Challenges in evaluating complex IT security management systems

Pooya Jaferian, Kirstie Hawkey, Konstantin Beznosov
University of British Columbia, Vancouver, Canada
{pooya,hawkey,beznosov}@ece.ubc.ca

ABSTRACT

Performing ecologically valid user studies for IT security management (ITSM) systems is challenging. The users of these systems are security professionals who are difficult to recruit for interviews, let alone controlled user studies. Furthermore, evaluation of ITSM systems inherits the difficulties of studying collaborative and complex systems. During our research, we have encountered many challenges in studying ITSM systems in their real context of use. This has resulted in us investigating how other usability evaluation methods could be viable components for identifying usability problems in ITSM tools. However, such methods need to be evaluated and proven to be effective before their use. This paper provides an overview of the challenges of performing controlled user studies for usability evaluation of ITSM systems and proposes heuristic evaluation as a component of usability evaluations of these tools. We also discuss our methodology for evaluating a new set of usability heuristics for ITSM and the unique challenges of running user studies for evaluating usability evaluation methods.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Human Factors

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

In this paper, we present our methodology for evaluating a new set of usability heuristics for IT security management technologies. We also discuss the challenges of evaluating a new usability evaluation method and our approach to address the challenges. Our need for new usability heuristics arose during our attempt to evaluate the usability of a complex IT security management technology. User studies are often seen as the optimal path to evaluation; but sometimes it can be difficult to perform an ecologically valid evaluation. Evaluating information technology (IT) security technologies is one of the more challenging domains in which to perform an ecologically valid user study. Laboratory experiments can have little validity due to the complexity of real-world security problems and the need to situate a specific technology within a larger organizational context. Furthermore, it is difficult to recruit IT security practitioners for simple interviews, let alone field observations [2, 12]. Direct observation of tool use can be time consuming as much security work is spontaneous (e.g. security incident response [24]) or occurs over many months (e.g., deploying an identity management system [8]). As ITSM tool use is intrinsically cooperative, its study inherits the difficulties of studying cooperation [15].

Our research has the goal of designing usable identity management systems. While we have been able to perform seven interviews with security practitioners during the past year, we have not yet performed any observations or contextual interviews. Due to the nature of security, security practitioners are reluctant to give away information about their identity management system; and they do not like to be observed while they are working with the system. It is also difficult to be able to recruit them for a lab study. Security practitioners are very busy with their daily activities and contacting them by email often leads to no response. Furthermore, they are well paid; and it is hard to attract them with the honorarium paid in academic user studies.

As a result, a low cost usability evaluation method could be a desirable component of a usability evaluation, particularly if it may help identify the major usability problems in ITSM tools before costly user studies are conducted. We suggest that Heuristic Evaluation (HE) is a candidate approach. HE is based on a set of usability principles called heuristics. An evaluator inspects a user interface and identifies usability problems and their severity based on heuristics and his judgement of the interface. A survey by Vredenberg et al. [23] shows HE as the most popular informal usability evaluation technique. A study by Jeffries et al. [10] shows HE identifies more serious problems compared to usability testing, guidelines and cognitive walkthrough. Nielsen's theoretically grounded and extensively tested heuristics [18] are the most widely accepted heuristics.

Symposium on Usable Privacy and Security (SOUPS) 2010, July 14–16, 2010, Redmond, WA USA
.

Nielsen's heuristics were developed based on the theory of action [19], in which the unit of analysis is an individual action that consists of a human actor and a physical system. Yet, ITSM involves multiple actors across the organization working with different artifacts. ITSM activities are distributed across time and space and they require collaboration between different stakeholders working in an organizational context with certain rules and norms. As a result, Nielsen's heuristics will have difficulties in accounting for the characteristics of the ITSM domain.

Prior research has extended or adapted Nielsen's usability heuristics for a specific domain (e.g., ambient displays [13], video games [20], virtual reality [22], medical devices [25], intelligent tutoring systems [14], and intrusion detection systems [26]). They also have developed new heuristics based on a specific theory that addresses characteristics of the target domain (e.g. heuristics based on locales framework [3] for the evaluation of groupware [4], heuristics based on the mechanics of collaboration [5] for the evaluation of shared visual work surfaces for distance-separated groups [1]).

