Demonstrating the Need for Usability Assessment within Software Development Standards

Emily Rickel, Barbara S. Chaparro
Embry-Riddle Aeronautical University
Daytona Beach, FL
rickele@my.erau.edu, barbara.chaparro@erau.edu

Mitchell J. Tindall, Beth F. Wheeler Atkinson, Emily Anania
Naval Air Warfare Center Training Systems Division
Orlando, FL
mitchell.tindall@navy.mil, beth.atkinson@navy.mil, emily.c.anania1@navy.mil

ABSTRACT

Despite the availability of numerous usability assessment methods, current software development practices often lack a user-centered design approach. However, early and continued implementation of usability methods in the software development process can yield a significant return on investment by reducing the resources and manpower required to address usability issues, minimizing maintenance costs and training requirements, and increasing user efficiency and satisfaction (Nielsen, 1993b; Ohnemus, 1996; Rajanen, 2003). Heuristic evaluation and user testing are two methods used to assess the usability of a system. Heuristic evaluation involves assessing an interface against general usability standards (Nielsen & Molich, 1990). Typically performed by individuals with a human factors background, results from heuristic evaluation may not capture data relevant to the background and expertise of end-users. User testing fills this gap by collecting feedback directly from end-users as they complete representative scenario-based tasks using the interface under evaluation. These complementary methods can produce unique results and influence interface design from different perspectives to create a more comprehensive evaluation (Tan, Liu, & Bishu, 2009). Currently in development, Workbench is a web-based interface that will be integrated with the Post Mission Assessment for Tactical Training and Trend Analysis (PMATT-TA) software suite currently embedded in P-8 Weapons Tactics Trainers (WTTs). PMATT-TA supports anti-submarine warfare (ASW) training assessments by providing a central location for the collection, aggregation, and visualization of ASW measures of performance (MOPs). Workbench aims to establish a more efficient process for training instructors to update and create MOPs without the assistance of a software engineer. This paper demonstrates the use of heuristic evaluation and user testing as they apply to the Workbench case study. The techniques and accompanying results are compared and provide insight into the need for standardized, iterative, user-centered design software development processes when creating training technologies.

ABOUT THE AUTHORS

Emily Rickel is a doctoral student currently pursuing her PhD in Human Factors at Embry-Riddle Aeronautical University (ERAU). Her research interests include interface design and analysis, usability/user experience, and communication, particularly as they apply to military systems, consumer products, and healthcare. Through her internships at the Naval Undersea Warfare Center (NUWC, Newport) and the Naval Air Warfare Center Training Systems Division (NAWCTSD), she has experience applying human-computer interaction principles and physiological considerations to the development and assessment of Navy training systems. She is a recipient of the 2019-2020 RADM Fred Lewis I/ITSEC Postgraduate Scholarship. Emily received her Master of Science in Human Factors from ERAU.

Barbara S. Chaparro has a PhD in Experimental Psychology from Texas Tech University. She is a Professor in the Human Factors and Behavioral Neurobiology Department and head of the Research in User eXperience (RUX) Lab at Embry-Riddle Aeronautical University**.** Her research interests include the study of factors that influence the user experience (UX) of products, software and systems, the investigation of usability assessment methods, and the efficacy of mobile computing devices.

Mitch Tindall, PhD, is a Research Psychologist at NAWCTSD in the Basic and Applied Training and Technology for Learning and Evaluation (BATTLE) Laboratory. During his graduate work, he performed research and consulting

in human performance assessment and productivity enhancement. His work for the Navy includes several areas such as psychometric development and validation, human performance measurement, human-computer interaction, data management and analytics, and training systems enhancement and validation. His research interests include human performance measurement, physiological episode training enhancement, and data science. His PhD is in Industrial-Organization (I-O) Psychology from University of Central Florida (UCF).

Ms. Beth F. Wheeler Atkinson is a Senior Research Psychologist at NAWCTSD and a NAVAIR Associate Fellow. She has led several research and development efforts devoted to investigating capability enhancements for training and operational environments, and has successfully transitioned a post-mission reporting and trend analysis tool that leverages automated performance measurement technology. Her research interests include instructional technologies (e.g., performance measurement, post-mission reporting/review), human-computer interaction (HCI)/user interface design and analysis, and aviation safety training and operations. She holds an MA in Psychology, Applied Experimental Concentration, from the University of West Florida (UWF).

Emily Anania, PhD, is a Research Psychologist at NAWCTSD in the BATTLE Laboratory. Her PhD is in Human Factors from Embry-Riddle Aeronautical University. Her research interests include human-automation interaction, human factors analyses of training systems, and aviation human factors.

