Heuristics for Evaluating IT Security Management Tools

ABSTRACT

The usability of IT security management (ITSM) tools is hard to evaluate by regular methods, making heuristic evaluation attractive. In this paper, we explore how domain specific heuristics are created by examining prior research in the area of heuristic and guideline creation. We then describe our approach of creating usability heuristics for ITSM tools, which is based on guidelines for ITSM tools that are interpreted and abstracted with activity theory. With a between-subjects study, we compared the employment of the ITSM and Nielsen's heuristics for evaluation of a commercial identity management system. Participants who used the ITSM set found more problems categorized as severe than those who used Nielsen's. We analyzed several aspects of our heuristics including the performance of individual participants using the heuristic, the performance of individual heuristics, the similarity of our heuristics to Nielsen's, and the participants' opinion about the use of heuristics for evaluation of IT security tools. We then discuss the implications of our results on the use of ITSM and Nielsen's heuristics for usability evaluation of ITSM tools.

CONTENTS

1. INTRODUCTION

Information technology security management (ITSM) tools serve several purposes including protection of network and data, detection of threats and vulnerabilities, and management of users and their accesses (Beal, 2005). Recent research (Botta et al., 2007; Goodall, Lutters, & Komlodi, 2004; Werlinger, Hawkey, Botta, & Beznosov, 2009) has highlighted the importance of collaboration and information sharing support between various stakeholders in IT security tools. Werlinger et al. (2009) identified nine security activities that require collaborative interactions and developed a model of the complexity of their interactions. This complexity arises from organizational attributes (e.g., distribution of IT management); the need for security practitioners (SPs) to interact with multiple stakeholders; and their need to engage in multiple security related activities. Each of these activities may require different explicit or tacit knowledge and kinds of information to be conveyed.

Usability is an important quality for ITSM tools (Chiasson, Van Oorschot, & Biddle, 2007), but evaluating the usability of specific ITSM tools is challenging. Laboratory experiments may have little validity due to the complexity of real-world security problems and the need to situate a specific tool within a larger context (Neale, Carroll, & Rosson, 2004). However, it is difficult to recruit SPs for simple interviews, let alone field observations (Botta et al., 2007; Kotulic & Clark, 2004). Direct observation of tool use can be time consuming as much security work is spontaneous (e.g., security incident response), or occurs over many months (e.g., deploying an identity management system Jaferian, Botta, Hawkey, & Beznosov, 2009). As ITSM tool use is intrinsically cooperative, its study inherits the difficulties of studying cooperation (Neale et al., 2004). Therefore, heuristic evaluation of ITSM tools could be a viable and low cost component of tool usability evaluation.

The existing sets of heuristics did not capture many of the challenges specific to the ITSM domain. As we report in this paper, we needed to explore how domain specific heuristics have been created in the past, develop a methodological approach for creating them, and apply the method to the creation and evaluation of a new set of heuristics for usability evaluation of ITSM tools. Our results suggest that using a combination of a bottom-up approach by deriving guidelines from literature and interview data, and a top-down approach by abstracting the guidelines using activity theory (Kaptelinin & Nardi, 2006) can lead to a set of heuristics that can find problems in IT security tools. In this paper, after presenting the set of heuristics we created, we report on the empirical evaluation of our heuristics in which we compared their usage to Nielsen's heuristics (Nielsen & Molich, 1990). We conducted a between-subjects study with 28 participants and examined different aspects of evaluation when deploying the two sets of heuristics. Our results suggest that the number of major problems that are found using the ITSM heuristics is higher than the number of problems that are found using Nielsen's. Furthermore, while Nielsen showed that about five evaluators are able to find about two thirds of the problems, in our evaluation of the IdM system, five evaluators only found about half of the problems found by 14 evaluators; we observed few overlaps between problems identified by individual participants using either Nielsen's or the ITSM

heuristics. Based on the result of evaluation and participants' feedback, we discuss how ITSM and Nielsen's heuristics can be employed for usability evaluation of ITSM tools.

2. BACKGROUND AND RELATED WORK

In this section, we provide a brief overview of the definition and scope of ITSM technologies before reviewing the prior research on the socio-technical aspects of ITSM. We then provide background on heuristic evaluation method, and domain specific heuristics. Finally, we provide a brief overview of activity theory.

Background: ITSM products are components in the design, development, and maintenance of a secure information technology infrastructure (Grance, Stevens, & Myers, 2003). Penn (2009) provides a taxonomy that classifies ITSM products into eight categories: (1) content security (e.g., email or web security), (2) endpoint security (e.g., personal firewalls or antiviruses), (3) identity and access management, (4) application security (e.g., code testing or web application firewalls), (5) network security (e.g., firewalls), (6) data security (e.g., file encryption), (7) security operations (e.g., log management, forensics, or security configuration management), and (8) risk and compliance management (e.g., security assessment, training). These technologies are used directly by multiple stakeholders and may also affect them indirectly. For example, the primary users of content security, network security, data security, and security operations tools are SPs; application security tools are used by developers; risk and compliance management tools are used by auditors; and end-point security and identity and access management tools are used by variety of end-users in their day-to-day activities.

Socio-technical aspects of ITSM: ITSM is one aspect of IT management (Botta et al., 2007), and the results of socio-technical studies of IT are applicable to ITSM. IBM researchers have done extensive research on the nature of IT administration work and its challenges (Barrett, Prabaker, & Takayama, 2004) and have identified recommendations for tool design (Barrett, Maglio, Kandogan, & Bailey, 2005; Haber & Bailey, 2007). As a part of their research, they focused on ITSM tools and practices (Kandogan & Haber, 2005). They show that SPs work in a collaborative environment, communicate with people with different backgrounds, work with large data sets, and interact with complex systems. Researchers in the HOT Admin project at UBC performed an extensive study of ITSM based on interviews, observation, and surveys. They found that ITSM activities are constantly changing, complex, challenging, and collaborative (Botta et al., 2007; Werlinger, Hawkey, & Beznosov, 2009; Werlinger, Hawkey, Botta et al., 2009). They further studied these aspects of ITSM in diagnostic work (Werlinger, Hawkey, Muldner, & Beznosov, 2009) and deployment and on-going usage of an intrusion detection system (Werlinger, Hawkey, Muldner, Jaferian, & Beznosov, 2008). Other research includes that of: Siegel, Reid, & Dray (2006), who show the importance of organizational and human factors in ITSM; Kraemer & Carayon (2007), who studied the factors that result in human errors in IT security; and Goodall et al. (2004), who studied intrusion detection task in ITSM and showed its complex and collaborative nature. In summary, the prior research found that ITSM activities involve technical complexity, collaboration, and stakeholder diversity.

Heuristic Evaluation: Heuristic evaluation (HE) is a usability evaluation method based on a set of usability principles called heuristics. They are called "heuristics" because they are more in the nature of rules of thumb than specific usability guidelines (Nielsen, 2005). An evaluator inspects a user interface and identifies usability problems and their severity based on heuristics and his judgment of the interface. Heuristic evaluation is the most popular informal usability evaluation technique (Vredenburg, Mao, Smith, & Carey, 2002) and it can lead to finding more serious usability problems compared to usability testing, guidelines and cognitive walkthrough (Jeffries, Miller, Wharton, & Uyeda, 1991). Nielsen's theoretically grounded and extensively tested heuristics (Nielsen & Molich, 1990) are the most widely accepted heuristics. They were developed based on existing HCI guidelines, are consistent with Norman's theory of action (Norman & Draper, 1986), and focus on the dialogue between a single user and the physical world. Nielsen's heuristics have been modified or extended to create domain specific heuristics (e.g., ambient displays by Mankoff et al., 2003, virtual reality by Sutcliffe & Gault, 2004, medical devices by Zhang, Johnson, Patel, Paige, & Kubose, 2003, intelligent tutoring systems by Muller, & McClard, 1995, and intrusion detection systems Zhou, Blustein, & Zincir-Heywood, 2004). Furthermore, heuristics have been created without using Nielsen's heuristics (e.g. computer games by Pinelle, Wong, & Stach, 2008, groupware by Greenberg, Fitzpatrick, Gutwin, & Kaplan, 2000, shared visual work surfaces for distance-separated groups by Baker, Greenberg, & Gutwin, 2002, Large Screen Information Exhibits [LSIE] by Somervell, 2004, Ubiquitous systems by Scholtz & Consolvo, 2004). We only have found one instance of applying heuristic evaluation on an ITSM tool. Zhou et al. (Zhou et al., 2004) developed six heuristics based on Nielsen's heuristics for intrusion detection systems. The authors described that they developed the heuristics based on surveys and interviews, but they did not provide any detail of their methodology. Four of their six heuristics are identical to Nielsen's heuristics and the rest are extensions to Nielsen's.

Activity Theory: Activity theory was developed by Leont'ev (1974) as a general psychological theory, and was later proposed as a potential framework for HCI research (Kuutti, 1995). Activity theory moves the unit of analysis beyond user actions, with "*Human Activity*" as the unit of analysis. Kaptelinin & Nardi (2006, pp. 66-72) suggested five basic principles of activity theory: (1) *Consciousness and object-orientedness:* all human activities are performed by a conscious actor towards an object. (2) *Hierarchical structure:* activities have three levels: activity, actions, and operations. (3) Activities involve *internalization* and *externalization*. (4) *Mediation:* activities are performed by using and transforming artifacts. (5) *Development:* activities evolve and develop over time.

Engeström (1999) proposed a formulation of activity theory to explicate the components and internal relations of an activity system. His model suggests that the components of activity can be classified into subject, object, mediating artifacts, rules, community, and division of labor. He also suggested five activity theory principles including: (1) activity as a unit of analysis, (2) multi-voicedness, (3) historicity, (4) contradictions, and (5) transformation (Engeström, 2001). The sets of principles suggested by Kaptelinin & Nardi (2006) and Engeström (2001) are not mutually exclusive or contradictory; but they do provide different perspectives on activity theory.

3. PROPOSED ITSM HEURISTICS

In this section, we describe how usability heuristics can be created, classify prior heuristic creation literature according to its methodology, and discuss the benefits and drawback of each approach. Then we describe the method of heuristic creation we employed, followed by the list of proposed heuristics.

3.1. Methods for creating heuristics

Guidance to designers can emerge in three forms: "(1) high level theories and models, (2) middle-level principles, and (3) specific and practical guidelines" (Shneiderman, 1997). Principles represent the theory with an eye to what should be practiced, and the guidelines take the principles one step further toward their application (Te'eni, Carey, & Zhang, 2007). Nielsen (1994) defines heuristics as "general rules that seem to describe common properties of usable interfaces." Considering these definitions, we classify heuristics as middle-level principles. In addition, a list of heuristics should be short (about seven to ten) and easy to teach. Heuristics should also be open to interpretation, so that multiple evaluators can use them to find diverse problems (Nielsen & Molich, 1990).

Two approaches can be used to develop domain specific principles: (1) Bottom-up: qualitative data is collected and analyzed to understand the characteristics of the domain, and principles are created using real-world data. (2) Top-down: high-level expert knowledge, and/or theory is used to derive specific recommendations for the target domain. Figure 1 provides comparison of the major literature on heuristic creation.

In a bottom-up approach, two types of qualitative data were used in the literature to synthesize heuristics. First, researchers studied positive and negative aspects of specific systems in the target domain by either employing claims analysis[1] (Somervell, 2004) or by analyzing the content of product reviews (Pinelle et al., 2008). Second, guidelines from literature were used to synthesize heuristics (Nielsen, 1994). The advantage of the bottom-up approach is that the heuristics are grounded in real-world data, and reflect real problems with the tools in the target domain. The disadvantage is that the produced heuristics are limited by the scope and richness of the qualitative data and the interpretation of that data by the researchers.

In a top-down approach, expert knowledge is used to derive heuristics from high-level theories or principles. Heuristics can be derived from a substantive theory,[2] a formal theory,[3] or existing heuristics. For example, the mechanics of collaboration framework (Baker et al., 2002) (a substantive groupware theory), or the general HCI literature (formal theories) (Scholtz & Consolvo, 2004) have been used to derive heuristics. Also

[1] See (Carroll & Rosson, 1992) for details of the claims analysis method

[2] A theoretical interpretation or explanation of a delimited problem in a particular area (Charmaz, 2006).

[3] A theoretical rendering of a generic issue or process that cuts across several substantive areas of study (Charmaz, 2006).

expert knowledge can be used to modify Nielsen's heuristics for the target domain (Mankoff et al., 2003). The top-down approach relies on expert knowledge to modify a theory or an existing heuristics set, and customize it for usability evaluation of the domain specific systems. Therefore, the process of heuristic derivation is not systematic, and is prone to researcher bias.

