errors [11].

The CC questions were derived using theory of collaboration [12, 5] and systematically assess the system's behavior on a task as well as the cognitive and physical effort expended by the user. We note that in system behavior we include not only the dynamics of its interaction with the user, but also the way the system presents itself, i.e. the static components of the user interface.

To test the CC method, we conducted a laboratory study aimed at comparing the results of the collaborative critique by usability professionals with the results of user testing in the domain of Enterprise Resource Planning (ERP). Part one of the experiment consisted of user testing of two task scenarios with subsequent analysis of the collected usability data by the authors of this paper. Part two involved usability profes-

CHI'12, May 5–10, 2012, Austin, Texas, USA.

sionals evaluating the user-tested task scenarios with the CC walkthrough. Findings from comparing the CC-based predictions to the actual usability problems identified in the user study are presented here.

ERP systems, which integrate information and manage processes from across the organization, were chosen for our testing because of our knowledge of the realities of use of these complex systems [22, 4], as well as the relevance and significant impact of their usability on a large constituency of users. Our familiarity with this domain from our field work enabled the selection of realistic task scenarios and the preparation of user training materials similar to those found in the field.

The paper is organized as follows: after presenting a review of related literature, we introduce the CC method and questions. This is followed by a description of the experimental methodology and analysis of the study data. We conclude with a discussion of the results.

RELATED WORK

A thorough summary on the state of usability inspection methods and their evaluation methodologies is presented in [10]. As noted there, while the popularity of inspection methods has decreased in the past decade, they remain an essential and useful tool for cost-efficient assessments of interfaces at the early stages of development that can be applied to driving design iterations and used as an educational tool for user experience professionals and designers.

Collaborative Walkthrough(CW) [23, 14] is the method most closely related to the Collaborative Critique. Like CW, whose design was based on theory of learning to evaluate learnability of interfaces, CC is theoretically motivated but focuses on evaluating the collaborative strength of the system. CC and CW have very similar procedures of step-by-step evaluation of the task interface, but in our method the evaluators are also encouraged to explore the interface if necessary for answering the CC questions. The scope of the aspects of the system-user interaction evaluated by the CC questions is broader than CW's and includes the adequacy of the system's support of the user's cognitive and physical effort in the context of the overall task. CC pays special attention to evaluating system-user support in error situations because, as we found in our field studies [4], they are a major source of user angst and a time sink for diagnosing and fixing problems yet are ordinarily overlooked in usability evaluations. The theoretically derived systematic focus on the system's static and dynamic behavior is what also distinguishes our method from the empirically motivated HE [16] and the hybrid HW [20] methods.

The methodology of evaluation and comparison of the walkthrough methods has matured in the past years [20, 13]; however, challenges remain due to the absence of a generally accepted framework for usability problem categorization. As a result, usability problem counts, on whose basis the assessments and comparisons are made, are subjective and depend on the abstraction level used in a particular study [10, 13]. This makes it difficult to make a meaningful comparison of the results of the CC evaluation presented in this paper to the results of other studies. For example, Hartson et al. [13] cite a number of studies with a wide range of thoroughness and validity measures for the same method.

The process of moving from the candidate problem discovery stage of the inspection methods to problem confirmation, which includes analysis of causes and the seriousness of the problem's impact on user performance, remains one of the key challenges for usability inspection methods. Andre et al. [3] have argued that these difficulties can be partially alleviated by using a systematic framework for guiding the assessment of the candidate problem data. They have proposed a User Action Framework (UAF), which is closely related to Norman's theory of action's model of interaction [17]. UAF organizes the interaction events into three major components: planning goals and intentions for physical action, physical action, and assessment of the outcome and system response. These components are further elaborated into several levels of more detailed categories. The authors report a high degree of reliability (the degree of agreement by trained usability analysts in classifying usability problems according to UAF) from their laboratory study of 10 usability analysts categorizing 15 usability problems. The CC questions (fig. 1) map to the categories of UAF as well as to Norman's action theory in a straightforward way, which suggests that reliability properties of UAF may apply to the Collaborative Critique.

On a different front, several researchers have noted the need for developing usability evaluation methods and practices for *complex* information systems [15, 19, 8]. The work presented in this paper is a step in that direction. While the CC method is not directly intended to address all of the challenges of complex problem solving, as defined by Mirel [15], it places an emphasis on evaluating system *usefulness* within the task context by focusing on the system's strength-as-a-partner in an interaction [11, 4] and the resulting efforts put forth by the user. Development of inspection-based usability methods for complex domains is especially important due to the fact that user testing in such domains requires recruiting expert users who may not be readily available in sufficient numbers.

COLLABORATIVE CRITIQUE METHOD

Collaborative Critique is a walkthrough method that aims to assess system usability from the standpoint of the Human-Computer Collaboration (HCC) paradigm of system-user interaction. HCC views the interaction of a user with a system as a process in which they work together on a common goal [21, 11, 12]. We followed the "theory as insight" approach [11] and designed the Collaborative Critique to capture and assess the aspects of system behavior that are essential to an effective collaboration between the system and its user.