Demonstrating the Need for Usability Assessment within Software Development Standards

Emily Rickel, Barbara S. Chaparro
Embry-Riddle Aeronautical University
Daytona Beach, FL
rickele@my.erau.edu,
barbara.chaparro@erau.edu

Mitchell J. Tindall, Beth F. Wheeler Atkinson, Emily Anania
Naval Air Warfare Center Training Systems Division
Orlando, FL
mitchell.tindall@navy.mil, beth.atkinson@navy.mil,
emily.c.anania1.ctr@navy.mil

BACKGROUND

Usability refers to the "extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (ISO 9241-210:2019). The term is an overarching, multi-faceted concept affected by the relationship between a system (e.g., a software program), its tasks, and its users. Traditionally, usability is associated with the following five attributes: 1) learnability, the system should be easy for the user to learn; 2) efficiency, the system should allow the user to perform at a high level of productivity; 3) memorability, the system should be easy to remember such that a user who spends time away from the system does not have to completely relearn the system upon return; 4) errors, the system should prevent users from making errors and should provide users methods by which to easily recover from errors if they occur; and, 5) satisfaction, the system should be enjoyable and pleasant for users to interact with (Nielsen, 1993a). While there are several usability assessment methods available for implementation in the software development process, their use is often dismissed due to misconceptions that the resources expended to implement these methods do not yield a return on investment. The purpose of this paper is to demonstrate the importance of usability assessments as they apply to software development standards that impact human-computer interaction. That is, strict adherence to standards without considering the impact on the system's front-end can have a deleterious impact on the system's overall usability. As a use case, this paper provides an overview of applicable standards and usability methods used to evaluate a software for developing automated performance measures (APMs) in simulation-based anti-submarine warfare (ASW) training. We will present results from two different usability approaches used to assess this ASW training APM system to illustrate the role usability assessment plays in standard implementation.

NEED FOR USABILITY METHODS IN SOFTWARE DEVELOPMENT

Current Software Development Standards

Historically, military software development and documentation standards were outlined in MIL-STD-498 (1994). These standards imposed very little emphasis on usability, offering a limited definition of the concept, few suggestions for how to implement a user-centered design focus in software development, and little to no references to where guiding usability principles could be found (e.g., MIL-STD-1472D: Human Engineering, Design Criteria for Military Systems, Equipment, and Facilities). In 1998, MIL-STD-498 was superseded by ISO/IEC/IEEE 12207 (International Organization for Standardization / International Electrotechnical Commission / Institute of Electrical and Electronics Engineers; Bourque & Fairley, 2014; MIL-STD-498 NOTICE 1, 1998). ISO/IEC/IEEE 12207 is an international standard that provides a framework for the processes and activities associated with all stages of the software life cycle, from acquisition to disposal. While the current standards include more references to usability compared to MIL-STD-498, they do so with passive verbiage that does not emphasize the criticality of usable systems. Therefore, it is easy for those who use these standards to overlook or dismiss the need for implementing user-centered design practices. Additionally, usability processes are typically isolated to the requirements gathering and validation stages, when implementation throughout the process would be more effective. Another limitation of ISO standards are the upfront costs required to review them, creating an obstacle to their accessibility, particularly when usability guidelines are spread across multiple documents (e.g., ISO 9241, ISO/IEC 25060). All these factors play a role in disregarded or deprioritized usability-focused standards due to tradeoff decisions when minimal resources are available, which can result in a heavy focus on functional requirements rather than the system usability.

Another set of standards that impacts the current effort addressed in this paper is the development of the Human Performance Modeling Language (HPML). The purpose of HPML is to transform data collected from training and operational environments into performance measurements and assessments. HPML makes up the foundation of the Workbench software program, a product to be integrated with the Post Mission Assessment for Tactical Training and Trend Analysis (PMATT-TA) software suite currently embedded in P-8 Weapons Tactics Trainers (WTTs). A Product Development Group within the Simulation Interoperability Standards Organization (SISO) is currently establishing standards associated with HPML. The proposed SISO standard directs that HPML code be both machine- and human-readable, but it provides no guidance related to the usability of the code.