We have developed a set of usability heuristics (Table 1) with a focus on the social and collaborative aspects of ITSM by interpreting our previously developed guidelines [9] and using the theoretical lens provided by activity theory [11]. These heuristics still need to be proven to be effective in finding usability problems and shown to be useful as discount usability evaluation method. In this paper, our goal is to review the methodology that we designed to validate the set of heuristics, the challenges in designing such a methodology, and our approach in addressing the challenges. Currently, we are piloting the study; therefore, the focus of this paper is on the methodological details.

Table 1: Seven heuristics for evaluation of IT security management tools

1	Visibility of activity status
2	History of actions and changes on artefacts
3	Flexible representation of information
4	Rules and constraints
5	Planning and dividing work between users
6	Capturing, sharing, and discovery of knowledge
7	Verification of knowledge

2. RESEARCH QUESTIONS

While our ITSM heuristics are grounded in empirical data and supported by theory, the use of the heuristics should be tested in a standard HE process. Our goal is to answer the following research questions:

1. Can ITSM heuristics find unique usability problems in ITSM tools that can't be found by Nielsen's heuristics?

2. Are ITSM heuristics more useful than Nielsen's heuristics in finding usability problems in ITSM tools?

3. What are the characteristics of HE using ITSM heuristics? How are these characteristics compared to those of Nielsen's heuristics? Such characteristics include: number of evaluators, performance of evaluators using the heuristics, required HCI or computer security background, evaluators' feedback about the heuristics)

3. METHODOLOGY

To answer the research questions, we will perform a comparative evaluation of the ITSM heuristics with Nielsen's heuristics, using a between-subjects design. One group will use Nielsen's heuristics (Nielsen condition) and the other group will use the ITSM heuristics (ITSM condition). Participants will take part in one of the sessions related to their condition. An overview of the evaluation methodology is provided in Figure 1. The details of each step in the evaluation protocol are presented in Section 3.2.

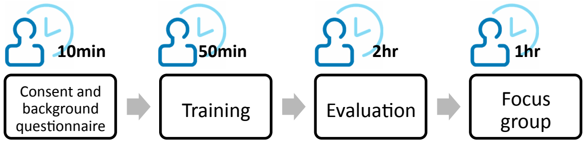

Figure 1: Study protocol overview

3.1 Independent and Controlled Variables

The independent variable in our study is the set of heuristics used in each condition. We will also control several variables across the two conditions:

1. Received training: Similar training material has been prepared for the two conditions. The introduction to HE, IdM system and IdM system demo will be identical between the two conditions. For training participants on HE using each set of heuristics, we developed separate sets of training materials for each condition. However, we use the same structure and running example in both sets. Training will be provided to the participants using pre-written scripts to ensure we deliver the same content in each session. To ensure that the design of the training material is not biased towards one set of heuristics, the training material has been reviewed by all researchers working on the project.

2. Background of participants in each condition: We will attempt to balance the expertise of the participants between the two conditions by screening them during recruitment. Participants will be screened and an experience metric (see Section 4.2) will be calculated for them before their assignment to a condition. Also we will only recruit participants who are familiar with HE.

3. Evaluation session length: Unlike two similar studies [1, 13], we will perform the HE session in a controlled environment to control the time participants spend on evaluation.

4. Evaluated systems: An instance of the IdM system is installed on a VMWare machine. Each participant has remote access to an instance of VM; therefore, each participant has access to an identical system. All VMs will be reset after each evaluation session.

3.2 Study Protocol

We now describe the components of our study protocol.

Consent and background questionnaire: We will begin the evaluation session by obtaining participants' consent and then ask them to complete a background questionnaire. In the questionnaire, we will obtain demographic information and collect data to assess the background of the participants on HCI and computer security. This questionnaire is similar to the screening questionnaire; but we will also collect qualitative data to clarify quantitative data (e.g. we ask for a list of HCI courses in addition to the number of courses).

Training: We will provide training on HE for the participants and then described the heuristics that they will use in HE. We will demonstrate the use of heuristics with a running example of evaluating a network firewall system, an ITSM tool. We will conclude the training session with an introduction on the IdM system, the specific ITSM tool that they will evaluate. The training material will be presented to the participants using a pre-written script to ensure that all the groups receive the same training.

Evaluation: The participants will inspect the interface individually. Each participant will have access to an instance of the IdM system and all the instances are identical. Since the IdM system is complex and it is not possible to cover the whole software in one evaluation session, we limit the scope of the evaluation to few typical usage scenarios [21]. The scenarios were designed for an example organization, and each IdM instance is configured with the users and structure of the example organization. Besides limiting the scope, the scenarios should guide evaluators who are not domain experts in performing tasks on the IdM system. We provide participants with the list of scenarios and ask them (1) to identify the usability problems using the provided set of heuristics, and (2) to specify the scenario and the heuristic for each problem found. Participants will enter the identified problems in an online form and will have two hours to perform the evaluation.