Figure 1 - Comparison of the major heuristic creation literature. The "T" and "B" indicate top-down and bottom-up method of heuristic creation.

Author	Domain	Creation Method	Creation method details
(Nielsen, 1994)	General	BT	First, a bottom-up approach was used to gather 101 usability guidelines and principles from different sources. Then a top-down approach was used by performing factor analysis and expert review to combine and narrow the guidelines to heuristics.
(Mankoff et al., 2003)	Ambient displays	T	The heuristics were developed by changing Nielsen's heuristics based on prior experience of authors and by getting feedback from experts in designing ambient displays.
(Baker et al., 2002)	Shared visual workspaces	T	Mechanics of collaboration theoretical framework was used to derive heuristics. The exact process of derivation was not described.
(Greenberg et al., 2000)	Groupware	T	Locales framework concepts were re-cast as heuristics.
(Pinelle et al., 2008)	Computer games	B	108 game reviews from Gamespot.com were analyzed using qualitative analysis techniques, problems were extracted from reviews, categorized, and then heuristics were derived from the categories.
(Scholtz & Consolvo, 2004)	Ubiquitous computing	T	Expert knowledge and general HCI literature were used to derive a framework for evaluation of ubiquitous systems.
(Somervell, 2004)	Large Screen Information Exhibits (LSIE)	BT	A bottom-up approach was used by performing claims analysis to analyze design decisions of five major LSIE systems. A top-down approach was used to combine similar claims and derive high-level heuristics. Each claim was classified according to its impact on three critical parameters for design of notification systems. In addition, scenario-based design categories (Carroll & Rosson, 1992) were used to further classifying the claims. The heuristics were created using the categories.

To address the above limitations, a more rigorous process can be used by combining both bottom-up and top-down heuristic creation. The process can be started bottom-up

from empirical data by using a qualitative data analysis method such as Grounded Theory (Charmaz, 2006). This process will result in design guidance grounded in empirical data. Then a top-down approach can be used to justify, support, and combine the identified design guidance into heuristics. We advocate that the complementary top-down approach be rooted in a theory rather than expert knowledge, to leverage a more formal and less ad-hoc process. The use of theory reduces researcher bias in interpreting qualitative data, and abstracting and refining the findings into heuristics. In addition, a link between theory and heuristics will provide insight into the theory behind the heuristics, and help in communicating them to others. In the literature, Somervell (2004) adopted this approach by combining both bottom-up and top-down approaches systematically. We used a similar approach to create ITSM heuristics that suits best to the ITSM context. Unlike large screen information exhibits systems (the domain of interest in Somervell, 2004), which are limited in number, there are a vast number of ITSM tools. Because analyzing those tools was not a viable approach, we used literature and interviews as a data source, and grounded theory as the analysis method. For our top-down approach, we used formal theory as there was no substantive theory for ITSM. We further discuss our creation process in the next section.

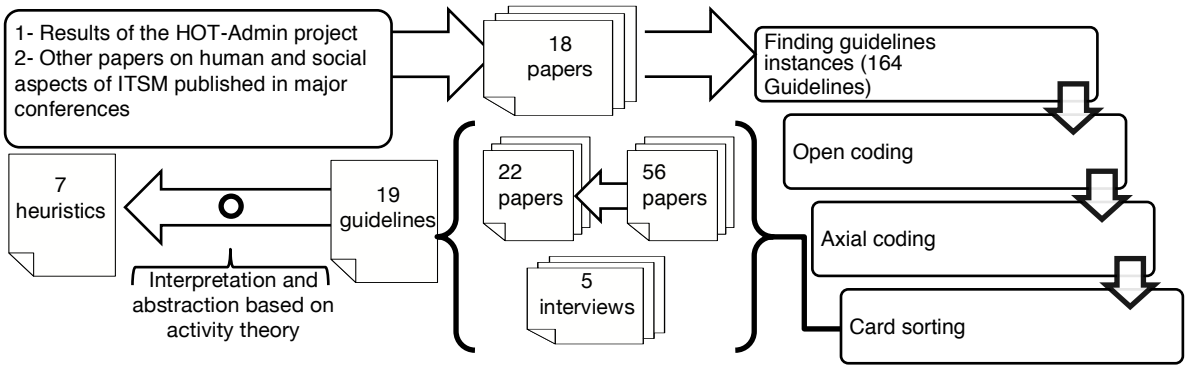
3.2. Our methodology for creating ITSM heuristics

We used a combination of top-down and bottom-up approaches to develop ITSM heuristics. These approaches consisted creating guidelines from the literature, and interpreting and explaining guidelines using the theoretical lens of activity theory.

Guideline Creation

We started with a bottom-up approach by understanding the characteristics of IT security management (ITSM) tools that help SPs perform activities more effectively and efficiently. We collected data from two sources: related work and interviews performed in the HOT-Admin project (see Section 2). We first analyzed a set of primary publications that included HOT-Admin publications (4 papers) and other publications about ITSM tools (14 papers). We identified 164 explicit guidelines for building ITSM tools, recommendations for improvement, design decisions in a particular tool that have positive impact on usability, and pros and cons of various tools. We categorized these using Grounded Theory (Charmaz, 2006). First, we performed open coding using codes that emerged from the data, followed by axial coding to combine conceptually similar open codes. Meanwhile, following the theoretical sampling technique, we broadened our sources of data by reviewing the papers published in well-known conferences related to the topic, performing keyword searches, and mining the references from our original set of 18 papers. Our goal in this stage was to saturate the identified themes in the first round of analysis, refine the identified guidelines and find a better relationship between them. The result of this search was a list of 56 papers. We then reviewed the papers and found another 22 papers that could contribute to our guidelines. We also analyzed five semi-structured interviews with SPs to find support for our guidelines and illustrative examples. This process resulted in 19 guidelines (Jaferian, Botta, Raja, Hawkey, & Beznosov, 2008) for ITSM tools.

Figure 2 - Overview of the process of developing ITSM heuristics



The identified guidelines were based on collected data and were specific and limited to the data we analyzed. Therefore, we used a top-down approach to look at the guidelines through a theoretical lens provided by a theory that can describe the characteristics of ITSM domain. Using theory leveraged our interpretation of data and added another level of validation to our findings.

Choosing a theory

To choose a theory that can be applied to the analyzed data, we searched for specific IT security management theories, but to our knowledge, no such theory existed. We then sought general HCI theories that have been applied to contexts with technological, social, and organizational complexity. Information processing psychology, which has been extensively used as the dominant theoretical foundations for HCI (Kaptelinin et al., 2003), was the first candidate; but it was rejected quickly as it doesn't take into account the context in which users' actions are situated. Consequently, we reviewed three widely used post-cognitivist theories: Activity Theory (Kaptelinin & Nardi, 2006), Distributed Cognition (DCog) (Hollan, Hutchins, & Kirsh, 2000), and Phenomenology (Dourish, 2001). All of these theories can be used as foundations for understanding the use of technology in a social and organizational context (Kaptelinin et al., 2003). As we describe next, we found the activity theory perspective to be the best fit for the ITSM domain. From the phenomenology perspective, each instance of human activity is unique and different from other instances. Phenomenology argues against abstracting human activities and finding commonalities between various instances. Such a perspective is advantageous in describing a specific human activity, but it has limited ability in "higher-order scientific tasks where some abstraction is necessary" (Nardi, 1995) (e.g., developing abstract heuristics). From the DCog perspective, a social system can be modeled as a network of people, and artifacts, all of which are capable of cognition and transformation of information. Two main assumptions of DCog are the symmetry of human and tools, and smooth functioning of the system. While such viewpoints can be advantageous in contexts where smooth functioning and limited creativity is expected (e.g., the call center in an organization), it is limited in the ITSM domain, which involves unknown situations, breakdowns, creative use of artifacts, judgment and reflection, contradictory goals, and learning. Activity theory principles fit well with ITSM characteristics. For example, principles such as contradictions can describe breakdowns,

mediation can describe creative use of artifacts, and internalization and externalization can describe judgment and reflection. Additionally, the prior use of activity theory for modeling certain aspects of IT security shows the fit between the theory and the domain (Zager, 2002).

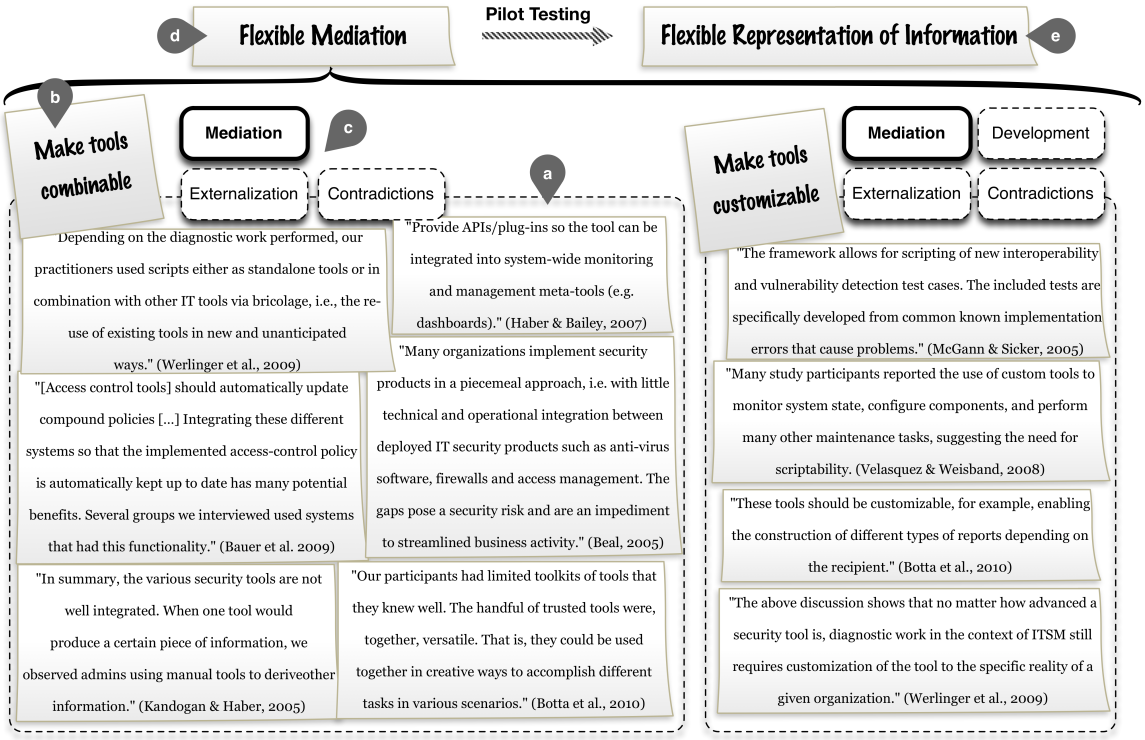
Applying the theory

To use activity theory to abstract and combine the guidelines, we analyzed the guidelines using the theoretical lens provided by activity theory. We used ten activity theory principles from two well-known sources (Engeström, 2001; Kaptelinin & Nardi, 2006) (see Section 2 for the list of principles), and cross-tabulated them with the guidelines in a matrix. The matrix allowed us to summarize how theory explains each guideline. We then chose one of the principles as the main explaining principle and the rest as supporting principles before combining guidelines explained by the same main principle. This led to 13 guidelines combined under six categories. The remaining guidelines could not be classified under a single category as the guidelines had different components explained by different main principles. These guidelines were broken down and classified under four of the previous categories and a new category. We then tried to convert each category into a heuristic. When categories are crafted as heuristics, they should be concise, easy to understand, and open to interpretation. We used an iterative approach of multiple piloting sessions and getting feedback from peers. We illustrate an example of our heuristic synthesis process in Figure 3. In this example, we generated the guideline "make tools combinable" (Figure 3b) using six sources (Figure 3a), and "make tools customizable" using four sources. Then activity theory could explain "make tools combinable" by the creation (*externalization*) of mediating artifacts (*mediation*) to address unexpected conditions (*contradictions*), and "make tools customizable" by the customization (*externalization*) of the mediating artifacts (*mediation*) as users' knowledge or activity evolves over time (*development*) and the tool is no longer best suited to the activity (contradictions). We then chose "mediation" as the main principle. We then combined these two guidelines into "flexible mediation" (Figure 3d). We later reworded the heuristic to "flexible representation of information" based on the feedback from our pilot testing participants (Figure 3e).

3.3. Proposed ITSM Heuristics

In this section, we present seven heuristics for the usability evaluation of ITSM tools. We provide the title and the description of each heuristic and then empirical support for it from the literature. To illustrate the importance of the heuristic with real-world examples, we include interview snippets from seven interviews with SPs conducted as a part of our ongoing research projects (participants are identified by codes from SP1 to SP7). We then provide theoretical support for the heuristics.