Benefits of Early and Continued Implementation of Usability Methods

In general, usability assessment methods provide empirical, data-driven insight into how the system is actually used, as opposed to how it is assumed to be used. Additionally, this approach facilitates the retrieval of feedback about the system from individuals other than the system's designers and engineers (Mills et al., 1986). This is important because designing systems based solely on assumptions and developer schemas creates an insufficient picture of user interaction requirements with little consideration for the end-users' background, the tasks they need to complete, or the environment and constraints in which they will interact with the system. If users think the system does not support their goals, they are less likely to use the system, or will use it inefficiently. Thus, usability is critical for users' acceptance of the system (Ferre, Juristo, Windl, & Constantine, 2001).

There are several methods (e.g., heuristic evaluation, user testing) that can be employed to assess and improve the usability of a system throughout the duration of the system's development life cycle. Despite the availability of these methods, many obstacles prevent them from implementation throughout the software development process. For one, there is stigma surrounding the efficiency of performing usability assessments. Many view usability assessments to be slow and cumbersome, but short, iterative testing can prove to be both comprehensive and timely (Mills et al., 1986). Additionally, many usability techniques can be adapted or abbreviated to help stakeholders remain on schedule. For example, low-fidelity prototypes, such as paper prototypes, are cost-effective mockups used to quickly conceptualize an interface (Sauer & Sonderegger, 2009). A similar timeliness concern is the potential delay of releasing the product due to time expended to perform usability assessments. Such a delay can create significant issues in financial budget constraints and market competitiveness. However, delays in product release can be circumvented by producing both short- and long-term recommendations in order to provide guidance as to which issues should be addressed immediately (i.e., within the established timeline) and which issues can wait to be resolved before the next version's release (Rosenbaum, 1989). Another challenge to effectively including usability assessment methods during the software development process is the notion that utilization of these methods must wait until an interface is produced. However, doing so can mean that it is too late to implement valuable findings obtained from usability methods (Mills et al., 1986). Instead, usability assessment methods should be implemented early in the software development process (i.e., when establishing requirements) and should maintain a notable role throughout the product's life cycle.

The early and continued implementation of usability methods in the software development process is not only critical and feasible, but it also yields measurable benefits and a significant return on investment. According to Nielsen (1993b), more-usable systems are associated with increased productivity and satisfaction, as well as decreased training time. To quantify these measures, increased satisfaction can make a user's job more pleasant, resulting in decreased absenteeism and turnover. Decreased training time also translates to cost-savings, as every hour of training means less time being productive and potentially more time paying an instructor (Nielsen, 1993b). Increased usability is also tied to reduced support and maintenance costs (Ferre et al., 2001; Nielsen, 1993b; Ohnemus, 1996). For example, research shows that calls to technical support can range from $12-250 per call (Ohnemus, 1996), which is approximately equivalent to $20-415 per call in 2020 (U.S. Bureau of Labor Statistics, n.d.). This financial loss is in addition to the time expended as a result of the user's inability to perform their tasks efficiently while getting assistance. Ohnemus (1996) also cites that 80% of maintenance is the result of unmet or unforeseen user requirements, and that 80% of software life cycle costs are expended during the maintenance phase. More recent estimates suggest that software maintenance cost is growing, accounting for about 90% of the software life cost (Dehaghani & Hajrahimi, 2013). Thus, usability issues addressed earlier in the life cycle can result in significant savings later. Finally, for commercial products, systems with higher perceived usability are more competitive than other products on the market. This increased competitiveness can result in more sales and repeat customers (Ferre et al., 2001; Nielsen, 1993b; Ohnemus, 1996; Rajanen, 2003).

In summary, while current software development processes do not explicitly advocate the early and continued implementation of usability assessment methods, doing so is feasible, cost-effective, and critical for user acceptance.

COMPLEMENTARY USABILITY METHODS: HEURISTIC EVALUATION AND USER TESTING

Heuristic evaluation and user testing are two popular methods used to evaluate the usability of a system. The popularity of these methods is likely attributed to how easy they are to implement in product development life cycles. Nielsen (1993a) dubs heuristic evaluation and user testing as "discount usability engineering" methods because their simplicity facilitates their use in practical design situations that must adhere to budget and time constraints.

Heuristic Evaluation

Heuristic evaluation involves analyzing an interface through systematic inspection to assess its compliance with recognized usability principles (i.e., heuristics) (Nielsen, 1992; Nielsen, 1993a). Examples of popular heuristics include the 10 Usability Heuristics for User Interface Design (Nielsen, 1994a) and the Eight Golden Rules of Interface Design (Shneiderman, 1998). Results from heuristic evaluations include a list of usability principles violated, along with an output of specific violations from the system. To achieve best results, a team of three to five evaluators should independently assess the system and then come together to aggregate their findings. This process prevents bias and increases the likelihood that more usability issues are found. Under ideal circumstances, evaluators should have a background in usability, such as usability specialists or human factors practitioners. Common practice is to inspect the entire system at least two times; once to get a general feel of the program, and twice to focus on interaction with specific interface elements. A heuristic evaluation typically lasts one to two hours, but longer sessions may be necessary for complex interfaces (Nielsen, 1994b).