Focus Group: After the evaluation session, participants will be provided with a post-evaluation questionnaire to rate their experience in using heuristics. We will then conduct a focus group session to discuss their experience in using the heuristics, capture their suggestions for improving heuristics, and discuss usability problems that cannot be associated with any of the heuristics. We chose to add this step as qualitative data can better show whether the heuristics are useful, easy to use, and easy to apply. Furthermore, we will be able to probe the usability of the interface in general and discuss usability issues that might be related to neither Nielsen's nor the ITSM heuristics.

We have piloted and refined our study through several iterations during which we refined the description of heuristics, our study material, training, and the way the material is presented to the participants.

4. STUDY DESIGN CHALLENGES

In this section, we review the challenges of the study design and our approach in addressing them:

4.1 Participant Recruitment

Recruitment of a representative sample for this study is one of the major challenges. Unlike user studies that can be performed with participants from the general population, our study requires participants with a specific background. Normally, those who perform HE have prior background on the method and have experience with performing evaluation so an HCI background and familiarity with HE is our main inclusion criteria. To recruit participants, we will send emails to all graduate students in the Computer Science and Electrical and Computer Engineering departments of UBC. We will also send emails to the user experience mailing list in Vancouver in order to reach participants with professional HCI experience. Furthermore, we plan to contact the instructors of HCI courses and ask them to distribute the recruitment letter. All participants will be given a \$50 honorarium for their participation. To balance the expertise of participants in each group, we will screen them before recruitment and have them rate their expertise in computer security and HCI. We will use the experience metric provided in Section 4.2 to determine their expertise.

4.2 Determining Participants' Experience

As Nielsen mentioned in his original HE experiment [16], the evaluators' HCI and domain expertise are two factors that impact the quality of evaluation. For the recruited evaluators, we will measure their HCI and computer security expertise and we use them later to find their impact in identifying usability problems.

As we have two conditions, we will not further divide them based on the experience of participants (e.g., having novice, expert, double expert conditions) as Nielsen [16] and Baker [1] did in their experiments. We will limit our analysis to the correlation between background and the characteristics of evaluation results.

Table 2: Variables for calculating experience metrics

Parameter	Description
$Experience_{HCI}$	Years of HCI research or professional experience
$Training_{HCI}$	Number of courses on HCI
$Training_{HE}$	1 point if HE experience/training
$Experience_{HE}$	Number of HEs performed
$Experience_{CSR}$	Years of computer security (CompSec) research experience
$Experience_{CSP}$	Years of CompSec professional experience
$Training_{CS}$	Number of CompSec courses
$Experience_{RBAC}$	1 point if has managed a role-based access control (RBAC) system for <10 users, 2 points for 10-50 users, 3 points for >50 users.
$Experience_{ORG}$	1 point if has worked in an organization that uses RBAC

In order to do this correlation, we need to quantify participants' experience. We will adopt an approach similar to Iachello [7], who calculated a score for design experience of participants based on years of relevant experience and number of relevant courses. We will calculate two experience metrics, $Score_{HCI}$ (HCI expertise score) and $Score_{CS}$ (computer security expertise score), based on a set of variables (Table 2) that will be collected through a background questionnaire:

$Score_{HCI} = Experience_{HCI}$
$+ \frac{1}{3}(Training_{HCI} + Training_{HE} + Experience_{HE})$

$Score_{CS} = Experience_{CSP} + \frac{1}{2}Experience_{CSR} +$
$\frac{1}{3}(Training_{CS}) + Experience_{RBAC} + Experience_{ORG}$

In the calculation of $Score_{HCI}$, we considered each course equal to a semester long experience in HCI. Furthermore, since the focus of the study is HE, we considered specific training on HE and each HE performed equal to the experience from a course. In the calculation of $Score_{CS}$, we give more weight to professional experience in computer security compared to research experience, as people with professional experience have more practical access control and identity management experience.