Figure 3 - An example of heuristic synthesis process: we used a bottom-up approach by analyzing literature on ITSM tools (a) to create guidelines (b), and a top-down approach (c) using activity theory to extract preliminary heuristics (d) which later reworded to final heuristics (e).



Heuristic #1 - Visibility of activity status: "*Provide users with awareness of the status of the activity distributed over time and space. The status may include the other users involved in the activity, their actions, and distribution of work between them; rules that govern the activity; tools, information, and materials used in the activity; and progress toward the activity objective. Provide communication channels for transferring the status of the activity. While providing awareness is crucial, limit the awareness to only what the user needs to know to complete his actions.*"

Discussion: In ITSM, the actions that form an activity are distributed across time and space. These actions are performed in an organizational context with certain norms and rules. Plans are created and modified by different stakeholders, and roles are assigned dynamically to address unknown conditions. Prior ITSM research shows the importance of providing awareness of organizational constraints (Zager, 2002), communication channels (Werlinger, Hawkey, Botta et al., 2009), methods for sending cues to different stakeholders to inform them about when and how to act (Botta, Muldner, Hawkey, & Beznosov, 2011), awareness of what other stakeholders perform in the system, sharing the system state between different SPs, and grounding new participants in ITSM activities (Haber & Bailey, 2007). To illustrate this, SP2 said his team receives daily reports on employees' malicious actions and described the importance of awareness in preventing insiders' malicious behavior: "*We can lock down - we use policies and things*

like this to keep people from doing what they shouldn't be doing. We lock down firewalls so that they cannot do what they shouldn't be doing. Because we are running reports and you know who is doing stuff."

Looking at the problem through the lens of activity theory, tools can provide awareness about the components of activity including artifacts, community, and rules. Carroll, Neale, Isenhour, Rosson, & McCrickard (2003) described three types of awareness: (1) social awareness, the understanding of current social context in an activity (e.g., rules, artifacts); (2) action awareness, the understanding of actions of collaborators on shared resources; and (3) activity awareness, the understanding of how shared plans are created and modified, how things are evaluated, and how roles are assigned. As ITSM tools deal with sensitive information, visibility should be in the form of social translucence rather than social transparency (Erickson & Kellogg, 2000).

This heuristic is different from Nielsen's "Visibility of System Status", which focuses on immediate status of the system to help users select appropriate actions and evaluate the outcome of their actions. The ITSM heuristic includes aspects of the system status that might not be available locally and immediately.

Heuristic #2 - History of actions and changes on artifacts: "*Allow capturing of the history of actions and changes on tools or other artifacts such as policies, logs, and communications between users. Provide a means for searching and analyzing historical information.*"

Discussion: Accountability and reflecting on work are important aspects of ITSM (Gagné, Muldner, & Beznosov, 2008; Velasquez & Durcikova, 2008). As ITSM involves creative work to address unknown conditions, providing usage histories supports creativity, learning, and quality improvement (Shneiderman, 2000). Audits, which aid in reflecting on work, are mandated in IT security as a part of regulatory legislations such as the Sarbanes-Oxley Act (Sarbanes, 2002). Prior ITSM research (Gagné et al., 2008) showed SPs archive logs and keep a history of communications for audit and accountability. To illustrate, SP7 described that healthcare organizations allow physicians to openly access patient data but they archive and audit every access attempt: "*I let you do it, but I audit a crap out of the system. So if somebody complains or someone reports that I saw somebody accessed something and I don't think it is appropriate then you've got a really robust audit records.*" Histories can also be used to understand other stakeholders' actions. For example, sometimes access control policies are changed by multiple SPs; keeping track of changes will help other SPs maintain a working knowledge of the implemented policy (Bauer, Cranor, Reeder, Reiter, & Vaniea, 2009). Finally, historical information can be used for trend analysis, learning about the network, and evaluating the outcome of actions that span time and space (Velasquez & Durcikova, 2008).

From the theoretical perspective, artifacts in an activity carry a history with them. Awareness of this history influences the way those artifacts are used. Hollan et al. (2000) studied experts working in complex environments and found that usage histories are incorporated in cognitively important processes. Historical information could be in the

form of the usage histories of the user himself or of other users of the system. Usage histories can be employed to reflect on work, and to get feedback from peers (Shneiderman, 2000).

Heuristic #3 - Flexible representation of information: "*Allow changing the representation of information to suit the target audience and their current task. Support flexible reports. Allow tools to change the representation of their input/output for flexible combination with other tools*."

Discussion: SPs often use inferential analysis and pattern recognition to develop policies, audit security, or troubleshoot security incidents (Botta et al., 2007). For example, they need to look for certain patterns in network traffic to detect an anomaly; or they need to analyze users' access to different resources in order to build an effective set of role-based access control (RBAC) roles. To perform these activities, SPs often use their tools in creative ways that were not anticipated by tool developers; or alternatively, they combine their tools. For example, SP2 described their reason for building custom tools: "*Sometimes, I do need to custom craft something or I need to automate something. Or I need to do something maybe that the tool doesn't do."* Botta et al. (2007) identified SP's practice of bricolage (i.e., combining different tools in new ways) to address complex problems and argued that ITSM tools should survive in the arena of bricolage. Haber & Bailey (2007) and Beal (2005) also highlighted the need for better integration between ITSM tools. SP3 described the problem with correlating data from 17 vulnerability analysis tools: "*We are really, really having a problem at correlating output from all these tools. At the beginning they were using three or four, it was easy to manually correlate, but when they started hitting six, seven, eight, plus, it was very difficult to correlate because the outputs are all different*". Therefore, they wrote a homegrown solution to convert and import all the data into a database, and cross-reference the findings of different tools.

Tools should also be flexible in representing information. This allows users to use a representation that best suites the task, and the background and expertise of the user. SP3 described why they preferred vulnerability analysis tools with command line interface when they built the homegrown solution: "*We actually use the command line interface route and we try to keep it as simple as possible because we were putting another layer on top of it, we couldn't go into the graphical one*." SP3 also clarified why some users prefer GUI based tools: "*... sometimes clients want graphical stuff. Especially if they are not 100% techie, it's easier*." In prior ITSM research, the need for flexible interaction methods (e.g., Command Line Interface and Graphical User Interface) (Botta et al., 2007; Thompson, Rantanen, Yurcik, & Bailey, 2007), flexible reporting (Botta et al., 2007; Velasquez & Weisband, 2008; Werlinger, Hawkey, Botta et al., 2009), visualization techniques (Dourish & Redmiles, 2002), and multiple views (Haber & Bailey, 2007) are highlighted.

From the activity theory perspective, ITSM tools are mediating artifacts. Their mediation role can be between users (e.g., wiki, or other communication channels), between users and other tools (e.g., visualization of network traffic), or between two other tools (e.g., a script) (Botta et al., 2011; Maglio, Kandogan, & Haber, 2003).

Therefore, a tool should be able to process an input from a user or another tool, and to provide an output that is understandable to the user or the tool that receives the output. This concept was further explained by Norman (1991). He described that artifacts have two types of representation: the internal representation that is not accessible by the outside world, and the surface representation that is their interface to the world. Providing a flexible surface representation is particularly important in ITSM as tools are creatively combined by SPs and their output is used by users with different knowledge and background (Botta et al., 2011). Previous activity theory research (Kaptelinin & Nardi, 2006; Rabardel & Bourmaud, 2003) also argued in favor of highly customizable and open tools, when users combine and adapt different tools to build instruments for unexpected and unknown conditions.

Heuristic #4 - Rules and constraints: "*Promote rules and constraints on ITSM activities, but provide freedom to choose different paths that respect the constraints. Constraints can be enforced in multiple layers. For example, a tool could constrain the possible actions based on the task, the chosen strategy for performing the task (e.g., the order of performing actions), the social and organizational structure (e.g., number of stakeholders involved in the task, policies, standards), and the competency of the user.*"

Discussion: ITSM tools are used in organizational context with rules, norms, and constraints. Violating these constraints will result in sub-optimal situations; therefore, tools can help enforce such constraints. Botta et al. (2011) show that enforcing norms by ITSM tools in the form of procedures for notification and support for particular templates and standards can prevent communication and collaboration breakdowns. Werlinger, Hawkey, & Beznosov (2009) argue that ITSM tools can promote security culture in organizations and address the lack of training by enforcing policies. SP4 further clarified the importance of leveraging policies using tools: "*So really IDS [Intrusion Detection System] should really be something that supports you in your ability to leverage policy so really security I think is 90% policy and the rest of it is tools. [...] You've got to have the policy structure behind you and the tools to find out if your policies are being respected.*"

From the activity theory perspective, there are rules and norms that govern every activity. Promoting rules and norms by tools can lead to awareness and internalization of those norms by stakeholders (Kaptelinin & Nardi, 2006). Vicente (2000) points out the importance of enforcing rules and constraints by tools, while allowing users to flexibly explore the possible action space. This helps users be aware of constraints, and gives them flexibility to adapt to unexpected situations. Vicente argues that constraints can be expressed at five different levels: work domain, control tasks, strategies, social-organizational, and worker competencies. Rules in ITSM can include security and privacy policies or standards, organizational constraints, and organizational culture.

Heuristic #5 - Planning and dividing work between users: "*Facilitate dividing work between the users involved in an activity. For routine and pre-determined tasks, allow incorporation of a workflow. For unknown conditions, allow generation of new work plans and incorporation of new users.*"

Discussion: The ITSM context requires quick responses to unknown conditions by stakeholders, who work with tight schedules in which ITSM has a low priority (Botta et al., 2007). Therefore, planning and dividing work between stakeholders is important (Werlinger, Hawkey, Botta et al., 2009). SPs often need to coordinate activities with multiple stakeholders involving other SPs, IT admins, managers, end-users, and external stakeholders. For example, to address a security incident, SPs often need to collect data from end-users; analyze the incident; coordinate and collaborate with IT specialists, who own the impacted sub-systems (e.g., database admins); communicate with managers to warn them about the risks associated with the incident and possible disruptions in service; and even collaborate with external SPs to solve the problem. In all of these cases, proper planning tools should be available to quickly involve stakeholders and divide work between them. SP1 described the importance of dividing work on the efficiency of the security group: "*First and foremost, explicit definition of what you do and do not do. Everyone is capable of doing more than they can do on a day-to-day basis. However, if you haven't established clear lines of demarcation from what is your responsibility and what is not, particularly security people have the need to save the world and so they tend to do everything and therefore they burn out*."

Activity theory points to the division of labor as an important aspect of activity. Furthermore, division of labor should take into account constraints at the social organizational level, as well as possible methods for generating plans and collaborating considering those constraints.

Heuristic #6 - Capturing, sharing, and discovery of knowledge: "*Allow users to capture and store their knowledge. This could be explicit by means of generating documents, web-pages, scripts, and notes, or implicit by providing access to a history of their previous actions. Tools should then facilitate sharing such knowledge with other users. Furthermore, tools should facilitate discovery of the required knowledge source. The knowledge source can be an artifact (e.g., document, web-page, script) or a person who possesses the knowledge. Provide means of communicating with the person who possesses the knowledge*."

Discussion: SPs rely heavily on knowledge to perform their tasks (Botta et al., 2011). For example, to implement security access controls, a SP needs to know about the activities that a stakeholder performs, and the resources required for performing those activities. When asked how one can know the people that should be contacted in order to grant access to any of the 600 available applications in the organization, SP6 responded: "*So our access procedures state that every application that has any level of criticality is supposed to have a published knowledge-base document in our service desk. [It] defines what the application is, who owns it, who is the technical owner*." Therefore, SPs need to discover and use the knowledge of other stakeholders whether they are inside or outside of the organization. Prior research shows the importance of managing knowledge (Botta et al., 2011; Kesh & Ratnasingam, 2007) and suggests policy specification as a method to transfer knowledge (Werlinger, Hawkey, Botta et al., 2009). Rogers (Rogers, 1992) shows the importance of transmitting knowledge at the "window of opportunity" during troubleshooting in a network environment that involves multiple stakeholders and describes it as a challenging task.

From the theoretical perspective, the relationship between different actors in the activity is mediated by artifacts. As a result, in order to transfer knowledge, users should be able to externalize their knowledge as artifacts (Engeström, 1999). Facilities for identification and access to the required knowledge must then be provided. If externalization of knowledge is not feasible, a method for finding and starting collaboration with the person who possesses the knowledge should be provided. In this case, the communication channel is considered the mediating artifact.