Two of the primary benefits associated with heuristic evaluation are the flexibility and comprehensiveness it provides. Evaluators are able to interact with and assess all aspects of the system without being limited by a scenario-based interaction. This method can also be applied (and reapplied, for iterative testing) during many stages of the software development life cycle. Compared to other usability assessment methods, heuristic evaluation has shown it is able to identify more problems within a system, but, these problems are often lower-priority and less critical to an end-user's successful interaction with the system (Jeffries, Miller, Wharton, & Uyeda, 1991; Tan, Liu, & Bishu, 2008). Adding to its flexibility, heuristic evaluations can be performed by individuals with little to no usability expertise (Nielsen, 1993a); however, this can also serve as a limitation, as the quality of results depend upon the expertise of the evaluator. Nielsen (1992) compared the heuristic evaluation results of three evaluator groups: novices (i.e., evaluators with no usability experience), single experts (i.e., usability specialists), and double experts (i.e., usability specialists who also had experience with the interface under evaluation). He found that double experts were able to find more problems with fewer evaluators compared to single experts and novices. If double experts are not available, one way to circumvent this limitation is to have individuals with interface or domain expertise provide usability specialists with a walkthrough of the system before the evaluators complete their assessment. Furthermore, another limitation of heuristic evaluation is that it requires multiple evaluators in order to achieve best results (Jeffries et al., 1991; Nielsen, 1992; Nielsen, 1993a). Averaged across six projects, Nielsen (1993a) found that single evaluators only found 35% of all known usability problems. Additional evaluators translates to use of more resources, but doing so results in a more comprehensive report of usability issues.

A tool that leverages the heuristic evaluation method is the Experience-based Questionnaire for Usability Assessments Targeting Elaborations (EQUATE). The EQUATE was designed as a hybrid approach to usability, combining heuristic and survey methods in order to improve end-user feedback, increase efficiency, and develop a method that can adapt to the functionality of the system being evaluated. The EQUATE's heuristic basis comes from a synthesis of evaluation approaches (such as Nielsen & Molich, 1990), resulting in the Multiple Heuristic Evaluation Table (MHET; Wheeler Atkinson, Bennett, Bahr, & Walwanis Nelson, 2007). The MHET provides general heuristic rules of thumb, guidelines, and specific dos and don'ts that support practitioners in usability testing and design. However, the MHET initially focused on software graphical user interfaces (GUIs). In order to expand and advance the MHET, researchers performed a qualitative analysis of extant heuristic evaluations and relevant usability guidelines, as well as identifying of gaps in the MHET, with the purpose of providing a more robust evaluation. The updated tool was

entitled User Interface – Table for Evaluating and Applying Composite Heuristics (UI-TEACH), and the MHET's 12 heuristic categories were modified, resulting in 17 total categories on the UI-TEACH.

The EQUATE was developed from the 17 UI-TEACH categories, using an open and closed card sort to properly distill and organize the heuristic categories. The card sort allowed developers to gain an understanding of how other users (i.e., sample population of potential entry-level individuals) would organize the usability guidance under specific heuristic categories (where participants chose the category label). The card sort, as well as validity and reliability testing, resulted in nine final heuristic categories: Error Handling & Feedback, Graphic Design & Aesthetics, User Interaction Control, Memorability & Cognitive Facilitation, User Efficiency, Learnability, Consistency, and Help, that were developed into a usability questionnaire. The EQUATE is uniquely suited to give feedback on any system, in any part of the design process. Ideally, this allows practitioners to use this tool where others may not be applicable. It is in-depth, and targets elaboration of each heuristic in order to gather specific participant feedback. Often, simple scales like the System Usability Scale (SUS; Brooke, 1996) are quick and useful tools, but may result in limited information and limited capacity to diagnose issues. For example, in the early development stages of software, the SUS may be unable to identify targeted spaces for improvement, and may instead simply result in a low score on the short scale. The EQUATE is uniquely qualified to gauge system performance and usability in an in-depth manner, allowing users to not only rate the system, but also provide targeted feedback for system improvement.

