

Early Usability Evaluation to Enhance User Interfaces – A Use Case on the Yggdrasil Cybersecurity Mockup –

Denis Iorga

Interdisciplinary School of
Doctoral Studies in University
of Bucharest

26-46 Mihail Kogalniceanu,
050107, Bucharest, Romania
denis.iorga@drd.unibuc.ro

Octavian Grigorescu

CODA Intelligence SRL

60, Mircea Vulcanescu, 010813
Bucharest, Romania
octavian.grigorescu@
codaintelligence.ro

Mihai Predoiu

CODA Intelligence SRL

60, Mircea Vulcanescu, 010813
Bucharest, Romania
mihai.predoiu@
codaintelligence.ro

Cristian Sandescu

CODA Intelligence SRL

60, Mircea Vulcanescu, 010813
Bucharest, Romania
cristian.sandescu@
codaintelligence.ro

Mihai Dascalu

University Politehnica of

Bucharest

313 Splaiul Independentei,
060042, Bucharest, Romania
mihai.dascalu@upb.ro

Razvan Rughinis

University Politehnica of

Bucharest

313 Splaiul Independentei,
060042, Bucharest, Romania
razvan.rughinis@upb.ro

ABSTRACT

Most usability evaluation methods are designed to be applied after the implementation of a software product. As such, there is limited research on how to conduct usability evaluation procedures before the implementation of a software product, especially on cybersecurity platforms user interfaces. For this purpose, we illustrate the early usability evaluation of a cybersecurity graphical interface. In this sense, we conduct a usability evaluation procedure on the mockup of Yggdrasil, an application centered on the early identification of vulnerabilities using open-source intelligence, more specifically Twitter posts and news websites. To achieve this, we conducted a usability inquiry procedure using the adaptation of a questionnaire designed to evaluate the attractiveness of interactive products. In addition, we conducted a usability testing procedure based on the think-aloud protocol. The results of this study enabled the timely identification of improvement opportunities for the graphical interface of the Yggdrasil graphical interface before the actual implementation.

Author Keywords

Early usability evaluation; Open-Source Intelligence;
Graphical User Interface; Cyber-security.

ACM Classification Keywords

H.5.2: Information interfaces and presentation (e.g., HCI):
User Interfaces.

DOI: 10.37789/rochi.2021.1.1.16

INTRODUCTION

Yggdrasil [12; 13] is a cybersecurity application that automatically collects and analyses open-source intelligence (OSINT) for early detection of cybernetic vulnerabilities. As such, the aim of Yggdrasil is to collect information from online sources and display relevant insights regarding new

cybernetic vulnerabilities to cybersecurity experts. Previous work [12; 13] considered data extraction techniques and machine learning models to automatically identify new vulnerabilities. In this paper, we expand on the implementation of the user interface. More specifically, we present a usability evaluation of the graphical user interface (GUI) of the Yggdrasil cybersecurity application.

The lack of usability in cybersecurity applications often translates into negative consequences on the system that the applications aim to protect [20]. In other words, the lack of usability of cybersecurity systems can leave cybernetic infrastructures exposed to cyber-attacks, thus amplifying the risks of data alteration or loss. Therefore, usability studies in the field of cybersecurity play a crucial role in enhancing the cybernetic security of digital infrastructures. However, there is limited effort to study usability of cybersecurity platforms. Most studies in this sense are preoccupied with usability of security functionalities of web services, such as permission interfaces, authentication services, or/and security APIs.

The development process of the Yggdrasil GUI is similar to that conducted for other web applications. However, 80% of the usability studies on web applications are conducted in the implementation phase of web application development [11]. Our aim was to evaluate the usability of the Yggdrasil GUI before the implementation phase of the product.

To achieve this, we start from the Human Computer Interaction (HCI) literature that abounds in frameworks aiming to facilitate the design of „interactive products to support the way people communicate and interact in their everyday and working lives” [23]. Given our purpose, we decided to adapt one such framework in order to evaluate the usability of the Yggdrasil cybersecurity GUI before the actual implementation of the product. In this sense, our approach involved an adaptation of the AttrakDiff [3]

measurement instrument for the evaluation of the Yggdrasil GUI mockup. An availability sample of 15 individuals completed our questionnaire. Additionally, we conducted a think-aloud procedure with 5 respondents to identify improvement opportunities for the Yggdrasil GUI.