4.3 Data Preparation

To generate a set of usability problems as the output of HE in each condition, we need to synthesize the usability problems identified by each evaluator and produce an aggregated list of problems. The results of our pilot study showed that evaluators find problems at different levels of granularity, find duplicate problems, and state problems using different terminology. In order to aggregate the problems, they need to be reviewed and synthesized by the researchers. Furthermore, we observed that evaluators find out of scope problems, which are usability problems that cannot be classified in one of the heuristics. These problems need to be removed from the list of problems identified by the set of heuristics. Furthermore, we observed mistakes in the assignment of heuristics and problems. In order to have a consistent and repeatable methodology of synthesizing the problems, we use the following steps. These steps will be performed by two researchers in order to reduce experimenter bias in the process.

1. Decomposing problems: If any of the problems consists of multiple fine grained problems, we will decompose the problem. Examples of a compound problem would be problems that refer to different actions, different artifacts, or different mechanisms in the interface. Further, if a problem is system wide and occurs in different scenarios, we will consider each occurrence of the problem as a unique problem. There are two reasons for our interest in decomposing problems to finest level of granularity. First, when we aggregate results from different evaluators, we will not find a problem that is subset of another problem. Second, each part of a compound problem might have a different severity and, therefore, a different priority for fixing.

2. Removing duplicates: If any of the problems is an obvious duplicate, we will remove it.

3. Removing false positives: We will remove any identified problems with the following characteristics as they are considered to be false positives: (1) the evaluator did not understand the rationale behind the interface, (2) there is a known technical bug in the system that causes the problem.

4. Finding out of scope problems: Two researchers will analyze the list of identified problems to determine if a problem is out of scope or if the assignment of the problems to heuristics is wrong. Each researcher will individually go through the list of problems, mark the out of scope problems, and assign the heuristics to problems without looking at the evaluators' assignment of problems to heuristics. Then each researcher will compare his assignments and list of out of scope problems to the evaluators' assignments and adjust his previous decision of he believes the evaluators assignments were more accurate. Finally, the researchers will compare their assignments and the list of out of scope problems with each other. Any discrepancies will be discussed and the researchers will come to a consensus about the final list of problems.

We will use the method shown in Table 3 to determine the final set of out of scope problems and the assignments between problems and heuristics. This table shows the action that is applied to an identified problem after comparing the evaluators' decision to the researchers' decision. For a particular usability problem, the evaluators' decision can be assigning the problem to a certain heuristic(H$_i$, H$_j$) or to no heuristic(no assignment) and the researchers' decision can be assigning the problem to a certain heuristic(H$_i$, H$_j$) or marking the problem as out of scope(out of scope). The researchers' decision always overrides the evaluators' decision. The reason for this is twofold: first, we can argue that the researchers' decision is more reliable as they have a chance to revise their assignment after reviewing the evaluators' assignments and they also have a chance to discuss any conflicts between themselves. Second, if we leave the evaluators' assignment unchanged, we may have a hard time aggregating problems that are conceptually the same but are assigned to different heuristics.

Table 3: Method for identification of out-of scope problems and reassignment of problems to heuristics

Evaluator/ Researcher	H$_i$	H$_j$	Out of scope
H$_i$	Keep	Re-assign to H$_j$	Out of scope
H$_j$	Re-assign to H$_i$	Keep	Out of scope
No Assignment	Re-assign to H$_i$	Re-assign to H$_j$	Out of scope

4.4 List of known usability problems

One of the major challenges in comparing usability evaluation methods is having a list of known usability issues in an interface to compare with the output of each evaluation method. Hartson et al. [6] propose four methods for generating the list of known usability problems: (1) seeding the target design with usability problems, (2) finding usability problems through lab based testing, (3) asymptotic usability testing, and (4) union of usability problem sets.

In the context of our study, each approach has certain limitations. First, we cannot seed the target system with usability problems as we cannot modify the source code. Moreover, the validity of the problems used to seed the interface is questionable as they might be hypothetical and biased towards our heuristics. Second, lab testing is shown to find different types of problems as compared to HE [10].

Therefore, neither lab testing nor asymptotic usability testing provides a comprehensive list of usability problems. In similar studies evaluating new sets of heuristics, researchers used the aggregate of all problems from different evaluators and the researcher [18], the aggregate of all problems from different evaluators [1], and expert review [13] to generate the list of known usability problems.

In this study, we will use the aggregate of individual lists of problems from different evaluators as the list of known usability problems. We will not include the list of problems generated by the authors as it might be biased towards one set of heuristics.

After the problem synthesis stage, two researchers will analyze the list of problems and generate an aggregated list. In this process, each researcher starts with an empty list of aggregated problems. Each usability problem is then compared with the problems in the aggregated list. If the problem does not exist in the list, it will be added to the list. Otherwise, the description of the problem in the aggregated list is refined based on the description of the usability problem. The evaluators will compare their list of problems and discuss any inconsistencies until consensus is reached.