Heuristic #7 - Verification of knowledge: "*For critical ITSM activities, tools should help SPs validate their knowledge about the actions required for performing the activity. Allow users to perform actions on a test environment and validate the results of these actions before applying them to the real system. Allow users to document the required actions in the form of a note or a script. This helps the users or their colleagues to review the required actions before applying them to the system.*"

Discussion: Many actions in ITSM are responses to new, unseen, and complex situations (Botta et al., 2007, 2011), and they are performed on artifacts critical to the organization. Moreover, the actions are distributed in time and space and the result of an action cannot be evaluated in real time. Therefore, errors in ITSM activities could lead to a security breach or disrupt services, which might impose high costs on the organization. For example, an error during deployment of a security patch might disrupt service and conflict with an organization's business activities (Botta et al., 2011). It can be hard to predict, or instantly determine, the outcome of the patching process, as other stakeholders need to confirm that the service is not impacted. To mitigate this, SPs employ "rehearsal and planning" (Haber & Bailey, 2007), by rehearsing the actions on a test system before performing it on a production system. SP5 described this activity during installation of an IdM system update: "*We [have] multiple environments where we can rehearse different [changes to the system]. Because the customer releases are so complex to do, you definitely want to try it a couple of times before you do it in production.*"

This practice can be clarified from a theoretical perspective. To find a solution to a problem, a SP consults several information sources and combines them into a single artifact (e.g., a plan, a guide document, a check list). This artifact acts as an external memory to the SP. The SP also internalizes knowledge from different sources, which might not be completely correct or applicable to the situation at hand. Therefore, this knowledge should be verified before applying it to the system. Activity theory explains that the process of revising knowledge involves externalization, revision, and internalization of the revisions (Engeström, 1999). In the context of ITSM, SPs perform externalization when they employ rehearsal. If something goes wrong in the rehearsal, SPs re-examine their interpretation of the external knowledge sources and go through the rehearsal and revision cycle again. After successful rehearsal, SPs can perform the rehearsed actions on the critical artifact.

4. EVALUATION METHODOLOGY

Background: While the ITSM heuristics are grounded in empirical data and supported by theory, the effectiveness of them must be evaluated. Heuristic creation

literature has tackled the problem of evaluation in four ways: (1) no evaluation or informal evaluation (Greenberg et al., 2000; Scholtz & Consolvo, 2004), where the effectiveness of heuristics have not been formally evaluated; (2) long-term evaluation by using and refining the heuristics in real-world projects (Nielsen, 1994); (3) controlled study of the effectiveness without using a control group (Baker et al., 2002; Pinelle et al., 2008); and (4) controlled comparative evaluation, where the effectiveness of heuristics is compared to existing heuristics (Mankoff et al., 2003; Somervell, 2004).

We chose the last method to evaluate the effectiveness of the ITSM heuristics. Similar to other domain specific heuristics, we did not use a long-term evaluation approach, as it requires longitudinal studies, and access to real-world usability projects. The controlled study without a control group does not allow recommending the use of the new heuristics over Nielsen's. A controlled comparative evaluation can show us if the new heuristics are more effective than Nielsen's for the ITSM domain, and if using them adds value to the heuristic evaluation.

The ultimate criteria for effectiveness of a set of heuristics (or a usability evaluation method in general) is finding real problems that user will encounter in real work contexts, which will have an impact on the usability (e.g., user performance, productivity, and/or satisfaction) (Hartson, Andre, & Williges, 2001). However, it is almost impossible to determine if each usability problem is real or not (Olson & Moran, 1998). The best we can do is to estimate the impact of the potential problem on the users who will use the system. Therefore, we evaluated our approach based on the following criteria proposed by Hartson et al. (2001): (1) thoroughness, the ability of the method to find most of the known problems (see Section 4.1 for the definition of known problems); (2) reliability, the ability of the method to find severe problems; and (3) validity, the ability of the method to find valid problems (4) effectiveness, ability of the method to find most of the known problems while it leads to few invalid problems, (5) cost-effectiveness, the cost of using method. While Hartson proposed six criteria for evaluation, we excluded downstream utility, which refers to the quality of the reported problems and how well they lead to effective redesign of the technology. According to Hartson, while evaluating the downstream utility of usability evaluation methods is commendable, it requires long-term studies of the impact of identified problems and it is out of the scope of this paper.

We also investigated the characteristics of heuristics evaluation using the ITSM heuristics including (1) the impact of the number of evaluators on the results; (2) performance of individual heuristics; (3) similarity between ITSM and Nielsen's heuristics; (4) the impact of participants' background on the reported problems; and (5) the usefulness, learnability, and applicability of heuristics.

To achieve the aforementioned goals, we performed a comparative study of the ITSM heuristics with Nielsen's heuristics. This between-subjects study divided participants into two groups: those that used Nielsen's heuristics (Nielsen condition, 14 participants) and those that used the ITSM heuristics (ITSM condition, 14 participants). For the Nielsen condition, we performed four in person (three, two, two, and one participants per session) and six remote evaluation sessions (one participant per session). For the ITSM condition,

we performed three in person (three, three, and one participants per session), and seven remote evaluation sessions (one participant per session).

Recruitment: The main inclusion criteria were a HCI background, and familiarity with heuristic evaluation. We sent emails to all graduate students in the CS and ECE departments of UBC. We also sent emails to the user experience mailing lists in Vancouver, to online HCI communities, and the CHI-Announcements mailing list, in order to reach participants with HCI experience; and to the HCI-Sec mailing list[4] to reach participants with a background in both security and usability. All participants were given a $50 honorarium for their participation.

Participants: In an attempt to balance the expertise of participants in each group, we screened them to assess their HCI and computer security background. In Figure 4, we present the participants' demographics. All but one participant had received formal HCI training, with the majority (17) receiving formal training on heuristic evaluation. The majority (19) had performed at least one heuristic evaluation in the past. The participants' average years of professional computer security experience in ITSM condition was about 3 times more than that of Nielsen condition. This difference was mainly due to the high variance in computer security background[5]. We further examine if the difference in computer security background could have an impact on the outcome of the evaluation in Section 5.2.

Figure 4 – Participants' demographics for each condition

Condition		**ITSM**	**Nielsen**	**Total**
Group Size (N)		14	14	28
Age	19-24, 25-30, 31-35, 36-45	2, 6, 4, 2	2, 7, 1, 4	4, 13, 5, 6
Gender	Female, Male	6, 8	6, 8	12, 16
Educational Level	Diploma, Undergrad, Graduate	1, 6, 7	0, 8, 6	1, 14, 13
Years of Experience (Average, Median)	HCI Research and professional	3.57, 2.5, 13.03	3.29, 2.0, 8.49	3.43, 2.0, 10.70
	Computer security research	0.64, 0, 1.93	0.50, 0, 2.57	0.57, 0, 12.18
	Computer security professional	1.0, 0, 4.46	0.32, 0, 0.52	0.66, 0, 2.52

As we described, the majority of participants had performed heuristic evaluation before. According to Nielsen (1994), it would be impossible to wipe the mind of evaluators of the additional usability knowledge they have, and in reality each evaluator would apply certain heuristics from the sets he or she was not suppose to use. Therefore, familiarity with Nielsen's heuristics would be an advantage for participants in ITSM group. We deliberately recruited participants with a heuristic evaluation background, and made the trade-off between controlling differences in the heuristic evaluation background and ecological validity of the study. Rather than controlling the knowledge of evaluators in the ITSM group by recruiting participants without prior exposure to Nielsen's

[4] HCI-Sec is a mailing list for those who do research on usability of security technologies.

[5] There was one outlier with 8 years of professional computer security experience in the ITSM condition. Removing the outlier changes the average years of professional computer security experience to 0.46, and variance to 0.44.

heuristics, we recruited participants who were representative of those who will use ITSM heuristics in the real world.

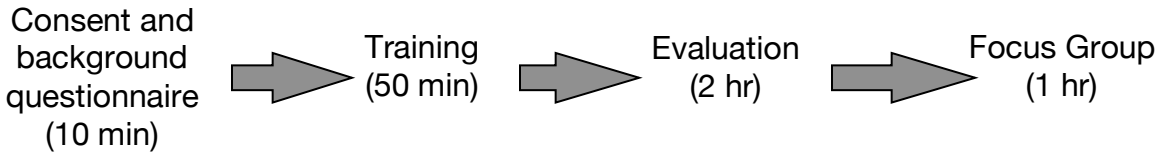
Target System: We chose an Identity Management (IdM) system for performing the heuristic evaluation. An IdM system is used to manage the digital identities of users in an enterprise and control the accesses of those identities to resources. Furthermore, the system allows request, review, approval, certification, and removal of access. An IdM system is used by various stakeholders in an organization. End-users use the system for creating accounts, requesting access, or changing their passwords. Managers use the system for approving employees' requests for access, reviewing and verifying the validity of their employees' access, and checking who have access to the resources they own. Security admins use the system to implement the requests for access, perform large scale provisioning and de-provisioning of access, and create roles.

We chose IdM system because of its significance. We showed (Section 2) that IdM systems have wider reach across the organization, and are used in day-to-day activities by various stakeholders. This increases the importance of usability in such tools. Additionally, IdM systems impact the functioning of other applications, because they integrate with and manage the access to those applications.

We installed CA Identity Manager 12.0 CR3 in a laboratory environment on a virtual machine using VMWare Server. Access to the system was through its web interface.

Study protocol: An overview of the study protocol is provided in Figure 5 we now describe the details of each step.

Figure 5 - Study Protocol Overview



We began by obtaining the participants' consent, and then asked them to complete a background questionnaire to obtain demographic information and data to assess their HCI and computer security background.

We then provided training on heuristic evaluation, and described the specific heuristic set to be used. We demonstrated the application of the heuristics in a running example of evaluating a network firewall system. The examples were designed to reflect problems with real network firewalls. For example, we described the application of ITSM heuristic #2 using a problem where two security admins make changes in the firewall rules, but there was no history of who made the changes. Or we described Nielsen's heuristic #4 using a problem where the firewall rules file contained the following rule: "eth0 inbound block", but in the UI, the same rule is shown as "block all incoming connections on eth0" (i.e., there is a lack of consistency between inbound, and incoming). We concluded the

training session with an introduction to the IdM system. In all cases, training material was presented through online slides with vocal narratives. That allowed us to provide exactly the same training to all participants regardless of their location.

After the training, participants inspected the interface individually. They had access to the list of ITSM (first paragraph about each heuristic in Section 3.3 including the bold title and italicized description) or Nielsen's heuristics (the version available at Nielsen, 2005), an instance of the IdM system, and an evaluation guide. In the evaluation guide, we limited the evaluation to a few typical usage scenarios (Rosson & Carroll, 2002) to manage the scope of the evaluation and guide evaluators during the evaluation. The participants could then login to the system as the various stakeholders while they performed the steps of the scenarios. An overview of the four scenarios used in the study is presented in Figure 6. We asked participants (1) to identify usability problems; and (2) for each problem, to specify the scenario and the heuristic (using an online form). Participants had two hours to perform the evaluation. We limited the evaluation time to control the time variable, avoid participant fatigue, and emphasize the discount and time-limited nature of heuristic evaluation.

After the evaluation, participants were provided with a post-evaluation questionnaire to rate their experience in using heuristics. We then conducted either a focus group (for sessions with multiple participants) or an interview (for sessions with one participant) to collect qualitative data on participants' experience.

We piloted and refined our study protocol and materials through several iterations. We performed two complete pilot study sessions (six and two participants); and we held several pilot tests as we iterated upon the individual study components, including the background questionnaire (six participants), the description of the heuristics (six), the training materials (two) and the evaluation guide (seven).

Figure 6 - Details of the four scenarios used during the comparative study.

Scenario	**Description**
Self-serve user creation	A *contractor* arrives at a company and wants to create a user account. He uses the self-service feature in the IdM system to create an account. Then a *SP* reviews and approves his request.
Bulk user creation	A *SP* receives a file containing all the users' job status changes in HR system, uploads the file to the IdM system, and troubleshoots errors.
Request privileges	An *employee* initiates an access request. The request is approved by a *manager*, and then reviewed, and implemented by a *SP*.
Certification	The security team initiates a certification campaign and sets a deadline. Managers receive requests to review and certify the privileges of employees. At the deadline, the security team closes the certification process by revoking all of the non-certified privileges.

4.1 Data Analysis

The following steps were performed by two researchers to aggregate the identified problems and determine their severity (any inconsistencies were resolved by consensus):

Aggregating problems: To aggregate problems found in each condition and generate a list of known problems we performed two steps:

1. *Problem Synthesis:* We first decomposed problems into their finest level of granularity. Compound problems should be decomposed as each component of the problem might have a certain severity, and therefore a priority for fixing. Compound problems include those that refer to different actions, different artifacts, or different mechanisms in the interface. Then, if an evaluator reported duplicate problems, we removed the duplicate. We then eliminated unknown problems, which we could not reproduce (e.g., it happened due to a sudden breakdown or crash in the system during the study, or the description of the problem was not understandable). We removed false positives, which had any of the following characteristics: (1) the problem was caused by the constraints or requirements of the underlying operating system or hardware/software infrastructure, (2) the problem was caused by the business constraints or requirements of the program, or (3) the reasoning of the participants in describing the problem was fallacious.