CC questions

Virtually all of the theoretical accounts of collaboration include the following essential attributes [21]: the parties share a *goal* of their collaboration and have *plans* for accomplishing it. Goals and plans may be incompletely specified in the beginning and refined in the course of action. Furthermore, partners have a *shared context* of interaction, and *communicate* with each other to maintain the shared context and refine

Collaborative Critique Questions

1. Will the user find the options for what he wants to do in the current screen?
2. For the user to figure out what to do now:
   (a) How much exploration is involved?
   Considering users with a range of experiences, answer with a number 1..5, where:
   1: most users will know what to do right away
   2: some users will have to explore to figure out what to do
   3: most users will have to explore to figure out what to do
   4: most users will have to explore and some will be unable to figure out what to do
   5: most users will have to explore and most will be unable to figure out what to do

   (b) How much confusion is involved? Considering users with a range of experiences, answer with a number 1..5, where:
   1: most won't be confused
   2: some will be somewhat confused
   3: most will be somewhat confused
   4: some will be very confused
   5: most will be very confused
3. Is the system using knowledge of the task in general, the current user, and the context of the current action to the fullest extent in order to:
   (a) appropriately guide the user?
   (b) reduce the effort involved in user input?
4. After execution of the current action, will the user understand
   (a) what progress has been made so far toward completing the overall task?
   (b) what remains to be done in order to complete the overall tasks?

The following three questions must be answered only in case an **error condition** *is reported.*

5. Does the system display information that clearly explains the problem to the user?
6. Does the system present steps the user can take for possible corrective actions?
7. Does the system present an easy way to take corrective actions?

Figure 1. CC Questions

their goals and plans. Such communication need not necessarily be verbal - systems and users communicate using common interface features, visual cues, and input devices [11]. In addition to the italicized attributes in the previous paragraph, Bratman's account of collaboration [5] and SharedPlans model of Grosz and Kraus [12] emphasize the parties' commitment to the success of the joint action, which requires their commitment to the success of their fellow collaborators. *Learning*, *adapting* and other kinds of *helpful behavior* that maximize the chances of success, are implied by this commitment. In the context of system-user interaction, the system provides the functions that need to be invoked to achieve the goal. Instead of waiting for the user to seek and enact them, a helpful collaborator-system can often successfully anticipate the need based on the current task context and prior history of interaction, presenting the user with appropriate choices. Last, but not least, collaboration also requires participants to *help a partner who is having a problem* accomplishing his or her part of the work [5].

The CC questions presented in figure 1 are intended to be answered for every step of an action sequence, like in a Cognitive Walkthrough. They ask the evaluator to assess if the system performs according to the tenets of collaboration and how much cognitive and physical effort is required of the user to be effective in working with the system on the task.

Question 1 assesses if the user will be able to relate their next goal (action) to the system-presented context and will be able to communicate that goal (action) to the system. Although the question is about the user, it indirectly assesses if the system's presentation of the context of the interaction and its means of communicating with the user are adequate. Questions 2(a) and 2(b) are intended to provide two summary numeric measures reflecting the cognitive (confusion) and physical (exploration) efforts involved in the user determining how to proceed in the current situation. The provided numeric scale of responses reflects the severity of the problem relative to the population of users. A summary description of the typical user's knowledge and experience with the task and the system interface is given to evaluators with every critique and is intended to guide their answer to this and other questions. The confusion and exploration metrics should help guide post-walkthrough decisions on whether the problem is serious enough to require a fix.

Questions 3(a) and 3(b) evaluate if the system is sufficiently helpful in guiding the user's actions and reducing the user's cognitive and physical efforts in performing the current step. Question 4(a) assesses the effectiveness of the system's communication on the progress accomplished thus far, and 4(b) focuses on the system's ability to keep the user informed about the possible plans for further action, which should help the user in determining what to do next.

Questions 5-7 focus on the system's helpfulness in error situations: whether the system provides a meaningful explanation of the error, presents possible fixes, and gives the user easy access to corrective actions.

CC Procedure

A Collaborative Critique of a user interface is performed by one or more evaluators familiar with the method for a given task specification based on a given sequence of actions-steps involved in achieving the task goals. The background information provided to each evaluator includes the general description of the evaluated task and related system functionality, as well as a profile of a typical user. The evaluators are given a spreadsheet (henceforth, CC Template), in which all the steps of the action sequence are listed along with space for recording the answers to the CC questions (fig. 1) at each step.

The evaluators are instructed to perform the following for each step of the action sequence:

1. See if they can discern the next action to be performed based on the task description and their own exploration of the interface.
2. Check the next action-step listed in the action sequence to see if they are proceeding correctly.
3. Perform the action as specified in the sequence.
4. Record their answers to CC questions 1-4 in the spreadsheet. For all questions except 2(a) and 2(b), the possible answers include "yes," "no," or "NA" (Not Applicable); A "no" answer must be explained. Explanations to other answers are not required.
5. If an error is reported by the system, record the error and answer questions 5-7.

In the next section, we describe the experimental methodology we used to test the CC walkthrough.

EXPERIMENTAL METHODOLOGY

An experimental evaluation of the Collaborative Critique was performed for assessing the effectiveness of this method at predicting usability issues with an ERP system from a leading manufacturer. There were two components to this evaluation: (1) a laboratory-based empirical user study, and (2) CC walkthrough evaluations. The purpose of the user study was to identify usability issues experienced by users performing two tasks with the ERP system. Teams of usability professionals conducted CC walkthroughs of those same tasks with the same ERP system. Issues identified by the user study provided a benchmark for analyzing responses to each of the walkthrough questions. A pilot study with three participants [18] informed the design of the final user experiment, while pilot walkthroughs with two participants informed the design of the final CC method.