User Testing

The goal of user testing is to collect feedback directly from representative end-users as they interact with the system under evaluation by completing scenario-based tasks (Dumas & Redish, 1999). User testing typically involves running participants unfamiliar with the system through a series of tasks with the interface. These tasks are carefully designed by usability specialists to encourage interaction with critical aspects of the interface that are expected to be most used or are expected to be associated with usability issues. As participants complete each task, their performance (e.g., success rate, efficiency, time to complete each task) and comments (e.g., likes/dislikes, points of confusion, perceived difficulty) are recorded by an observing usability specialist, who acts as a facilitator during the testing session. The facilitator's role is to provide participants with tasks and encourage their feedback throughout the session. Facilitators do not assist participants in completing the tasks. Additionally, facilitators are trained to ensure that users are not swayed into providing specific feedback that does not represent their independent, candid comments (Nielsen, 1993a). Quantitative and qualitative data gathered during user testing is used to diagnose the system's usability problems and to provide recommendations to address those problems.

The primary benefit of user testing is the ability to observe and record real interactions between users and the system interface, which help to uncover usability issues. Such results have a large impact on the system's stakeholders, as seeing users become frustrated with or confused by their system carries a lot of weight and incites the need to address the source of those negative experiences. Additionally, user testing is extremely flexible. The type and number of tasks and measures (i.e., the quantitative and qualitative data gathered during the study) can be customized to suit the project's needs and constraints. Limitations of user testing include the amount of resources required to recruit, run, and compensate participants. Also, user testing is more expensive and time-consuming compared to other usability assessment methods (Jeffries et al., 1991). Furthermore, some argue that user testing has limited impact when implemented early in the software development life cycle and is best applied to later stages (Jeffries & Desurvire, 1992). However, others argue that iterative user testing should be performed as early and as frequently as possible in the development life cycle, as doing so can call attention to issues before addressing them involves substantial cost (Genov, 2005). Additionally, iterative user testing provides an opportunity to assess the implementation of changes made to the interface, which can serve to evaluate the reliability of the results of earlier user testing sessions (Karat, 1989). Finally, the comprehensiveness of this method has been questioned, as the results are limited to the scenario-based tasks completed; it is unlikely that all interactions with every feature of the program can be tested in a single user test (Tan, Liu, & Bishu, 2008).

Comparing Heuristic Approaches to User Testing

What is the difference in the outcomes of heuristic evaluation and user testing? Is one better than the other? As is true with all usability assessment methods, the application of multiple evaluation techniques results in the best evaluation of a system (Jeffries & Desurvire, 1992). Leveraging multiple methods helps mitigate the limitations of individual methods and provides a more comprehensive assessment of the system. For example, heuristic evaluation can identify

a greater number of problems compared to user testing (Tan, Liu, & Bishu, 2008), but the types of issues uncovered during user testing are often more severe, more recurring, and more directly related to user performance and acceptance (Bailey, Allan, & Raiello, 1992; Jeffries et al., 1991). Additionally, heuristic evaluation uncovers issues that users *may* struggle with, while user testing sheds light on issues that with which users *actually* struggle. While the ability to predict usability issues is beneficial, system developers may not prioritize the implementation of *potential* problems gathered from heuristic evaluation as much as the results obtained from user testing (Bailey, Allan, & Raiello, 1992; Dumas & Redish, 1999). However, heuristic evaluation is a cost-effective and valuable method that can be used early in the development life cycle, or when available time and resources are limited (Jeffries & Desurvire, 1992). Table 1 shows a comparison of the two methods in terms of resources required and recommended timing.

	Heuristic Evaluation	**User Testing**
Cost	Less	More
Time Required	Less	More
Type of Feedback	Predictive (identifies *potential* user issues)	Confirmative (identifies *actual* user issues)
Early Life Cycle	✓	
Middle Life Cycle	✓	✓
Late Life Cycle	✓	✓

Table 1. Comparing the advantages and disadvantages of heuristic evaluation and user testing.

In summary, heuristic evaluation and user testing are two unique methods that seek to address the same goal of improving system usability. These two methods should be considered complementary; neither can replace the other and both should be used to reap optimal benefits. For best practice, usability assessment methods should be employed multiple times throughout the software development process. Iterative implementation of heuristic evaluation should begin during early stages of the development process, while user testing should be iteratively performed during later stages (Tan, Liu, & Bishu, 2008). As Genov (2005) cites, "it is better to have several small tests that span the length of the development process than fewer larger tests towards the end."