2. *Combining problems:* We began with an empty list of aggregated problems. Each identified problem was compared with the problems in the aggregated list and if it was not present, it was added to the list. Otherwise, the description of the problem and its associated heuristics in the aggregated list were updated.

Assigning severity ratings: Similar to (Nielsen, 1992), we asked four researchers with knowledge of usability and security, who had training in heuristic evaluation, to independently judge the severity of the problems. The participants used the following protocol to determine the severity of the problems: First, we asked them to answer the following questions: (1) Will the problem happen frequently to the users when they perform the activity? (2) Will it be easy for users to achieve their goal when they face the problem? (3) Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem? Then, we asked them to use subjective judgment to categorize the problem into one of the five levels of severity proposed by Nielsen (2005): 0- not a usability problem (I do not agree that this is a usability problem at all), 1- cosmetic (need not be fixed unless extra time is available on project), 2- minor (fixing this should be given low priority), 3- major (important to fix, so should be given high priority), and 4- catastrophe (imperative to fix this before product can be released). We gave the list of all problems to each expert without any information about the evaluators or heuristics with which the problems were found. Based on the mean rating, we categorized problems into major (mean severity > 2) and minor (mean severity ≤ 2). We demonstrate examples of problems and their severity in Figure 7. From 131 identified problems, we chose two high severity (3.00 and 2.75), and two low severity (1.25 and 1.25) problems that were found mainly by ITSM and Nielsen participants.

Figure 7 - Examples of the problems identified by the participants. "Context" describes the context in which the problem was identified. "Problem" describes the problem. "Freq." shows the number of times the problem is reported in the ITSM(I), and Nielsen(N) conditions. "Avg. Sev." shows the average severity of the problem. "H" shows the heuristics with which the problems were identified (e.g., I4 means ITSM heuristic #4). "IC" indicates that the problem could not be associated to a heuristic by an ITSM participant.

Context	Problem	Freq.		Avg. Sev.	H
		I	N		
As a part of Scenario #2, participants should upload a file that performed a bulk create, update, and remove. Out of 8 actions scripted in the file, 1 always failed, and the system showed a message that 7 records have been updated and 1 has failed.	There is no way for the user to know if the file causes an error. If the file is large, there is no way for the user to determine which record caused an error.	9	2	3.0	I1, I7, N1, N5
As a part of scenario #3, the employees could write their access request in a free text submission form, and then submit the request for processing.	During writing the request, if the user pressed "enter", the request was submitted, instead of creating a new line. There was no way to edit the request again, or revert the action.	1	4	2.75	IC, N5
As a part of scenario #1, the employee could perform self-registration from the login page by filling a form. After self-registration, the user was presented with a screen saying you are successfully logged out of the system.	There is no link back to the login, or any other pages from the login page.	5	8	1.25	N7, IC
As a part of scenario #3, the security admin should review the employee's access request and grant the required access.	There is no knowledge base for the security admin to see the consequences of such access or a way to get a second opinion on giving user the access.	3	0	1.25	I6

5. EVALUATION RESULTS

Figure 8 shows the classification of the problems in each condition. "Problem Reports" shows the initial number of problems reported by the participants. "Valid" shows the number of valid reported problems after the synthesis stage. "Known" shows the number of problems after the combining step in which we combined similar valid problems into one known problem. Figure 8 also shows the classification of known problems as either major or minor severity. We provide a summary of participants' individual performance in Figure 9. We calculated the performance of the strongest and

weakest participants, the proportion of problems found by the first and third quartile, and the ratio between these values. These proportions are calculated based on the total number of problems (131).

Figure 8 - Overview of the number and classification of identified problems in each condition.

Condition	Reports	Valid	Known	Major	Minor	False Positive	Unknown
ITSM	239	201	93	38	55	18	16
Nielsen	233	187	86	20	66	45	17
All	472	388	131	43	88	62	33

Figure 9 - Individual differences in participants' ability to find problems.

Condition	Max (%)	Min (%)	Q_1 (%)	Q_3 (%)	Max/Min	Q_3 / Q_1
ITSM	23.7	3.82	7.1	13.9	6.2	2.0
Nielsen	18.3	3.1	5.9	11.5	6.0	1.9

Performance of heuristics: We compared the performance of the heuristics used in each condition according to their thoroughness, reliability, validity, and effectiveness. We will discuss the cost-effectiveness in Section 5.3. We compared the results from two different perspectives: (1) a per condition basis: we compare the output of evaluation as a whole. (2) a per evaluator basis: we compare the performance of individual participants.

Thoroughness: We calculate thoroughness as the proportion of the problems identified in each condition. Our results show that the evaluation with the ITSM heuristics resulted in finding 71% of total known problems (93 out of 131) while the evaluation with Nielsen's heuristics resulted in finding 66% of them (86 out of 131). In some cases, finding fewer, but more severe, problems might be more important than finding many minor problems. To examine this, we used the notion of Weighted Thoroughness (WT) by increasing the weight of the problems based on their severity (Hartson et al., 2001). Using equation: $WT = \frac{\sum_{p \epsilon known(ITSM)} Severity\,(p)}{\sum_{p \epsilon known} Severity\,(p)} \times 100$ (we used an equivalent equation for Nielsen condition), the weighted thoroughness of ITSM and Nielsen's heuristics are 77% and 60% respectively.

To compare two conditions on a per evaluator basis, we tested the following hypothesis: (1) H_1: Participants will report more problems if they use ITSM heuristics than Nielsen's. H_0: There is no difference in the number of reported problems. The result of a Mann-Whitney U test did not reject H_0.

Reliability: It is important for a set of heuristics to be able to identify major usability issues as they may seriously hinder the ability of the user to operate the system effectively and efficiently. The results (Figure 8) show that participants using the set of ITSM heuristics found almost twice as many major usability problems than the participants using Nielsen's set.

We tested the following hypothesis to show the difference in severity on a per-evaluator basis: H_1: The average severity of the problems reported by participants will be

higher if those individuals use ITSM heuristics than if they use Nielsen's. H_0: There is no difference in the average severity. The result of a Mann-Whitney U test rejected H_0 in favor of H_1 (*U=26, Z=-3.309, p=0.001*).

Validity: We examined whether the evaluation with the ITSM heuristics generated fewer false positives than Nielsen's. Participants using the ITSM heuristics reported 201 valid problems and 18 false positives, whereas participants using Nielsen's heuristics reported 187 valid problems and 45 false positives. The ITSM heuristics yielded fewer false positives (Figure 8) than Nielsen's heuristics. Comparing the number of unknown problems identified in each condition revealed a very small difference between conditions.

We tested the following hypothesis about difference in the number of false positives on a per-evaluator basis: H_1: participants will report fewer false positives if they use ITSM heuristics than if they use Nielsen's. H_0: There is no difference in the number of false positives. The result of a Mann-Whitney U test rejected H_0 in favor of H_1 (*U=38, Z=-2.823, p=0.005*).

Effectiveness: We calculated the effectiveness using equation suggested by Hartson et al. (2001): $Effectiveness = 1/[\alpha\left(\frac{1}{validity}\right) + (1-\alpha)\left(\frac{1}{thoroughness}\right)]$. We used the same weight (α) for validity and thoroughness. Our results showed that the effectiveness of ITSM heuristics was 0.80 and the effectiveness of Nielsen's heuristics was 0.72.

The number of evaluators required to perform the evaluation: To replicate Nielsen's original analysis (Nielsen & Molich, 1990), we formed aggregates of participants and found the proportion of usability problems identified by each size of aggregate. Following Nielsen's methodology, we calculated the proportion of found problems based on the total number of problems found in each condition. The result is depicted in Figure 10. The graph shows that increasing the number of evaluators will increase the proportion of the identified problems, but the rate of the increase diminishes as we increase the number of evaluators. The two plots from our experiment are very similar and they show a similar trend as compared to the results from the Mantel, Groove, and GroupDraw experiments.[6] Yet, Nielsen's experiment shows faster diminishment compared to our results.

We illustrate the distribution of the known problems that are found by participants in the ITSM or Nielsen condition in Figure 11. To generate the diagram, we grouped participants based on their condition and then sorted them from weak to strong (participant A is stronger than participant B, if A found more problems than B). We also sorted the problems from easy to hard (problem A is easier to find than problem B, if A was found by more participants than B). We highlighted the severity of the problems by

[6] To allow comparison, and since the mentioned experiments employed more evaluators, we assumed that the total number of problems in each experiment was equal to the problems found by aggregate size of 14.

color. The diagram shows that, similar to Nielsen's original experiment (Nielsen & Molich, 1990), there are easy problems that are overlooked by strong participants while there are hard problems that are only found by weak participants. Also, there were major problems that were only found by weak participants and there were minor problems that were only found by strong participants. This confirms Nielsen's argument that heuristic evaluation is a method that should be done collectively (i.e., no strong evaluator can uncover all of the major problems). Figure 11 also shows that there was relatively little duplication between participants in each condition. We further discuss the lack duplication in Section 6.

Figure 10 - Average proportion of problems found by aggregate of participants in ITSM and Nielsen conditions. We also overlaid the results from Nielsen's Mantel experiment (Nielsen & Molich, 1990), and Baker's Groove and GroupDraw (Baker et al., 2002) experiments to allow comparisons.

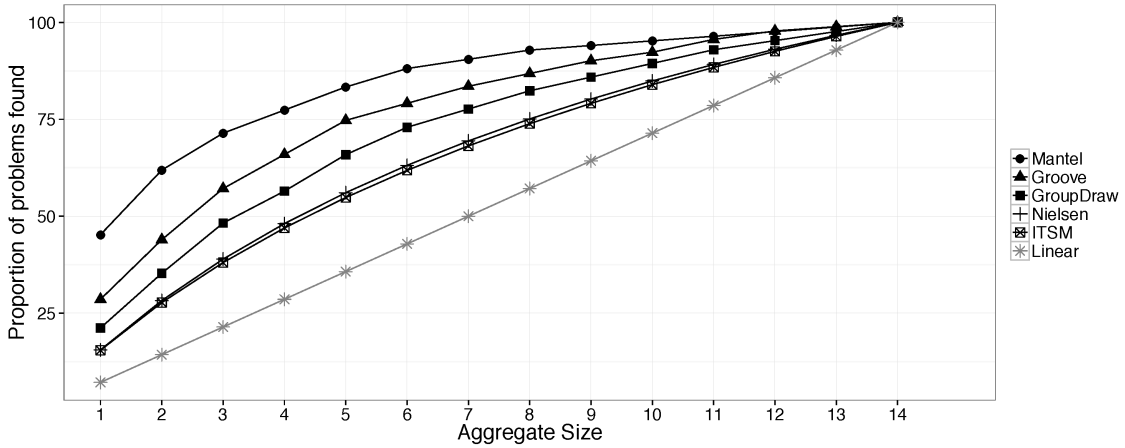
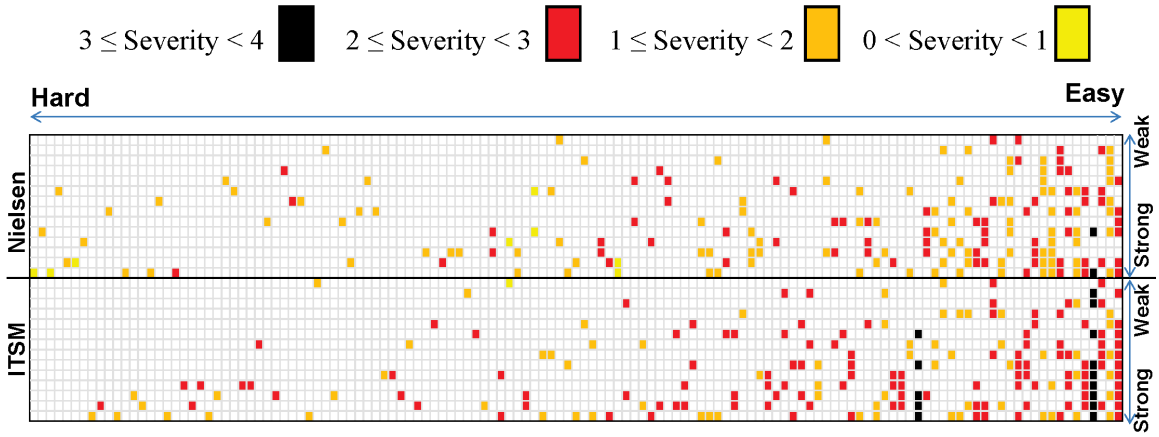