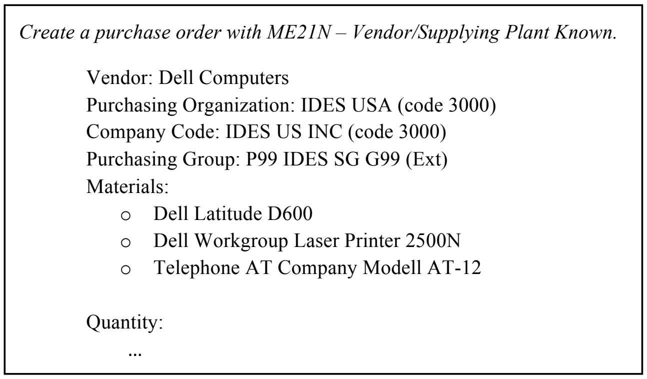
In this section, we first describe the two tasks that were chosen from three piloted tasks. We then describe the participants, the experimental setup, and the protocol for each study.

Experimental Tasks

The two tasks chosen for evaluation in the user study and the Collaborative Critique were an Authorizations task (Task 1) and a Purchase Order task (Task 2). Task 1 is a typical administrative task whose purpose is to create an authorization profile for a new user. This task has 52 steps and is comprised of these subtasks: create a new role, associate the new role with an authorizations profile, create a new user, assign the role to the user, and log on as the new user. Task 2 is a commonly performed transactional task whose goal is to create and submit a Purchase Order (PO). This task has 66 steps and involves filling in the required information to multiple sections of a PO and then submitting the completed product. Task 2 was also designed to include three likely error situations.

The following materials were created for each task for use in the experimental evaluation:

- Task description document outlining the task to be performed and containing the data needed to log in and perform that task. Figure 2 shows a partial snapshot of this document for the PO task.



Create a purchase order with ME21N – Vendor/Supplying Plant Known.

Vendor: Dell Computers
Purchasing Organization: IDES USA (code 3000)
Company Code: IDES US INC (code 3000)
Purchasing Group: P99 IDES SG G99 (Ext)
Materials:

- Dell Latitude D600
- Dell Workgroup Laser Printer 2500N
- Telephone AT Company Modell AT-12

Quantity:
...

Figure 2. Partial snapshot of task description document for Task 2

- Task training document containing an overview of the task followed by step-by-step instructions and screenshots on how to perform the task using different data from that for the experiment.

- Task video tutorial showing the ERP screen as an instructor describes and performs the task using the same data as in the task training document.

- Task action sequence with step-by-step instructions on how to perform the task using the data for the experiment.

The following sections on the user study and CC walkthrough methodologies include descriptions of how these materials were used.

User Study Methodology

We adapted the user-reported critical incident technique [7] to a laboratory-based user study via an approach similar to [1], in which users reported on negative incidents, or usability issues, while performing a task with a system. This was followed by retrospective reporting [6] for providing the researcher with additional clarity on the context and nature of the problems encountered from the participant's perspective. Each participant first viewed a video tutorial on the task, followed by a video tutorial on reporting usability issues. Next, he performed the task and reported any usability issues as they were encountered, with all system interactions and verbalizations recorded by screen- and audio-capture software. After completing the task, the participant and a researcher together reviewed the video of the user session and discussed the usability issues the participant had experienced.

Participants

For this study we sought participants who had varying levels of experience with ERP systems and some degree of business-related work experience, to reflect the typical user populations found in the workplace. Twenty participants, all with at least some exposure to using an ERP system, were recruited and randomly assigned to each of the two tasks. Seventeen had full-time work experience, while three had part-time experience. Table 1 shows their levels of ERP expertise, where the scale item of "novice" indicates little to no experience, "intermediate" indicates a moderate level of experience, and "expert" indicates considerable experience. All were college-educated, with nineteen holding either graduate or undergraduate business degrees or enrolled in business degree programs. The participants were compensated for their time following their study sessions.

		ERP Experience		
Task	Age (average)	Novice	Inter.	Expert
1	21 - 34 (26)	6	3	1
2	22 - 29 (26)	6	3	1

Table 1. User study participants.

Setup

Each participant was provided with two laptops. One of these was used solely for viewing the video tutorials and was removed when the training had concluded. The other had the following software installed on it: the ERP client for use in training and task performance, a word processing application for note-taking, screen- and audio-capture software, and a Java application for reporting usability problems during task performance.