APPLYING AND COMPARING USABILITY ASSESSMENT TECHNIQUES WITHIN THE CONTEXT OF AN ASW TRAINING PROGRAM INTERFACE

The following sections aim to assess prior literature's recommendation to employ the complementary methods of heuristic evaluation and user testing. This assessment will be applied to the evaluation of an ASW training technology interface that is currently under development.

Program Overview: Workbench

Workbench is a web-based, simulator-integrated application designed to be implemented within the PMATT-TA software suite. PMATT-TA is an existing program in P-8 WTTs that supports ASW training assessments by providing a central location for the collection, aggregation, and visualization of ASW measures of performance (MOPs). Currently, creating and modifying MOPs using PMATT-TA necessitates a software engineer to hardcode such updates into the system, a time-consuming and costly process. Workbench aims to provide creation and modification capabilities to training community end-users by creating a more efficient process for updating MOPs without the assistance of a software engineer. Individuals expected to utilize Workbench include personnel with ASW domain and training expertise, personnel with software programming expertise, and personnel with overlapping backgrounds that include ASW, training, and programming expertise. The same version of Workbench was evaluated using heuristic evaluation and user testing (i.e., heuristic evaluation evaluators and user testing participants interacted with the same interface).

Leveraging Complementary Usability Methods to Evaluate Workbench

To perform a heuristic evaluation of Workbench, three evaluators (i.e., two Research Psychologist interns and one Aerospace Engineer intern) viewed a demonstration of Workbench that was led by the engineers and stakeholders overseeing the program's software development. After the demonstration, the evaluators independently assessed

Workbench using the EQUATE. Then, the evaluators came together to discuss their independent findings and compile them into a comprehensive report of known usability issues. These issues were given criticality scores based on the severity of the issue to assist the developers' understanding of higher- and lower-priority areas of improvement. Higher-priority issues included hidden functions (e.g., keyboard and mouse interactions that were not intuitive, program-critical menus that were difficult to find) and lack of embedded help materials (e.g., tutorial, on-hover tool tips, descriptions of program elements) to facilitate the system's learnability. Lower-priority issues included minor interaction (e.g., activation of unexpected cursor when hovering over a clickable button) and aesthetic issues (e.g., buttons with the same appearance and placed next to one another but provide different functions).

Remote user testing was completed to evaluate Workbench. First, a research team composed of human factors practitioners viewed a demonstration of Workbench to get a better understanding of the program's purpose, functions, and intended users. Based on this demonstration, the research team developed a series of scenario-based tasks that reflected actions expected to be performed by the system's end-users, as well as actions that were expected to be associated with usability issues that were uncovered by the heuristic evaluation. These tasks underwent several iterative modification cycles to ensure they encouraged interaction with a range of features and were representative of the program's target audience. This process resulted in ten tasks provided to participants during remote user testing sessions. During these sessions, participants used a computer to connect to a virtual Microsoft Teams meeting with at least two members of the research team. One research team member facilitated the session by controlling the pace of provided tasks and by prompting users to provide clarifying information about their perceptions of the program, while the other took notes on the participants' comments and experience. Participants were asked to enable the screen-sharing feature of Microsoft Teams so the research team members could observe their interactions with Workbench. Participants were also asked to verbalize their thought process as they completed each task (i.e., think-aloud protocol). Additionally, after each task, participants rated how difficult they perceived the task to be and how confident they felt that they successfully completed the task. After all of the tasks were completed, participants were prompted to elaborate on overall likes, dislikes, and modifications they would make to the program. Participants also were asked to sum up their perceptions of Workbench by choosing three adjectives from an abbreviated list of the Microsoft Product Reaction Cards (Benedek & Miner, 2002). Nine participants completed remote user testing sessions. Each session lasted approximately 60 to 90 minutes. Four participants were active duty U.S. Navy officers with ASW experience, three were Department of Defense (DoD) civilians or contractors with backgrounds in human factors, and two were DoD civilians with backgrounds in computer science and engineering.

Comparison of Results from Heuristic Evaluation and User Testing

One interesting difference between the results collected from the heuristic evaluation and user testing sessions is the type of feedback gathered from each method (see Table 2 for a side-by-side comparison of select usability recommendations that were identified through each method). Heuristic evaluation results revealed issues with the interaction and organization of the interface (e.g., visibility of controls and menus, aesthetic traits). User testing uncovered some of the same issues, but also shed light on issues related to the features and operations currently provided or desired by the system (e.g., desired ability to test-run MOPs to confirm the logic is running as expected). These were reported to be higher priority to the participants and more influential on user trustworthiness and acceptance. The difference in heuristic evaluation and user testing results can be attributed to the fact that user testing participants were more representative of the software program's target population compared to the evaluators in the heuristic evaluation and as such, were able to focus more on the software functionality and related ease of use.