4.5 Assigning Severity ratings

There are several approaches to assigning severity ratings to usability problems. Nielsen [17] argues that assigning severity rating by the evaluators during the evaluation is not reliable. He recommends that the severity rating should be assigned after the experiment, when the evaluators have access to the the full list of problems. He also argues that assigning severity rating should be done by at least 3 evaluators. In his experiments, Nielsen used different numbers of severity raters including 1, 2, and eleven raters. In our study, we will follow Nielsen's original recommendation and use the mean of the severity ratings from three HCI researchers who participated in our pilot study. Furthermore, we will follow Nielsen's approach [16] to classifying the severity of problems and use two levels of severity (Major and Minor). Based on Nielsen's definition, "Major usability problems are those that have serious potential for confusing users or causing them to use the system erroneously while minor problems may slow down the interaction or inconvenience users unnecessarily" [16].

We extend Nielsen's definition to include the problems that might impact multiple stakeholders or the overall activity:

Major: the problem has a serious potential for confusing the user or any of the involved stakeholders in the activity, or causing them to use the system erroneously.

Minor: the problem might slow down the interaction of different stakeholders with the system or produce inefficiencies in the activity.

4.6 Scenario-based HE

Since the target system for evaluation is a highly domain specific system with a wide-range of functionality, one of the major challenges is asking participants to evaluate such a large scale system during a 2 hour evaluation session with only a brief introduction to the system in the training session. While similar studies [16, 1, 13] allow participants to freely explore the interface and identify problems, we limited the scope of the evaluation to four usage scenarios. These scenarios will allow participants to focus their attention to a subset of system and also allow us to give them some information about the context in which the tasks are performed in the real world. To ensure the ecological validity of the scenarios, we used previous findings from our field study of IdM systems [8] in the design of scenarios. This approach is inline with Nielsen's recommendation of supplying evaluators with typical usage scenarios when the target system is domain-specific.

5. HE AND USABLE SECURITY

In this paper, we have proposed the use of ITSM heuristics to evaluate ITSM systems. We are not proposing that these heuristics be used in the evaluation of the more general domain of usable security tools for end users. The ITSM heuristics are aimed to address usability issues in a complex social and organizational context in which IT security is considered a primary activity. However, the focus of usable security in general is on devising usable mechanisms for a single user to manage security as a secondary task. We therefore do not believe that our heuristics should be extended outside of the ITSM domain, except perhaps to other tools used in a complex social and organizational context, such as general IT tools used by IT practitioners.

We do, however, believe that HE can be a viable approach in the design of usable yet secure end user technologies. There could be multiple future research directions investigating the use of HE by the usable security community. First, we invite researchers to compare the effectiveness of user studies to HE in finding usable security problems. Second, a new set of heuristics or an extension to Nielsen's heuristics might be helpful in finding usability problems in security mechanisms. Developing such heuristics would first require developing and refining a set of principles for designing usable security mechanisms.

6. CONCLUSION

In this paper, we discussed the challenges of evaluating the usability of complex IT security management technologies with security practitioners and other pertinent stakeholders in the organization. These challenges motivated our decision to pursue a discount usability evaluation method. We argued that HE could be a viable approach, but that there is a need for a new set of usability heuristics more applicable to the typically collaborative and asynchronous workflows of these systems. However, any new usability heuristics must first be tested and validated before they can be adopted. We described our methodology for validating a new set of ITSM usability heuristics.

Our discussion reveals the challenges of validating usability evaluation methods, and in our case, a new set of usability heuristics. These challenges include recruiting participants with the knowledge of HE, quantifying the participants' HCI and computer security expertise, generating a list of known usability problems, synthesizing the result of evaluation by each evaluator, aggregating individual evaluator's results, assigning severity ratings to the identified problems, and performing HE of the complex identity management system in the limited time of a lab study and with the limited domain expertise of participants. Our discussion might be helpful to usable security researchers planning to develop and validate a new set of heuristics for the usability evaluation of security mechanisms.

Currently, we are preparing for participant recruitment. We will provide reflections on the outcomes of our methodological choices in the workshop.

7. REFERENCES

[1] K. Baker, S. Greenberg, and C. Gutwin. Heuristic evaluation of groupware based on the mechanics of collaboration. *Lecture Notes in Computer Science*, pages 123–140, 2001.