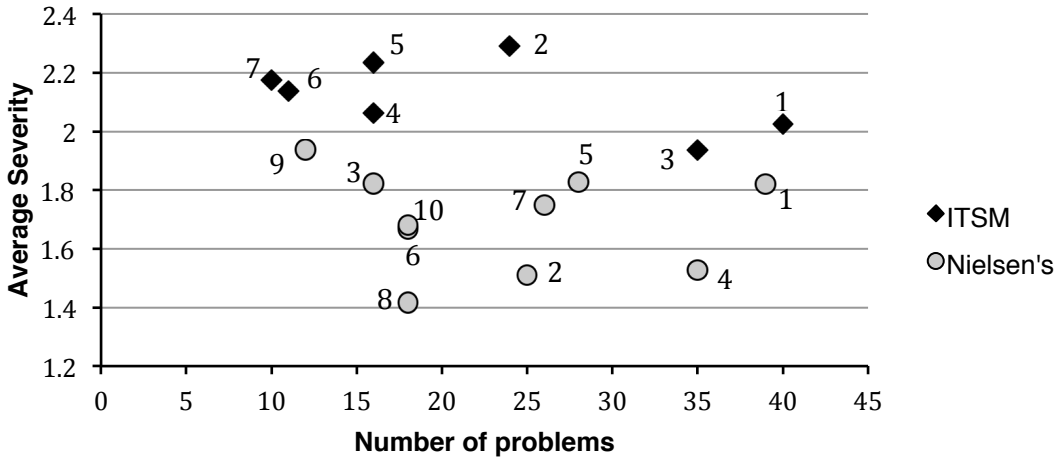
Figure 11 - Problems identified by each participant in each condition. Each row corresponds to a participant and each column corresponds to a problem. Participants in each condition are sorted from top (weak) to bottom (strong) and problems are sorted from right (easy) to left (hard).



5.1. Performance of Individual Heuristics

To see which heuristics contributed the most in finding usability problems, we visualized the number and mean severity of known problems associated with each heuristic in Figure 12. The graph shows that the severity of the problems found with ITSM heuristics is higher than the ones found with Nielsen's heuristics. The graph also indicates that ITSM heuristics #6 and #7 were associated with the fewest problems and ITSM heuristic #1 was associated with the most problems. The average severity of the problems associated with ITSM heuristic #2 was the highest and Nielsen's heuristic #8 was the lowest.

Figure 12 - The number and mean severity of problems identified by each heuristic. The ITSM heuristics are shown using black diamonds and Nielsen's heuristics are shown using gray circles. Each heuristic is labeled with its number.



As our results indicated a large overlap between the problems found using ITSM and Nielsen's heuristics (i.e., there were known problems that were found in both conditions), we further investigated the similarity between the two sets. For this we calculated a similarity metric between the two heuristics (A and B) as follows: $\frac{|A \cap B|}{|A \cup B|} \times 100$.

The result of the similarity analysis is presented in Figure 13. For each ITSM heuristic, we highlighted the most similar Nielsen heuristic. These similarities are not surprising. For instance, the ITSM #1 and Nielsen's #1 can both lead to finding a subset of visibility problems; both ITSM #3 and Nielsen's #7 can lead to problems related to interacting with users with different levels of expertise, ITSM #4 and Nielsen's #5 will both lead to preventing errors by applying constraints. ITSM #6 and Nielsen's #9 are also shown to be similar, as providing users with the required knowledge will help them understand errors and recover from them. We also show how each of the Nielsen and ITSM heuristics performed in finding problems unique to their condition (i.e., problems that were not reported in ITSM and Nielsen conditions respectively), and the average severity of those unique problems in Figure 14. We will corroborate this data with

participants' feedback to determine which of Nielsen's heuristics complement the ITSM heuristics in Section 6.

Figure 13 - Similarity between individual ITSM and Nielsen's heuristics. Each cell shows the value of similarity metric for the heuristics denoted by row and column indexes. For each ITSM heuristic, the cell with the highest number (i.e., the most similar Nielsen heuristic) is highlighted.

		The Nielsen Heuristics									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
The ITSM Heuristics	**1**	29.5	16.1	14.3	17.2	15.3	16	17.9	11.5	10.6	11.5
	2	14.5	6.5	8.1	7.3	15.6	2.4	8.7	0.0	12.5	7.7
	3	25.4	11.1	13.3	18.6	18.9	15.2	27.1	12.8	9.3	10.4
	4	14.6	7.9	6.7	8.5	18.9	9.7	13.5	6.3	3.7	17.2
	5	5.8	7.9	0.0	4.1	4.8	6.3	10.5	3.0	0.0	0.0
	6	6.4	2.9	3.8	0.0	5.4	3.6	2.8	3.6	15.0	3.6
	7	6.5	2.9	4.0	4.7	8.6	3.7	9.1	3.7	10.0	0.0

Figure 14 - Ability of each of Nielsen's and the ITSM heuristics to find problems unique to their condition. The "Proportion of unique" row shows the proportion of problems uniquely found in the Nielsen or ITSM conditions using the corresponding heuristic. The "Average severity" row shows the average severity of those unique problems.

	The Nielsen Heuristics										The ITSM Heuristics						
	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7
Proportion of unique	0.41	0.52	0.19	0.49	0.32	0.39	0.35	0.61	0.25	0.56	0.43	0.50	0.37	0.25	0.63	0.55	0.50
Average Severity	1.47	1.31	2.06	1.10	1.49	1.57	1.42	1.18	1.42	1.33	2.12	2.31	1.87	2.38	2.35	2.29	2.10

5.2. Impact of participants' background on their performance

Nielsen suggests that the evaluator's HCI and domain expertise are two factors that influence the quality of heuristic evaluation (Nielsen, 1992). We analyzed the HCI and computer security background of the participants to find the correlation between expertise (years of HCI and computer security experience, number of previously performed heuristic evaluations), and performance (number of raw, known, false positive problems, and average severity).

We first used a factor reduction technique to find the possible medium or strong correlations and then investigated correlations and their statistical significance with either Pearson's product-moment coefficient (for normally distributed data) or Kendall tau rank correlation coefficient (for non-normal data). In the Nielsen condition, we found a strong negative correlation between the number of previously performed heuristic evaluations and the average severity of reported problems ($r=-0.70, p<0.05, N=14$). In the ITSM condition, we found a strong positive correlation between the number of previously performed heuristic evaluations and the number of false positives ($\tau=0.55, p<0.05, N=14$). For the overall study data, we identified a medium to strong correlation between the years of HCI experience and the number of reported problems ($r=0.47, p<0.05, N=28$). We did not find any correlation between the severity of the reported problems

and the background of the participants (i.e., between severity of the reported problems and years of HCI [τ=-0.17, p=0.21, N=28], professional computer security [τ=0.18, p=0.28, N=28], or academic computer security [τ=0.19, p=0.22, N=28] experience.) In Section 4, we reported that the average length of computer security experience in ITSM condition was more than three times higher than in Nielsen condition before removing the outlier. Yet, as we showed above, there is no correlation between the amount of computer security experience and the severity of the reported problems. This suggests the differences are due to the condition rather than participants' security experience.

5.3. Participants' Feedback in Post-evaluation Questionnaire

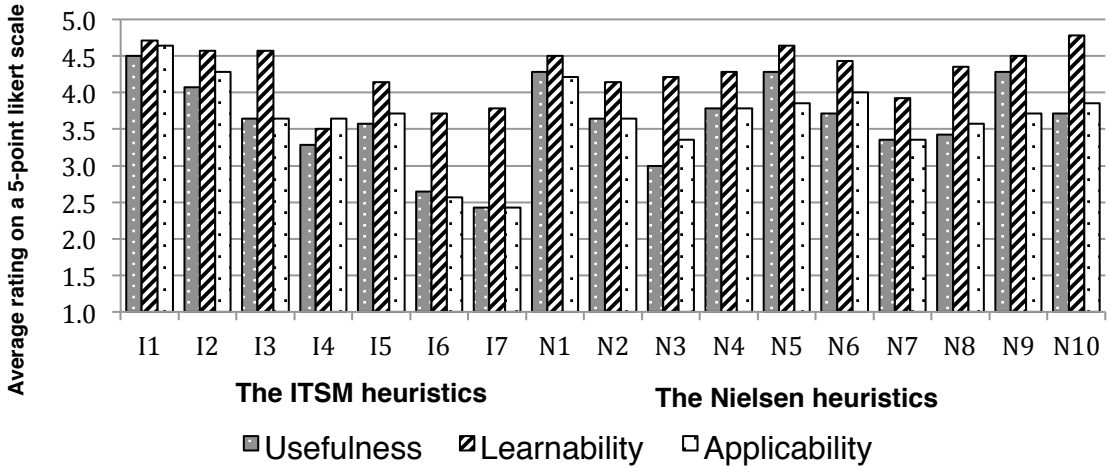
We asked our participants to evaluate with a 5-point Likert scale (5=strongly agree, 1-strongly disagree) how useful the set of heuristics was in identifying usability problems (usefulness), how easy it was to understand and learn the heuristics (learnability), and how easy it was to apply the heuristics to the IdM system (applicability). The mean usefulness, learnability, and applicability ratings for ITSM condition were 3.14, 3.36, and 2.86 respectively, and for Nielsen condition were 3.36, 3.57, and 3.5. We conducted a Mann-Whitney U test to evaluate whether the set of heuristics used impacted the usefulness, learnability, and applicability, as reported by our participants. Although, the ITSM heuristics were new to our participants there was no significant difference between the ratings for the two sets of heuristics. As we highlighted before, one measure of cost-effectiveness of a usability evaluation method is the effort required to learn it. Therefore, our results suggest that there is no statistically significant difference between the cost-effectiveness of the two sets of heuristics.

We also asked participants to evaluate with a 5-point Likert scale the usefulness, learnability, and applicability of each individual heuristics. The mean scores of the ITSM and Nielsen's heuristics are shown in Figure 15. Repeated measures ANOVA calculations between the mean scores of individual Nielsen's heuristics revealed only a significant difference in terms of usefulness $F(9, 117)=2.40$, $p<0.05$. Post hoc tests using Bonferroni correction showed a significant difference between heuristics N#1 and N#3, and N#5 and N#3. Repeated measures ANOVA tests for the ITSM heuristics determined a statistically significant difference in heuristics usefulness $F(6, 78)=10.18$, $p<0.005$, learnability $F(6, 78)=6.92$, $p<0.005$ and applicability $F(6, 78)=12.45$, $p<0.005$. Post hoc tests using Bonferroni correction shows that the significant difference in usefulness was mainly caused by heuristics I#6 and I#7, the significant difference in learnability was caused by the difference between heuristic I#4 and heuristics I#1 and I#3, and the difference in applicability was mainly caused by heuristic I#7.

5.4. Qualitative feedback during focus group/interview session

In this section, we provide a summary of participants' feedback during interviews and focus groups. We identify participants in the ITSM condition by PI1 to PI14, and the Nielsen condition by PN1 to PN14.

Figure 15 - Mean scores of participants' reported usefulness, learnability, and ease of application for the different heuristics (5=strongly agree, 1=strongly disagree).



We asked open-ended questions about the usefulness, ease of understanding, and applicability of the heuristics. Furthermore, we asked participants if they noticed problems that could not be found with or associated to the heuristics, and to improve the current heuristics set or add a new heuristic to it.

In the Nielsen condition, all participants confirmed the heuristics' usefulness, e.g., PN4 explained: "*They give me a standard way to review each of the screens. [...] At least be able to evaluate based on a common set of methods or processes*". But it was challenging for some of the participants to apply heuristics to the system, without understanding the background of the real users of the system, e.g., PN8 explained: "*I found them useful for some of the actors [in the scenarios]. When it gets to [a manager], it becomes harder to get into the user's mindset. And when we get to the [security admin], it is not useful at all because he is an expert*". This point was also confirmed by PN10, and PN13 who indicated that the heuristics such as "Flexibility and efficiency of use", and "Match between system and real world" require understanding of flexibility for a security admin and his mental model of the real world. PN8 also indicated that many of the problems that might be important for end-users, might not be as important for security admins who will be trained to use the tool.