Next steps of this case study include presenting results of the heuristic evaluation and user testing to the program's stakeholders and software developers. We will continue to maintain contact with these individuals as they make changes to the system, acting as a representative for the program's users to ensure updates address key issues uncovered by both usability assessment methods. Once the system is updated based upon the usability recommendations, we plan to perform further iterations of usability assessment to evaluate the implemented changes and to obtain more feedback that will be valuable in transitioning the program for use by end-users. Future efforts will also focus on assessing and improving the system's help documentation, as well as on providing guidance to developing effective help materials (e.g., an embedded program tutorial) that promote the system's learnability and memorability.

Description of Select Usability Recommendations	Identified through Heuristic Evaluation	Identified through User Testing
Activate the hand cursor when user hovers over a clickable button	✓	✓
Provide on-hover tool tips to facilitate on-the-job training	✓	✓
Modify property naming conventions to enhance efficiency	✓	
Include redundant buttons for frequently-used controls	✓	
Devote a section of the interface to serve as an informational panel that constantly displays content for a selected MOP attribute		✓
Develop a user manual or workbook with example MOPs and how to build them using Workbench		✓
Include ability to test-run MOPs to confirm logic runs as expected		✓

Table 2. Usability recommendations and method of identification (i.e., heuristic evaluation and/or user testing).

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE EFFORTS

The Workbench case study corroborates prior literature by demonstrating that heuristic evaluation and user testing are complementary assessment methods that identify usability issues (i.e., issues related to the program's learnability, efficiency, memorability, errors, and satisfaction). Our heuristic evaluation uncovered usability issues associated with the program's controls and aesthetics, some of which were confirmed by user testing to pose difficulties to participants. User testing also provided insight into key issues related to the program's features, which proved to have significant implications on reported user acceptance and trustworthiness. Similar to what has been previously published (e.g., Bailey, Allan, & Raiello, 1992; Jeffries et al., 1991; Tan, Liu, & Bishu, 2008), our case study found that the heuristic evaluation method identified many usability issues, but these issues were more superficial in relation to the critical issues discovered by user testing. However, results from the heuristic evaluation should not be discounted because our case study found that user testing confirmed issues found during the heuristic evaluation, indicating that heuristic evaluation produces strong predictions of user issues with fewer resources compared to user testing. Additionally, the evaluators' human factors background and their access to usability heuristics and the EQUATE prompts during their heuristic evaluations facilitated their ability to uncover a wide-range of issues, resulting in a more comprehensive assessment. Thus, both heuristic evaluation and user testing should be implemented throughout the duration of the software development life cycle in order to reap the benefits of both usability assessment methods.

Future policy and standard efforts surrounding the software development life cycle should advocate and provide guidance for successfully employing the early and continued use of usability assessment methods. In particular, standards groups should evaluate whether or not the implementation of a standard may impact front-end usability and determine what type and level of usability assessment should be applied to assess and improve system usability. Due to rapid advancements in technology and software becoming more prominent and complex in a wider variety of systems, the implementation of policy and standards to encourage consistent use of usability assessment methods may serve to assist with fiscal challenges that will emerge if software maintenance cost continue to grow (e.g., Dehaghani & Hajrahimi, 2013).

ACKNOWLEDGEMENTS

The views expressed herein are those of the authors and do not necessarily reflect the official position of the DoD or its components. This study was conducted in collaboration with Embry-Riddle Aeronautical University under an Education Partnership Agreement. This effort involves research and development sponsored by the Naval Aviation Training Systems and Ranges Program Office (PMA-205 Air Warfare Training Development program, NAVAIR Naval Innovative Science and Engineering (NISE) program, the Maritime Patrol and Reconnaissance Aircraft Program Office (PMA-290), and the Small Business Innovative Research/Small Business Technology Transfer (SBIR/STTR) program. The authors appreciate the support and participation by our end-user community at NAS Jacksonville. NAWCTSD Public Release 20-ORL038 Distribution Statement A – Approved for public release; distribution is unlimited.

REFERENCES

Bailey, W. A., Knox, S. T., & Lynch, E. F. (1988). Effects of interface design upon user productivity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 207-212).