Protocol

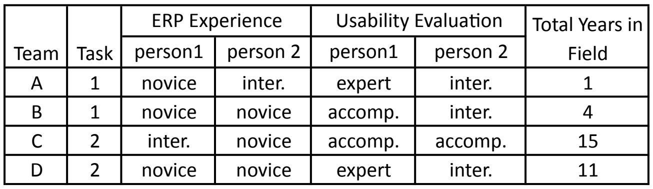
Each study session was conducted with an individual participant in a laboratory setting. The four components of the study were designed to fit into a two-hour session, though no time limitations were imposed. The components were performed sequentially as follows:

1. *Viewing of the task video tutorial* (30 minutes). The participant was instructed to view the video tutorial for the task he would be performing and encouraged to follow along with it using the ERP client running on the other computer. The participant was also told that he could take notes either on paper or in a word processing document and would be able to refer to those notes when actually performing the task. The video could be paused or replayed by the participant as needed. For Task 2, the video did not include the same errors that would be introduced to the experimental task, but the instructor in the video did demonstrate interface uses that would be helpful in resolving such issues.
2. *Viewing of the video tutorial on usability issues* (10 minutes). The participant then viewed a video tutorial on how to identify and report usability issues, which were defined in the video as anything in the system that is overly confusing or difficult to understand, requires too much effort, causes difficulties in performing the whole task or individual steps, or leads to frustration. It was stressed that it was the system, not the participant's performance, that was being evaluated. The instructor identified three issues that arose while she was performing different tasks from those in the experiment. To report an issue, the participant was instructed to click on the "Report" button appearing on the screen with the ERP client and then type the problem description into the text area opened in response to the click. He was also instructed to speak out loud about the problem.
3. *Performing the task* (45 minutes) The participant was provided with the task description document (see fig. 2) for either Task 1 or Task 2. This document contained the data required for performing the task but did not provide detailed instructions. The participant was also reminded that he could access any notes taken during training and could call upon a researcher if he was unable to proceed without external help. The ERP client was launched for the participant, who then proceeded with the task.
4. *Retrospective reporting* (30 minutes) Upon task completion, a researcher and the participant reviewed the video recording of the task session and the issues that had been logged with the Report button feature. The participant was also invited to discuss any issues that had not been reported for any reason, with the video serving as a reminder. For each user-reported issue identified either during task performance or in the review, the participant was asked for the following information: (1) Problem description, (2) Confusion level from 1 (not confused) to 4 (extremely confused), (3) Frustration level from 1 (not frustrated) to 4 (extremely frustrated), (4) What the participant had expected to happen, and (5) How the participant resolved the issue.

CC Walkthrough Methodology

CC walkthroughs were performed in a laboratory setting by two-person teams of usability professionals. The teams were randomly assigned to the two tasks, and each performed their evaluations independently. Several days prior to their scheduled evaluation, team members were sent training materials to review on the task and on the CC method.

At the time a walkthrough was conducted, all system interactions and verbalizations of evaluators were recorded by



Team	Task	ERP Experience		Usability Evaluation		Total Years in Field
		person1	person 2	person1	person 2	
A	1	novice	inter.	expert	inter.	1
B	1	novice	novice	accomp.	inter.	4
C	2	inter.	novice	accomp.	accomp.	15
D	2	novice	novice	expert	inter.	11

Table 2. Collaborative Critique participants.

screen- and audio-capture software. A total of seven teams performed the walkthrough. The results of three of those evaluations were discarded because of evaluators not completing the required training and failing to include explanations for questions in the template answered with a "no," as required by the CC method. Henceforth, all walkthrough references are to the results of the remaining four teams.

Participants

The eight participants in the CC walkthrough evaluations were randomly assigned to two-person teams, with two of those teams evaluating Task 1 and the other two evaluating Task 2. All were usability professionals with related graduate degrees. Table 2 contains summary information on the participants. The rankings for ERP experience uses the same three-level scale as in the user study. Participants ranked themselves on their proficiency in usability evaluation using this five-level scale: no experience, beginner, intermediate, accomplished, or expert. The last column in the table specifies the combined total number of years that the two evaluators in each team have spent performing usability evaluations in the field. Participants were compensated for their participation following the walkthrough session.

Setup

Each walkthrough team was provided with one desktop and one laptop computer. The CC template was on the desktop, which had a larger screen for easier viewing. The laptop was equipped with the task training video, the ERP client, and screen- and audio-capture software.

Protocol

Several days prior to their scheduled evaluation, each participant in the walkthrough team was sent the following materials to review for an estimated time of one hour: (1) Collaborative Critique documentation describing the method, including the CC questions, explanations of the purpose and intent of each question, and an example demonstrating a CC walkthrough evaluation of a basic calendar task, and (2) the task training documentation for the ERP task. In addition, they were sent the walkthrough session procedure and a questionnaire to complete and return by the start of their session.

The protocol for the CC walkthrough session, which was intended to fit into a three-hour timeframe, was as follows:

1. *Question and answer period with one of the researchers on the materials provided prior to the session* (5 minutes).

2. *Viewing of the task video tutorial* (Task 1 - 20 minute video, Task 2 - 22 minute video). The team members were shown the same task video tutorial as the users, in order to learn about the degree of the users' familiarity with the evaluated interface. Evaluators were not given the option of practicing along with the video.

3. *Review of evaluation materials* (10 minutes). The team was provided with a binder containing the following materials, which they were given time to review and ask questions about:

   - Task description document (same one as was given to the users).
   - Task action sequence.
   - Collaborative Critique documentation.
   - Task training documentation.
   - User specification, containing a general description of the users and assumptions about their knowledge in regard to performing the task with the ERP system.

   The team was then shown the CC template, in which the task was divided into subtasks, with each subtask containing one or more actions to be performed. Actions were further broken down into sub-actions as needed. For example, Task 2 includes the subtask of starting a new PO, which includes the action of specifying a vendor, which is comprised of five sub-actions. This structure corresponds to the step-by-step instructions in the task action sequence.