Similarly, the ITSM participants found the heuristics useful, but not without problems. Many of the participants (PI7, PI5, PI6, PI1, PI14, PI10) indicated that while they understood heuristics I#6 (Capturing, sharing, and discovery of knowledge) or I#7 (Verification of knowledge), they were not applicable to the four scenarios in the study. Yet, PI13 indicated that it took some time to grasp the last two heuristics: "*at the beginning, I was very focused on the first heuristics. But it was towards the end that I was starting to think about the problems related to [ITSM #6, #7]. But when you start thinking about them, it becomes intuitive to see the problems related to those*." On the

contrary, PI8 describe ITSM heuristic #4 (Rules and constraints) as hard to apply: "*I had a hard time to apply [ITSM #4] because it can be applied at different levels. You can say the system should limit what user can enter in his request as much as possible, or user can enter whatever he wants and then it is up to manager to review and decide whether user entered a valid request*." PI8 then expressed disagreement that tools always "should" constrain possible actions, and suggested that tools "could" constrain possible actions depending on the situation.

We asked participants in the Nielsen condition about the aspects of the system not covered by the heuristics. Four participants indicated that problems with the workflow cannot be classified in Nielsen's heuristic and that they classified them as lack of showing different steps of workflow (PN4, PN10), lack of ability to revert back to one of the previous steps of the workflow (PN6), and lack of a coherent workflow (PN1). PN12 believed that N#1 should be changed to "Visibility" because visibility can go beyond the system status. PN7 described that the interface offered too many options for performing tasks, and no heuristic covered that. Then, PN7 suggested a heuristic for changing the presentation based on the role of the user in the organization. Similarly, PN8 suggested dividing N#8 to two heuristics: (1) aesthetics and, (2) the level of detail for expert and non-expert users.

In the ITSM condition, PI6, PI14, and PI12 asked for Nielsen's heuristics set to use in addition to the ITSM heuristics, and other participants suggested individual Nielsen's heuristics. For example, PI7 indicated the need for an error prevention heuristics, as well as better error messages. PI1 suggested an error recovery like undo or redo. The need for a consistency heuristic was indicated by PI2, PI14, and PI6. A heuristic for organization of the screen was suggested by PI2, PI12, and PI6. Understandable language was indicated by PI10, PI12, PI14.

We mapped the heuristics that participants in each condition indicated as missing to one of the heuristics in the other condition. As a result, we saw participants in the Nielsen condition found ITSM heuristics #1 (visibility of activity status), #3 (flexible representation of information), and #4 (rules and constraints) necessary. Participants in the ITSM condition indicated the need for Nielsen's heuristics #2, #3, #4, #5, #8, and #9.

Almost all participants explained that they first identified problems and then tried to "assign" each problem to one of the heuristics. Yet, they still found the heuristics helpful in finding problems: e.g., PN2 explained: "*[The heuristics] remind you of existence of possible problems. I might forget to look at help and documentation if I don't have the heuristics*." PN9 had heuristics in mind when looking at the interface: "*For me, I found the problem and then I matched it. But I had heuristics in my mind, and when I looked at the interface I was thinking if it breaks anything*." PN8 explained the role of heuristics in disambiguating problems: "*I look for a submit button, I don't see it. I think it might be a problem. But then heuristics help me find exactly what the problem might be*." PI12 described the role of heuristics in predicting problems and designing test cases to uncover them: "*As soon as I read the description for scenario one, I thought 'oh I bet that is going to break heuristic number five'. And then I figured out a little test case and tested it*."

After this point was brought up by PI12, the two other participants in the same focus group confirmed it.

When commenting on the study procedures, participants indicated that if they had more time, they would have found more usability problems.

6. DISCUSSION

The evaluation results suggest that our heuristics performed well overall in finding usability problems in ITSM tools. In this section, we interpret the results and discuss their implications.

Few overlaps between individual evaluators: We observed fewer overlaps between problems identified by evaluators in our experiment than problems identified by evaluators in Nielsen's original experiment (Nielsen & Molich, 1990). In both conditions of our experiment, only three problems were identified by the majority of participants, and more than half of the problems were identified by only one. In contrast, in Nielsen's evaluations of the Mantel and Savings systems (Nielsen & Molich, 1990), only one and two problems respectively were identified by just one participant. In Baker et al. (2002) evaluation of GroupDraw and Groove, 14 out of 64 and 5 out of 43 problems were found only by one participant. Our results show fewer overlaps between problems identified by different participants, compared to Nielsen's and Baker's results. Four factors might have contributed to this outcome. First, the evaluated IdM system was fairly large; the participants had to visit 20 different web pages in order to successfully complete all scenarios. This multitude of web interfaces provided an opportunity for finding more diverse problems than in the cases of systems used in Nielsen's or Baker's evaluations (e.g., Mantel only had a single screen and a few system messages, GroupDraw had two screens). Second, we used fewer participants (14 per condition) compared to 77, 34, 25, and 27 participants in Mantel, Savings, GroupDraw, and Groove systems, respectively. Third, the evaluated system was a commercial product rather than a prototype and, it did not contain many obvious usability problems. Fourth, participants thought they could find more problems if they had more time. For reference, Nielsen et al. do not mention any time constraints during their study; and in the Baker study, researchers allowed participants to use as much time as they needed to evaluate the interfaces. Our results illustrate how hard and time consuming the heuristic evaluation of ITSM tools in the size and complexity of the IdM systems is; we discuss why we limited the evaluation time to two hours in Section 4.

Large overlaps between known problems reported in the ITSM and Nielsen conditions: The ITSM heuristics were consistent with activity theory and Nielsen's heuristics were consistent with action theory. As a result, we expected to have very few problems that overlap the two conditions. But our results show that 48 problems (37%) were found in both conditions. Three factors might have resulted in this observation. First, participants in the ITSM condition could remember Nielsen's heuristics, which helped them see problems at the action level. Second, most of the participants found a problem first, and then fit it into one of the heuristics. Third, the similarity between heuristics (Figure 13) could also result in overlaps.

Usefulness of ITSM heuristics: It is encouraging that despite the novelty of the ITSM heuristics, participants found them to be no less effective, easy to use, or easy to learn than Nielsen's heuristics. Yet, when we looked at the individual heuristics, we saw that ITSM heuristics #6 (capturing, sharing and discovery of knowledge) and #7 (verification of knowledge) were not as useful and easy to apply. Looking at the number of problems reported using those heuristics confirms this observation (Figure 12). We can provide several explanations for this observation. First, the study scenarios did not include extensive deployment or configuration tasks that involve verification to a great extent, or tasks that deal with unforeseen conditions or troubleshooting, which require extensive knowledge sharing. Second, one participant indicated that the last two heuristics were ignored because of focusing on the first heuristics in the list. Therefore, the order of the heuristics might have influenced their use in our time-limited evaluation sessions. Our judgment here is based on the participants' feedback and it needs further study. Third, we believe that the ITSM heuristics #6 and #7 were less open to interpretation than heuristics I#1 (visibility of activity status) or I#3 (flexible representation of information), which are applicable over a broad range of tasks.

Specificity of heuristics to the ITSM domain: While our goal was to develop specific ITSM heuristics, some of the heuristics seem to be rather general and applicable to other domains as well. This generality was the result of finding general guidelines that eventually led to creation of heuristics. Looking at the data that supports those general guidelines shows that ITSM shares characteristics with other domains. For example, it shares complexity with IT, creativity with software development, and uncertainty with military. As a result, some of the recommendations for designing better ITSM systems might be similar to the recommendations for designing other systems with similar characteristics.

An ideal set of heuristics for evaluation: Our results suggest that using the ITSM heuristics leads to finding more severe problems. However, using Nielsen's heuristics led to finding a unique set of problems that couldn't be found using ITSM; while those problems might not be as severe, addressing them can improve the interaction between user and the system. Therefore, we believe that the ITSM and Nielsen's heuristics can offer different perspectives for evaluation and they can be combined and used in three different ways all of which have trade-offs that should be considered according to the ITSM system being evaluated and the resources available for the evaluation: First, both sets can be used together in one evaluation session. This approach gives participants a holistic view of possible problems at both the action and activity levels. On the contrary, Nielsen argued that the use of more than 10 heuristics is not effective, and evaluators cannot remember all of the heuristics. Furthermore, evaluators who have previous experience with Nielsen's heuristics might tend to focus more on those heuristics and might ignore the ITSM heuristics. The second approach would be to use a subset of Nielsen's heuristics (at most three to be consistent with Nielsen's recommendation of using at most 10 heuristics) in addition to the ITSM heuristics. Our participants suggested the need for six of Nielsen's heuristics. Based on the data from Figure 14, we can suggest the use of Nielsen's heuristics #2 (match between system and the real world), #4 (consistency and standards), and #5 (error prevention). The benefit of this approach is reducing the evaluators' mental overload, and allowing them to find action level

problems that are critical to the target application. The drawback is that participants might focus on those three Nielsen's heuristics that they know, rather than the ITSM ones. The suggestion of specific Nielsen's heuristics is solely based on the data of this study, and the evaluated IdM system. The choice of the heuristics should be changed depending on the goal of the evaluation. For example, if the main design goal of a project is aesthetics, one can replace one of Nielsen's other heuristics with N#8 (aesthetic and minimalist design). The third approach would be to use Nielsen's and ITSM heuristics in separate evaluation sessions by the same or different evaluators. We expect this approach to have the highest thoroughness and yet the highest cost.

The impact of participants' background on their performance: Our results suggest that the average years of HCI experience is positively correlated with the number of reported problems, but not with their severity. This result supports Nielsen's finding that regular specialists will find more usability problems than novice evaluators. On the other hand, the results suggest that the number of previously performed heuristic evaluations negatively impacted the severity of the problems in the Nielsen condition and false positives in the ITSM condition. This observation was surprising. We hypothesize that participants with prior heuristic evaluation experience tend to evaluate the systems for end-users, with a focus on aesthetics of the interface. This resulted in minor problems in Nielsen condition, and false positives in the ITSM condition. Further study is needed to validate the reasons behind this observation.

Generalizability of evaluation results: If we were to replicate the comparative evaluation study for a different ITSM tool (e.g., one of those listed in Section 2), we would expect the ITSM heuristics to be still applicable and to find more severe problems than with Nielsen's. The scope of the empirical data, which the ITSM heuristics were created based on was ITSM tools in general, rather than a specific IdM system. Also, the heuristics were supported by a general HCI theory. This leads us to believe the ITSM heuristics are general enough for evaluating most ITSM tools.

At the same time, the performance of individual heuristics may vary for different categories of ITSM tools. As we discussed in Section 2, IdM systems have a wide reach across the organization, and are used by many users. Therefore, the study participants reported a large number of usability problems for the visibility of activity status. In contrast, a security operations tool such as network traffic analyzer has a narrow reach across the organization and is mainly used by SPs. Such a tool should help SPs deal with complex and large scale network traffic logs, and detecting malicious, unknown content in network traffic. Therefore, the evaluators of the tool may focus on I#3 (flexible representation of information), and I#6 (capturing, sharing, and discovery of knowledge) rather than I#1 (visibility of activity status). The evaluation results could vary between individual systems of the same type. For example, the evaluated IdM system offered rather meaningful error messages to users. Therefore, heuristic N#9 (help users recognize, diagnose, and recover from errors) was the least used Nielsen heuristic. Another IdM system may present errors as, say alphanumeric codes, or without indicating exact problems. In evaluating such a system, we might see more use of N#9.

7. LIMITATIONS AND FUTURE WORK

We used participants with HCI and heuristic evaluation background, and as we discussed in Section 4, we made a tradeoff to compare the two sets of heuristics in an ecologically valid setting. As a result, we cannot make arguments about the performance of the two sets of heuristics when they are used by participants who were not previously exposed to heuristic evaluation.

While increasing the number of participants in either of the conditions did not saturate the list of identified problems, we observed that the rate of finding problems decreased. Continuing the experiment with more participants would certainly allow us to find the point of diminishing returns in the number of problems. But comparing our results with the GroupDraw evaluation (Baker et al., 2002) suggests that even doubling the number of participants would not allow us to observe saturation. Furthermore, our study required a four-hour time commitment from participants with an HCI background, which made recruitment challenging. Because determining such a saturation point was not the main goal of our study, we leave such investigation for future work.

There are several opportunities for improvement and future work. First, during the problem synthesis stage, the severity of problems was estimated by four severity raters with a background in both usability and security. While this is a standard approach for determining the severity of problems in heuristic evaluation, it is only an approximation of severity. Asking the opinion of real system users to determine the severity of the problems is another method. Neither of these approximations might be precise, but combining the ratings would increase confidence.

CONCLUSION

In this paper, we reviewed the prior research on heuristic creation. We then described our methodic way of creating domain specific usability heuristics, which we applied to create a set of usability heuristics for evaluation of ITSM tools. To examine the applicability of the heuristics, we compared their use with Nielsen's heuristics for the evaluation of an IdM system. We tried to maximize the ecological validity of the study by using a real ITSM tool, and recruiting participants with an HCI background and familiarity with heuristic evaluation.