Benedek, J., & Miner, T. (2002). Measuring desirability: New methods for evaluating desirability in a usability lab setting. In *Proceedings of the UPA 2002 Conference,* Orlando, Florida, USA.

Bourque, P., & Fairley, R. E. (Ed.). (2014). *Guide to the software engineering body of knowledge*. IEEE.

Brooke, J. (1996). SUS: a 'quick and dirty' usability scale. In P.W.Jordan, B. Thomas, B.A. Weerdmeester, and I.L. McClelland (Eds.) *Usability Evaluation in Industry* (189-194). London: Taylor and Francis.

Dehaghani, S. M. H., & Hajrahimi, N. (2013). Which factors affect software projects maintenance cost more? *Acta Informatica Medica, 21*(1), 63-66.

Department of Defense Military Standard. (1994). Software development and documentation (Standard No. MIL-STD-498).

Department of Defense Military Standard. (1998). Software development and documentation (Standard No. MIL-STD-498 Notice 1).

Dumas, J. S., & Redish, J. (1999). *A practical guide to usability testing*. Portland, OR: Intellect Books.

Ferré, X., Juristo, N., Windl, H., & Constantine, L. (2001). Usability basics for software developers. *IEEE Software, 18*(1), 22-29.

Genov, A. (2005). Iterative usability testing as continuous feedback: A control systems perspective. *Journal of Usability Studies, 1*(1), 18-27.

International Organization for Standardization. (2017). *Systems and software engineering – Software life cycle processes* (ISO/IEC/IEEE Standard No. 12207).

International Organization for Standardization. (2019). *Ergonomics of human-system interaction – Part 210: Human-centered design for interactive systems* (ISO Standard No. 9241-210).

Jeffries, R., & Desurvire, H. (1992). Usability testing vs. heuristic evaluation: Was there a contest? *ACM SIGCHI Bulletin, 24*(4), 39-41.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. (1991). User interface evaluation in the real world: A comparison of four techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 119-124).

Karat, C. M. (1989). Iterative usability testing of a security application. In *Proceedings of the Human Factors Society 33rd Annual Meeting* (pp. 273-277).

Mills, C., Bury, K. F., Reed, P., Roberts, T., Tognazzini, B., Wichansky, A., & Gould, J. (1986, April). Usability testing in the real world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 212-215).

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 373-380).

Nielsen, J. (1993a). *Usability engineering*. Cambridge, MA: Academic Press, Inc.

Nielsen, J. (1993b, November/December). Is usability engineering really worth it? *IEEE Software, 10*, 90-92.

Nielsen, J. (1994a). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

Nielsen, J. (1994b, November). How to conduct a heuristic evaluation. *Nielsen Norman Group*. Retrieved from https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 249-256).

Ohnemus, K. R. (1996). Incorporating human factors in the system development life cycle: Marketing and management approaches. In *IPCC 96: Communication on the Fast Track. IPCC 96 Proceedings* (pp. 46-53). IEEE.

Rajanen, M. (2003). Usability cost-benefit models–different approaches to usability benefit analysis. In *Proceedings of the 26th Information Systems Research Seminar in Scandinavia* (IRIS26), Haikko, Finland.

Rosenbaum, S. (1989). Usability evaluations versus usability testing: When and why? *IEEE Transactions on Professional Communication, 32*(4), 210-216.

Sauer, J., & Sonderegger, A. (2009). The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behavior, subjective evaluation and emotion. *Applied Ergonomics, 40*(4), 670-677.Shneiderman, B. (1998). *Designing the user interface: Strategies for effective human-computer interaction.* Reading, MA: Addison Wesley Longman.

Tan, W. S., Liu, D., & Bishu, R. (2009). Web evaluation: Heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics, 39*(4), 621-627.

U.S. Bureau of Labor Statistics. (n.d.). *CPI Inflation Calculator*. https://www.bls.gov/data/inflation_calculator.htm

Wheeler Atkinson, B. F., Bennett, T. O., Bahr, G. S., & Walwanis Nelson, M. M. (2007, July). Development of a multiple heuristics evaluation table (MHET) to support software development and usability analysis. In *International Conference on Universal Access in Human-Computer Interaction* (pp. 563-572). Springer, Berlin, Heidelberg.

Wheeler Atkinson, B. F., Tindall, M. J., & Igel, G. S. (2015) Validated Usability Heuristics: Defining Categories and Design Guidance. In: Stephanidis C. (eds) *HCI International 2015 - Posters' Extended Abstracts*. HCI 2015. Communications in Computer and Information Science, vol 528. Springer, Cham.