   The team was instructed to select one person to navigate the ERP client and the other to enter responses to each CC question after discussing them together. They were further instructed about the importance of providing explanations to any question responded to with an answer of "no."

4. *Performing the walkthrough* (120 minutes). The ERP client and recording software were then started, and the team performed the walkthrough in accordance with the task action sequence, with one team member navigating the ERP client and the other recording the agreed-upon responses to each question in the CC template.[1] Participants had access to all materials in the binder throughout the walkthrough, and a researcher was on hand should any questions arise.

5. *Review* (10 minutes). Upon completion, a researcher asked for any comments or questions on either the CC method or the experience overall.

DATA PROCESSING AND ANALYSIS

In this section, we describe the data processing and analysis and present findings from the user study and the walkthroughs, with evaluations of the Collaborative Critique based on outcomes from the user study.

[1] There was a deviation to the protocol concerning Team D, which had not finished the walkthrough within a three-hour timeframe. At the end of three hours, one of the team members chose to continue with the walkthrough, while the other did not. A researcher took over the task of transcribing the remaining team member's responses to the CC template in the interest of saving time, but there was no discussion of those responses.

User Study Data Processing: Usability Problem Instances

The first step in identifying usability problem instances involved a detailed review of the video recordings from the user sessions, with each of the three researchers responsible for the data from a subset of the participants. The user's actions were transcribed on a step-by-step basis to a template containing the task action sequence, and the following information was recorded: if the step was attempted (yes/no), if the user was successful in completing the step (yes/no), if there was a system error/warning message (description if yes), the exploration time if 15 seconds or more, and exploration notes documenting what happened during the exploration.

The user-reported issues entered with the Report button during task performance or documented during retrospective reporting were also transcribed to the template. For each issue, the following was recorded: the time of occurrence, the user's description of the issue, the researcher's summary of the user's response, the confusion and frustration levels reported by the user, what the user expected to happen, and how the user got around the issue.

The transcripts were then merged by task and reviewed by two or three researchers. In the process, the researchers identified and agreed upon additional issues that users had failed to report, which were characterized by long exploration times, non-completion of the step as specified, confused behavior that did not progress the task, etc. User-reported and researcher-identified issues within each step, herein referred to as *usability problem instances*, were then annotated in the merged transcript to indicate if the user was unable to complete the step, resulting in a negative effect on the final outcome of the task; requested help from a researcher to complete the step; pursued the wrong path as a result of the problem but was able to recover; or consulted notes taken during training to resolve the problem.

Problem instances were then classified by *problem type*, with repeated instances of the same problem assigned the same type identifier. In addition to the problem instances and problem types in our subsequent analysis of the data, we use the number of *problem types per step (PTPS)*, which refers to the set of distinct problem types within a step. PTPS is a useful level of abstraction because it considers problem instances within the step context while distinguishing one problem type from another. This allows the assessment of the breadth of different problem types predicted for each step of evaluation.

The number of all usability problem instances, what number and percent of them were user-reported, the number of PTPS, and the number of problem types identified by the researchers by task and across both tasks are presented in table 3.

Task	P Instances	User-Reported	PTPS	P Types
1	115	101 (88%)	93	52
2	162	117 (72%)	109	62
Both	277	228 (79%)	202	98

Table 3. Summary of usability problem information collected from the user study.

CC Prediction Statistics

The value of a usability evaluation method is largely reflected by its ability to predict the issues that users would actually experience. To that end, we evaluated each of the problem instances from the usage data in terms of whether it was *predicted* or *missed* by each of the walkthrough teams performing the evaluations.

A problem instance was marked as *predicted* if the response of the evaluation team to one or more of the CC questions for the step in which the issue was reported described the issue directly or identified its cause. In making this determination, the researcher reviewed the full record of the problem instance in the merged transcript, which included information collected from the user and the transcriptions of the video. In addition, some problem instances were marked as predicted due to evaluator responses to other steps. For example, the evaluators may have experienced the same error as the user but in a different step, referred to a usability problem as being *global*, or noted problems with the system's response that caused a user-reported problem in the subsequent step. For each problem instance marked as predicted, the researcher noted the step and the questions that contained the predicting response.

Any problem instances that were not predicted were marked as *missed*. Missed instances were further classified as *predictable* or *unpredictable.* A *predictable* issue was one that occurred when the user was performing a step in the correct task sequence or when the user was performing a step outside of that sequence and the walkthrough team also performed that step. A user may have performed such a step if she veered from the correct path, followed a different sequence in performing the task, or experienced unanticipated system behavior, such as an error caused by the loss of the internet connection. An evaluation team could have performed the same out-of-sequence step as a result of system exploration, which was encouraged by the method (see step 1 of the CC Procedure), or due to experiencing the same unanticipated system behavior as the user. A problem instance was considered *unpredictable* if it occurred in an out-of-sequence step that was not performed by the evaluation team or if the user-reported issue was associated with the quality of the data used in the ERP system configuration, which the CC evaluation is not meant to address.