[2] D. Botta, R. Werlinger, A. Gagné, K. Beznosov, L. Iverson, S. Fels, and B. Fisher. Towards understanding IT security professionals and their tools. In *Proc. of Symp. On Usable Privacy and Security (SOUPS)*, pages 100–111, Pittsburgh, PA, July 18-20 2007.

[3] G. Fitzpatrick, T. Mansfield, and S. Kaplan. Locales framework: Exploring foundations for collaboration support. In *CHI'96 Sixth Australian Conference on Computer-Human Interaction*, Hamilton, New Zealand, November 24–27, 34–41 1996.

[4] S. Greenberg, G. Fitzpatrick, C. Gutwin, and S. Kaplan. Adapting the locales framework for heuristic evaluation of groupware. *Australian Journal of Information Systems*, 7(2):102–108, 2000.

[5] C. Gutwin and S. Greenberg. The mechanics of collaboration: developing low cost usabilityevaluation methods for shared workspaces. *IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proeedings*, pages 98–103, 2000.

[6] H. R. Hartson, T. S. Andre, and R. C. Williges. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 13(4):373–410, 2001.

[7] G. Iachello. *Privacy and proportionality*. PhD thesis, Georgia Institute of Technology, 2006.

[8] P. Jaferian, D. Botta, K. Hawkey, and K. Beznosov. A case study of enterprise identity management system adoption in an insurance organization. In *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, pages 46–55. ACM, 2009.

[9] P. Jaferian, D. Botta, F. Raja, K. Hawkey, and K. Beznosov. Guidelines for Designing IT Security Management Tools. In *CHIMIT '08: Proceedings of the 2008 symposium on Computer Human Interaction for the Management of Information Technology*, pages 7:1–7:10. ACM, 2008.

[10] R. Jeffries, J. R. Miller, C. Wharton, and K. Uyeda. User interface evaluation in the real world: a comparison of four techniques. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 119–124, New York, NY, USA, 1991. ACM.

[11] V. Kaptelinin and B. Nardi. *Acting with technology: Activity theory and interaction design*. MIT Press, 2006.

[12] A. G. Kotulic and J. G. Clark. Why there aren't more information security research studies. *Information & Management*, 41(5):597–607, 2004.

[13] J. Mankoff, A. K. Dey, G. Hsieh, J. Kientz, S. Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *Proc. CHI '03*, pages 169–176, New York, NY, USA, 2003. ACM.

[14] M. J. Muller and A. McClard. Validating an extension to participatory heuristic evaluation: quality of work and quality of work life. In *CHI '95: Conference companion on Human factors in computing systems*, pages 115–116, New York, NY, USA, 1995. ACM.

[15] D. C. Neale, J. M. Carroll, and M. B. Rosson. Evaluating computer–supported cooperative work: models and frameworks. In *CSCW '04*, pages 112–121. ACM Press, 2004.

[16] J. Nielsen. Finding usability problems through heuristic evaluation. In *Proc. CHI '92*, pages 373–380, New York, NY, USA, 1992. ACM.

[17] J. Nielsen. Severity ratings for usability problems. http://www.useit.com/papers/heuristic/severityrating.html, 2005.

[18] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM.

[19] D. A. Norman. *Cognitive Engineering*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[20] D. Pinelle, N. Wong, and T. Stach. Heuristic evaluation for games: usability principles for video game design. In *Proc. CHI '08*, pages 1453–1462, New York, NY, USA, 2008. ACM.

[21] M. B. Rosson and J. M. Carroll. *Usability engineering: scenario-based development of human-computer interaction*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[22] A. Sutcliffe and B. Gault. Heuristic evaluation of virtual reality applications. *Interacting with Computers*, 16(4):831 – 849, 2004. Human Computer Interaction in Latin America.

[23] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey. A survey of user-centered design practice. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 471–478, New York, NY, USA, 2002. ACM.

[24] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov. Preparation, detection, and analysis: the diagnostic work of IT security incident response. *Journal of Information Management & Computer Security*, 18(1):26–42, January 2010.

[25] J. Zhang, T. R. Johnson, V. L. Patel, D. L. Paige, and T. Kubose. Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics*, 36(1-2):23 – 30, 2003. Patient Safety.

[26] A. T. Zhou, J. Blustein, and N. Zincir-Heywood. Improving intrusion detection systems through heuristic evaluation. In *in IEEE Canadian Conf. on Electrical B. and Computer Engineering (CCECE)*, pages 1641 – 1644, 2004.