Our results show that a combination of a top-down and bottom-up approach resulted in a set of heuristics that were applicable to the target domain (as they were based on the domain-specific data), and yet were general enough to help evaluators find diverse set of problems. Comparing the new heuristics to Nielsen's heuristics revealed that the severity of the problems found by participants in the ITSM condition was higher than those found in Nielsen condition. Furthermore, our participants found the ITSM heuristics to be as relevant, easy to apply, and easy to learn as Nielsen's. The results of our evaluation also shed light on the use of the heuristic evaluation for evaluating a complex domain-specific system. While Nielsen found that five evaluators are able to find about two thirds of the problems, in our evaluation of the IdM system, five evaluators only found about half of the problems found by 14 evaluators. Additionally, the complexity and scale of the

system can result in a lack of overlapping problems between evaluators. Finally, our results show that Nielsen's heuristics can also be effective in finding a class of problems in ITSM tools that cannot be found by the ITSM heuristics. Therefore, we discussed three approaches for using a combination of the two sets of heuristics.

The ITSM heuristics are a component of tool usability evaluation and can be used as a discount method to find usability problems in prototypes or actual tools. These problems can be further investigated by a user study or a contextual inquiry session. Design guidelines (e.g., Jaferian et al., 2008) can then be used to address the problems.

REFERENCES

Baker, K., Greenberg, S., & Gutwin, C. (2002). Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (pp. 96–105). New Orleans, Louisiana, USA: ACM.

Barrett, R., Maglio, P. P., Kandogan, E., & Bailey, J. (2005). Usable autonomic computing systems: The system administrators' perspective. *Advanced Engineering Informatics*, *19*(3), 213–221.

Barrett, R., Prabaker, M., & Takayama, L. (2004). Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. In *CSCW '04* (pp. 388–395). Chicago, IL, USA.

Bauer, L., Cranor, L. F., Reeder, R. W., Reiter, M. K., & Vaniea, K. (2009). Real life challenges in access-control management. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems* (pp. 899–908). Boston, MA, USA: ACM.

Beal, B. (2005). IT security: the product vendor landscape. *Network Security*, *2005*(5), 9–10.

Botta, D., Muldner, K., Hawkey, K., & Beznosov, K. (2011). Toward Understanding Distributed Cognition in IT Security Management: the Role of Cues and Norms. *Cogn. Technol. Work 13*(2), 121-134.

Botta, D., Werlinger, R., Gagné, A., Beznosov, K., Iverson, L., Fels, S., & Fisher, B. (2007). Towards Understanding IT Security Professionals and Their Tools. In *Proc. of Symp. On Usable Privacy and Security (SOUPS)* (pp. 100–111). Pittsburgh, PA.

Carroll, J. M., Neale, D. C., Isenhour, P. L., Rosson, M. B., & McCrickard, D. S. (2003). Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies*, *58*(5), 605 – 632.

Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Trans. Inf. Syst.*, *10*(2), 181–212.

Charmaz, K. (2006). *Constructing Grounded Theory*. SAGE publications.

Chiasson, S., Oorschot, P. C. van, & Biddle, R. (2007). Even Experts Deserve Usable Security: Design guidelines for security management systems. In *SOUPS Workshop on Usable IT Security Management (USM)* (pp. 1–4). Pittsburgh, PA.

Dourish, P. (2001). Seeking a foundation for context-aware computing. *Hum.-Comput. Interact.*, *16*(2), 229–241.

Dourish, P., & Redmiles, D. (2002). An approach to usable security based on event monitoring and visualization. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms* (pp. 75–81). Virginia Beach, Virginia: ACM.

Engeström, Y. (1999). Activity theory and individual and social transformation. *Perspectives on activity theory*, 19–38.

Engeström, Y. (2001). Expansive Learning at Work: Toward an activity theoretical reconceptualization. *Journal of Education and Work*, *14*(1), 133–156.

Erickson, T., & Kellogg, W. A. (2000). Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.*, *7*(1), 59–83.

Gagné, A., Muldner, K., & Beznosov, K. (2008). Identifying Differences between Security and other IT Professionals: a Qualitative Analysis. In *HAISA'08: Human Aspects of Information Security and Assurance* (pp. 69–80). Plymouth, England.

Goodall, J. R., Lutters, W. G., & Komlodi, A. (2004). I Know My Network: Collaboration and Expertise in Intrusion Detection. In *CSCW '04* (pp. 342–345). Chicago, IL, USA.

Grance, T., Stevens, M., & Myers, M. (2003). *NIST Special Publication 800-36, Guide to selecting information technology security products*. National Institute for Standards and Technology.

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

Haber, E. M., & Bailey, J. (2007). Design guidelines for system administration tools developed through ethnographic field studies. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer human interaction for the management of information technology* (pp. 1:1–1:9). Cambridge, Massachusetts: ACM.

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

Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, *7*(2), 174–196.

Jaferian, P., Botta, D., Hawkey, K., & Beznosov, K. (2009). 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* (pp. 46–55). Baltimor, Maryland, USA: ACM.

Jaferian, P., Botta, D., Raja, F., Hawkey, K., & Beznosov, K. (2008). Guidelines for designing IT security management tools. In *Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology* (pp. 7:1–7:10). San Diego, California: ACM.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. (1991). 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* (pp. 119–124). New Orleans, Louisiana, United States: ACM.

Kandogan, E., & Haber, E. M. (2005). Security Administration Tools and Practices. In L. F. Cranor & S. Garfinkel (Eds.), *Security and Usability: Designing Secure Systems that People Can Use* (pp. 357–378). O'Reilly Media, Inc.

Kaptelinin, V., & Nardi, B. (2006). *Acting with technology: Activity theory and interaction design*. MIT Press.

Kaptelinin, V., Nardi, B., Bodker, S., Carroll, J., Hollan, J., Hutchins, E., & Winograd, T. (2003). Post-cognitivist HCI: second-wave theories. In *CHI '03 extended abstracts on Human factors in computing systems* (pp. 692–693). Ft. Lauderdale, Florida, USA: ACM.

Kesh, S., & Ratnasingam, P. (2007). A knowledge architecture for IT security. *Commun. ACM*, *50*(7), 103–108.

Kotulic, A. G., & Clark, J. G. (2004). Why There Aren't More Information Security Research Studies. *Information & Management*, *41*(5), 597–607.

Kraemer, S., & Carayon, P. (2007). Human errors and violations in computer and information security: The viewpoint of network administrators and security specialists. *Applied Ergonomics*, *38*(3), 143–154.

Kuutti, K. (1995). Activity theory as a potential framework for human-computer interaction research (pp. 17–44). Cambridge, MA, USA: Massachusetts Institute of Technology.

Leont'ev, A. (1974). The Problem of Activity in Psychology. *Journal of Russian and East European Psychology*, *13*(2), 4 – 33.

Maglio, P. P., Kandogan, E., & Haber, E. (2003). Distributed cognition and joint activity in collaborative problem solving. In *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*.

Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., & Ames, M. (2003). Heuristic evaluation of ambient displays. In *Proc. CHI '03* (pp. 169–176). Ft. Lauderdale, Florida, USA: ACM.

McGann, S., & Sicker, D. C. (2005). An analysis of security threats and tools in SIP-Based VoIP Systems. In *2nd VoIP Security Workshop* (pp. 1–8). Washington DC, USA.

Muller, M. J., & McClard, A. (1995). 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* (pp. 115–116). Denver, Colorado, United States: ACM.

Nardi, B. A. (Ed.). (1995). *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MA, USA: Massachusetts Institute of Technology.

Neale, D. C., Carroll, J. M., & Rosson, M. B. (2004). Evaluating computer–supported cooperative work: models and frameworks. In *CSCW '04* (pp. 112–121). ACM Press.

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 152–158). Boston, Massachusetts, United States: ACM.

Nielsen, J. (2005). *Severity Ratings for Usability Problems*. Retrieved from http://www.useit.com/papers/heuristic/ severityrating.html

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 249–256). Seattle, Washington, United States: ACM.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *Proc. CHI '92* (pp. 373–380). Monterey, California, United States: ACM.

Norman, D. A. (1991). Cognitive artifacts. *Designing interaction: Psychology at the human-computer interface*, 17–38.

Norman, D. A., & Draper, S. W. (1986). User Centered System Design; New Perspectives on Human-Computer Interaction (pp. 31–61). Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Olson, G. M., & Moran, T. P. (1998). Commentary on "Damaged Merchandise?" *Human-Computer Interaction*, *13*(3), 263–323.

Penn, J. (2009). *Market Overview: IT Security In 2009*. Forrester Research.

Pinelle, D., Wong, N., & Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design. In *Proc. CHI '08* (pp. 1453–1462). Florence, Italy: ACM.

Rabardel, P., & Bourmaud, G. (2003). From computer to instrument system: a developmental perspective. *Interacting with Computers*, *15*(5), 665 – 691.

Rogers, Y. (1992). Ghosts in the network: distributed troubleshooting in a shared working environment. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work* (pp. 346–355). Toronto, ON, Canada: ACM.

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

Sarbanes, P. (2002). Sarbanes-Oxley Act of 2002. In *The Public Company Accounting Reform and Investor Protection Act. Washington DC: US Congress*.

Scholtz, J., & Consolvo, S. (2004). Toward a framework for evaluating ubiquitous computing applications. *Pervasive Computing, IEEE*, *3*(2), 82 –88.

Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Shneiderman, B. (2000). Creating creativity: user interfaces for supporting innovation. *ACM Trans. Comput.-Hum. Interact.*, *7*(1), 114–138.

Siegel, D. A., Reid, B., & Dray, S. M. (2006). IT Security: Protecting Organizations In Spite of Themselves. *Interactions*, 20–27.

Somervell, J. (2004). *Developing heuristic evaluation methods for large screen information exhibits based on critical parameters* (PhD Dissertation). Virginia Polytechnic Institute and State University.

Sutcliffe, A., & Gault, B. (2004). Heuristic evaluation of virtual reality applications. *Interacting with Computers*, *16*(4), 831 – 849.

Te'eni, D., Carey, J., & Zhang, P. (2007). *Human Computer Interaction: developing effective organizational information systems*. Wiley.

Thompson, R. S., Rantanen, E. M., Yurcik, W., & Bailey, B. P. (2007). Command line or pretty lines?: comparing textual and visual interfaces for intrusion detection. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (p. 1205). San Jose, CA, USA: ACM.

Velasquez, N. F., & Durcikova, A. (2008). Sysadmins and the need for verification information. In *CHiMiT '08: Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology* (pp. 1–8). San Diego, California: ACM.

Velasquez, N. F., & Weisband, S. P. (2008). Work practices of system administrators: implications for tool design. In *CHiMiT '08: Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology* (pp. 1–10). San Diego, California: ACM.

Vicente, K. J. (2000). HCI in the global knowledge-based economy: designing to support worker adaptation. *ACM Trans. Comput.-Hum. Interact.*, *7*(2), 263–280.

Vredenburg, K., Mao, J.-Y., Smith, P. W., & Carey, T. (2002). A survey of user-centered design practice. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 471–478). Minneapolis, Minnesota, USA: ACM.

Werlinger, R., Hawkey, K., & Beznosov, K. (2009). An Integrated View of Human, Organizational, and Technological Challenges of IT Security Management. *Journal of Information Management & Computer Security*, *17(1)*, 4–19.

Werlinger, R., Hawkey, K., Botta, D., & Beznosov, K. (2009). Security Practitioners in Context: Their Activities and Interactions with Other Stakeholders Within Organizations. *International Journal of Human-Computer Studies*, *67*(7), 584–606.

Werlinger, R., Hawkey, K., Muldner, K., & Beznosov, K. (2009). Towards Understanding Diagnostic Work During the Detection and Investigation of Security Incidents. In *Proceedings of HAISA: Human Aspects of Information Security and Assurance* (pp. 119–132). Athens, Greece.

Werlinger, R., Hawkey, K., Muldner, K., Jaferian, P., & Beznosov, K. (2008). The Challenges of Using an Intrusion Detection System: Is It Worth the Effort? In *Proceedings of the 4th Symposium On Usable Privacy and Security (SOUPS)* (pp. 107–116). Pittsburgh, PA.

Zager, D. (2002). Collaboration as an activity coordinating with pseudo-collective objects. *Computer Supported Cooperative Work (CSCW)*, *11*(1), 181–204.

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

Zhou, A. T., Blustein, J., & Zincir-Heywood, N. (2004). Improving Intrusion Detection Systems through Heuristic Evaluation. In *in IEEE Canadian Conf. on Electrical B. and Computer Engineering (CCECE)* (pp. 1641 – 1644).