Table 4 shows the number and percent of problem instances, problem types per step, and problem types that were predicted by the CC method out of those that were predictable by team, by the two teams per task, and by all teams. The percentages reflect the *thoroughness* of the method, which has been defined by Sears [20] as the proportion of the number of real problems found to the number of real problems that exist. The teams assigned to Task 2 had higher thoroughness ratings than those assigned to Task 1, both individually and together. This is partly attributable to the Task 2 teams engaging in more exploration of the system than the Task 1 teams, thereby encountering more of the issues experienced by users in out-of-sequence steps. In addition, these teams typically provided more detailed explanations in their responses to the CC questions, which resulted in more issues being identified within each step. The differences in Task 1 and Task 2 teams' approaches to the evaluation may be a result of the Task 2 teams having much more field experience than the Task 1 teams (see table 2).

	P Instances		PTPS		P Types	
Team	predicted (predictable)	%	predicted (predictable)	%	predicted (predictable)	%
Team A	41(98)	**42**	32(77)	**42**	22(45)	**49**
Team B	38(97)	**39**	28(78)	**36**	21(45)	**47**
Team C	102(135)	**76**	65(92)	**71**	38(56)	**68**
Team D	89 (133)	**67**	58(89)	**65**	36(56)	**64**
Task 1 (A & B)	56(104)	**54**	44(83)	**53**	28(48)	**58**
Task 2 (C & D)	117(144)	**81**	78(98)	**80**	45(58)	**78**
All	173(248)	**70**	122(181)	**67**	66(93)	**71**

Table 4. Thoroughness measures for CC method.

Predictions by Severity

Severity ratings are used to differentiate significant issues from more trivial ones, thus adding another dimension to the thoroughness of an evaluation. We assigned severity ratings on usability problem instances based on user-reported confusion and user reported frustration (both ranging from 1 to 4, with possible intermediary values such as 2.5), exploration time (in seconds), success in completing the step, and effect on task outcome. We considered confusion and frustration levels from 1 to 2 to be indicative of less severe problems, while levels from 3 to 4 indicated more severe problems. Confusion was considered to be somewhat more important than frustration, since it is more likely to hinder user performance; exploration times were therefore compared to user-reported confusion levels. We found that no confusion on a step was typically associated with exploration times of 30 seconds or less, low levels of confusion with exploration times in the 30 to 60 second range, higher levels of confusion with 60 to 120 seconds, and the highest confusion with 120 or more seconds.

Problem instances were rated on a three-point scale ranging from 1 for least severe to 3 for most severe. The *most severe* rating was assigned to any problem instance that had a negative effect on the outcome of the task. It was also assigned to instances in which the user was not successful in completing the step in which the issue occurred and spent at least 120 seconds on exploration, or the user had both confusion and frustration levels of at least 3, or the user had either confusion or frustration levels of 3 or more and spent at least 120 seconds on exploration. An example is where a user reported a confusion level of 4 when it took over 6 minutes to identify the correct button to press. The *least severe* rating was assigned to any problem instance in which the user was able to complete the step in which the issue occurred but either had a confusion level of less than 3 and exploration of 30 seconds or less or had a confusion level of 2 or less, a frustration level below 3, and exploration in the range of 30 to 60 seconds, inclusive. For example, a user reported a confusion level of 2 but spent no additional time exploring when confronted by two buttons right next to each other that appeared to provide the same functionality. The *medium severity* rating was assigned to all problem instances that did not meet the criteria for either most or least severe. As an example, a user reported 2.5 for confusion, 2 for frustration, and spent over 1 minute figuring out how to create a new user due to a reported lack of guidance on how to proceed.

Of the 277 unique problem instances identified by users (see table 3), 85 have severity ratings of 3, 49 have severity ratings of 2, and 143 have severity ratings of 1. Table 5 shows predicted problem instances by severity rank for each team, the teams on each task, and overall.

	Severity 1		Severity 2		Severity 3	
Team	predicted (predictable)	%	predicted (predictable)	%	predicted (predictable)	%
Team A	26(61)	**43**	6(12)	**50**	9(25)	**36**
Team B	19(59)	**32**	6(15)	**40**	13(23)	**57**
Team C	48(66)	**73**	21(27)	**78**	33(42)	**79**
Team D	40(65)	**62**	20(26)	**77**	29(42)	**69**
Task 1 (A & B)	33(63)	**52**	9(15)	**60**	14(26)	**54**
Task 2 (C & D)	54(68)	**79**	25(28)	**89**	38(48)	**79**
All	87(131)	**66**	34(43)	**79**	52(74)	**70**

Table 5. Thoroughness by severity level for problem instances.

Notably, the prediction rates are relatively uniform across the severity levels for all teams, individually and grouped. This suggests that the CC method avoids the common problem of detecting a larger proportion of trivial problems.

Evaluator Mispredictions

We computed the number of walkthrough steps in which evaluators only predicted usability problems that users never experienced. If a walkthrough step predicted at least one problem instance, it was not counted as a false positive. We have identified a total of four false positives for Task 1, representing 7.7% of the 52 steps. For Task 2, three false positives were identified, representing 4.5% of the 66 steps. Overall, 7 false positives, or 5.9% of the 118 steps, were identified.

This analysis most likely understates the number of false positives when compared to assessments made on a problem rather than a step basis. This is a limitation of our study, which is due to the fact that we did not ask evaluators to derive a list of usability issues based on their observations.

We also assessed how accurately the teams' responses to question 2(a) (exploration) predicted the statistics on how many users spent time on exploring the interface and how long they spent. The data showed evaluators commonly overestimating the number of people unable to figure out what to do. For two of the teams, higher scores in question 2(a) corresponded to higher levels of exploration by the users, but there was no such relationship for the other two teams. Question 2 is the only one that asks evaluators to consider users

Problem Type (Frequency)
Insufficient guidance with error message (12)
Did not look up plant for material (10)
Confusion over warning concerning record flagged for deletion (6)
Confusion about fields in search interface (5)
Inappropriate error, avoidable with clearly marked required fields (3)
. . .

Table 6. Five most frequent error-related usability problem types for Task 2.

	errors critiqued	out of **9** in user-error set	%	out of **3** seeded errors	%
Team C	8	6	**67**	2	**67**
Team D	7	5	**56**	3	**100**
together	12	8	**89**	3	**100**

Table 7. Coverage of errors by evaluator teams in Task 2.

with a "range of experiences." This aspect of the question may have caused difficulties in the assessment due to the various ways that the evaluators might have accounted for that range. Method adjustments are needed to mitigate this issue.

Detecting and Evaluating Errors

To test the effectiveness of the method with respect to evaluating error-related system behavior, we have designed Task 2 (Create Purchase Order) to include three potential (*seeded*) errors by presenting the data in the user's task description in a way that did not match the system-prescribed order of data entry and including one incompletely specified task parameter. The action-step sequence given to the evaluator teams exposed them to the same set of seeded errors. The designed task with seeded errors represented a realistic scenario.

User data from Task 2 revealed that users experienced 13 different error situations, including the 3 seeded errors. Of these 13, the expert review identified 9 as having been a cause of a usability problem. Henceforth, we refer to these 9 problems as the *user-error set*. The two evaluator teams critiqued 12 different error situations, of which 8 were found in the user-error set.

We identified 48 problem instances, 31 PTPS, and 14 problem types related to the user-error set. A problem instance was deemed related to an error if it involved the user's inability to see or understand the error or warning message, the user's inability to respond to an error with an appropriate fix, or user frustration at the inappropriateness of the error to the situation. The five most frequent error-related problem types are shown in table 6.

Table 7 shows the error evaluation rates by the teams with respect to the user-error set and the set of seeded problems. As we can see, the evaluator teams detected and evaluated 89% of the errors in the user-error set.

The summary of the error-related prediction rates is presented in table 8. In each row, the data with respect to all evaluated errors is followed by the the seeded error data. The prediction rate data is shown by problem instance, PTPS, and problem type for all errors critiqued by a team and for the seeded errors. Within each of these categories, the number of predicted versus predictable items and the resulting prediction rate are reported. Note, however, that even if the rates were computed over the instances corresponding to the entire user-error set instead of only the predictable ones, they would remain high at 90%, 94% and 93%, respectively, for teams C and D taken together.

As we can see, the critique method led to an assessment of more than twice as many error situations as were seeded and covered all but one of the errors that caused usability problems in the user tests. There were four critiqued errors that did not predict any usability issues from Task 2. Of these four, two predicted problems related to the system login procedure and were experienced by users working on Task 1. The third was a false alarm, which predicted a problem in a situation where users did not experience any issues, and the forth evaluated a Task 2-specific error that was not experienced by any users.

DISCUSSION AND CONCLUSIONS

We have presented a novel usability evaluation method, called Collaborative Critique, and presented the results of its initial testing in which predictions derived with our method were compared to the usability problems experienced by users in a comprehensive laboratory study.

The results of the initial evaluation of the CC method presented here are promising, with our analysis confirming the soundness of this method. The observed strengths of CC lie in the specific attention to the error-related behavior of the system and its ability to uniformly detect problems across the severity spectrum. The seven CC questions revealed systemic as well as localized usability problems, as reflected by the problem instance and PTPS prediction rates. The four teams predicted 70% of all predictable usability instances, 71% of problem types, and 67% of problem types per step. On the task that included error evaluations, the error-related prediction rates ranged from 93% to 100%.

A distinguishing factor of our evaluation is the context of the ERP domain in which it was conducted. The number of usability problem instances identified in our user study attests to the scope of issues experienced by users of ERP systems, with 115 instances in the 52-step Authorizations task (Task 1) and 162 in the 66-step Purchase Order task (Task 2). Overall, 277 unique problem instances were identified.

The main limitation of the presented evaluation is the small number of walkthroughs performed. Because of this, the quantitative measures are only initial indicators of the efficacy of the method. Further studies are necessary to more thoroughly assess the strengths and weaknesses of the CC. There were obvious differences in the performances of the individual teams per task, with the two teams on Task 2 predicting noticeably larger proportions of usability problems. Our analysis of these differences shows that they can be at least partially attributed to the Task 2 teams engaging in more exploration of the system, and by the fact that these teams provided more detailed responses to the CC questions. This ex-

	all critiqued errors						seeded errors					
	P Instances		PTPS		P Types		P Instances		PTPS		P Types	
Team	predicted (predictable)	%	predicted (predictable)	%	predicted (predictable)	%	predicted (predictable)	%	predicted (predictable)	%	predicted (predictable)	%
Team C	37(39)	**95**	23(24)	**96**	10(11)	**91**	20(23)	**87**	12(14)	**86**	6(7)	**86**
Team D	25(32)	**78**	15(21)	**71**	6(10)	**60**	19(23)	**83**	11(14)	**79**	4(7)	**57**
together	43(45)	**96**	29(31)	**94**	13(14)	**93**	22(23)	**96**	14(14)	**100**	7(7)	**100**

Table 8. Summary of prediction rates of error-related problem instances in Task 2.

planation can only be borne out by conducting a larger scale evaluation. In addition, these results point to improvements that can be made to our method by honing our training materials and instructions to the teams.

Only some aspects of the data collected from this evaluation have been presented here. Additional quantitative and qualitative analyses are on-going, as is the fine-tuning of the CC questions and procedure.

ACKNOWLEDGEMENTS

We thank our study participants, without whom this work would not be possible.

This material is based in part upon work supported by the National Science Foundation under Grant No. 0819333. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Akers, D., Simpson, M., Jeffries, R., and Winograd, T. Undo and erase events as indicators of usability problems. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, ACM (New York, NY, USA, 2009), 659–668.
2. Albers, M., and Still, B., Eds. *Usability of Complex Information Systems: Evaluation of User Interaction*, 1st ed. CRC Press, Inc., Boca Raton, FL, USA, 2010.
3. Andre, T. S., Hartson, H. R., Belz, S. M., and McCreary, F. A. The user action framework: a reliable foundation for usability engineering support tools. *Int. J. Hum.-Comput. Stud. 54* (January 2001), 107–136.
4. Babaian, T., Lucas, W. T., Xu, J., and Topi, H. Usability through system-user collaboration. In *DESRIST*, Lecture Notes in Computer Science, Springer (2010), 394–409.
5. Bratman, M. E. Shared cooperative activity. *Philosophical Review 101*, 2 (1992), 327–341.
6. Capra, M. G. Contemporaneous versus retrospective user-reported critical incidents in usability evaluation. *Human Factors and Ergonomics Society Annual Meeting Proceedings 46*, 24 (2002), 1973–1977.
7. Castillo, J. C., Hartson, H. R., and Hix, D. The user-reported critical incident method at a glance. In *Proc. SIGSOFT 2002/FSE-10*, ACM Press (1997), 41–50.
8. Chilana, P. K., Wobbrock, J. O., and Ko, A. J. Understanding usability practices in complex domains. In *Proceedings of the 28th international conference on Human factors in computing systems*, ACM (New York, NY, USA, 2010), 2337–2346.
9. Cockton, G., and Woolrych, A. Sale must end: should discount methods be cleared off hci's shelves? *Interactions 9*, 5 (2002), 13–18.
10. Cockton, G., Woolrych, A., and Lavery, D. The human-computer interaction handbook. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2008, ch. Inspection-based evaluations, 1171–1190.
11. Grosz, B. G. Beyond mice and menus. *Proceedings of the American Philosophical Society 149*, 4 (12 2005), 529–543.
12. Grosz, B. G., and Kraus, S. Collaborative plans for complex group action. *Artificial Intelligence 86*, 2 (1996), 269–357.
13. Hartson, H. R., Andre, T. S., and Williges, R. C. Criteria for evaluating usability evaluation methods. *Int. J. Hum. Comput. Interaction 15*, 1 (2003), 145–181.
14. Lewis, C., and Wharton, C. Cognitive walkthroughs. In *Handbook of human-computer interaction*, M. G. Helander, T. K. Landauer, and P. V. Prabhu, Eds. Elsevier, 1997, 717–732.
15. Mirel, B. *Interaction Design for Complex Problem Solving Developing Useful and Usable Software.* Morgan Kaufmann, San Francisco, CA, 2004.
16. Nielsen, J. Heuristic evaluation. In *Usability inspection methods*, J. Nielsen and R. L. Mack, Eds. John Wiley & Sons, New York, NY, 1994, 25–62.
17. Norman, D. A. Cognitive engineering. In *User Centered System Design: New Perspectives on Human-Computer Interaction*, D. A. Norman and S. W. Draper, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.
18. Oja, M.-K., and Lucas, W. Evaluating the usability of erp systems: What can critical incidents tell us? In *Proceedings of Pre-ICIS Workshop on ES Research* (Saint Louis, Missouri, 2010).
19. Redish, J. G. Expanding usability testing to evaluate complex systems. *Journal of Usability Studies 2*, 3 (May 2007), 102–111.
20. Sears, A. Heuristic walkthroughs: Finding the problems without the noise. *Int. J. Hum. Comput. Interaction 9*, 3 (1997), 213–234.
21. Terveen, L. G. Overview of human-computer collaboration. *Knowledge-Based Systems 8*, 2-3 (1995), 67–81.
22. Topi, H., Lucas, W., and Babaian, T. Identifying usability issues with an ERP implementation. In *Proc. of ICEIS-05* (2005), 128–133.
23. Wharton, C., Rieman, J., Lewis, C., and Polson, P. The cognitive walkthrough method: A practitioner's guide. In *Usability inspection methods.*, J. Nielsen and R. L. Mack, Eds. John Wiley & Sons, New York, NY, 1994, 105–140.
24. Wixon, D. R. Evaluating usability methods: why the current literature fails the practitioner. *Interactions 10*, 4 (2003), 28–34.