The next section focuses on analyzing research in the relevant fields, such as HCI and cybersecurity. Following this, we present the evaluation method, as well as the applied usability evaluation procedures. Finally, the last sections illustrate the results and limitations of our approach.

STATE OF THE ART

In this section we present relevant research for the purpose of our study. As such, the first subsection is centered on usability as a HCI component. We review methods for usability evaluation across the HCI field. The second subsection focuses on methods for usability evaluation of cybersecurity systems. The last subsection highlights relevant early usability methods and techniques.

Usability Evaluation Methods

Usability evaluation is a crucial aspect to consider in HCI. As such, multiple usability evaluations methods (UEMs) are present in the literature. Existing techniques for usability evaluation can be classified using the following typology: testing (e.g., log file analysis, think-aloud protocol, remote testing), inspection (e.g., cognitive walkthroughs, guideline reviews, heuristic evaluation), inquiry (e.g., questionnaires, interviews, focus-groups), analytical modeling (e.g., cognitive task analysis, GOMS analysis), and simulation (e.g., information scent, Petri net models) [14].

A systematic review of UEMs used for governmental applications identifies the advantages and disadvantages of different classes of techniques such as: usability testing (automated testing, performance measurement, system usability score, think aloud protocol), inspection (heuristic evaluation), and inquiry (focus-groups, questionnaires, user feedback interviews, field observations) [17]. Automated testing is described as being cost-efficient, but limited in providing deep insights, lacking the actual interaction with a real user or the capacity to provide rich information beyond accessibility. Performance measurement is described as having a high precision and capacity to understand the users' cognitive load, but difficult to implement due to the need of potential users. Using the system usability scale is described by the authors as being resource efficient but limited in terms of the insights provided. The think aloud protocol is described as the opposite, with high resources costs and the capacity to provide deep insights. Heuristic evaluation is described as being resource efficient; however, it lacks the feedback of actual users. Focus groups, questionnaires, user feedback, and field observations are characterized as being able to provide deep insights, while requiring a considerable amount of time [17].

Another review of UEMs used for hospital electronic healthcare record systems identifies the ISO9241-11 and Nielsen's model as the most common used frameworks for usability evaluation [24]. The ISO9241-11 standard uses effectiveness (the accuracy in completing specific goals), efficiency (the amount of resources expended to achieve goals), and satisfaction (the amount of discontent during performance) as metrics to measure usability [8]. The Nielsen framework identifies learnability (how fast can the user learn to use the system), efficiency (how long it takes for the user to complete basic functionalities), memorability (how well can users remember to use different functions after learning), errors (how many errors do they encounter), and satisfaction (how pleasing the system is to use) as the main criteria on which usability is evaluated [19].

It is worth mentioning that a crucial step in the development or choice of an UEM rests on reviewing other empirical research that highlights common usability issues encountered in practice. For example, the most common components that rise usability issues in the case of UI environments are those related to content and representation, followed by those related to data discovery, access and manipulation, and finally by the logic flow of the application [1].

Other approaches to usability evaluation reveal that the choice of UEM must consider the characteristics of the system which is subject to evaluation. For example, the need for new instruments in the case of UEM for voice-assistants is emphasized because existing approaches are deeply rooted in graphical user interface (GUI) environments [25]. Moreover, attempts to evaluate and compare different UEMs can be found in the literature [10].

As software development organizations frequently adhere to the agile software development techniques, usability evaluation must be fit within a fast-paced development process [16]. For example, the discount usability engineering method provided a solution in this sense [18]. Other approaches aim to move the usability evaluation process before the actual implementation of the product. Such approaches are placed under the umbrella of early usability evaluation. The following section further expands on usability evaluation in the cybersecurity field.

Usability Evaluation in Cybersecurity

Research on usability evaluation in cybersecurity highlights experience and interaction factors as the two main components that need to be taken into consideration when applying UEMs [9]. However, most research in this sense focuses on the evaluation of specific components that ensure the security of larger systems, such as permission interfaces, authentication services, and/or security APIs.

There are attempts to apply UEM on cybernetic security systems rather than on cybersecurity components of larger systems. A study in this sense compares the usability of cybersecurity alerts in three different scenarios created

within an augmented reality environment [15]. The authors used a combination of usability testing (i.e., system usability scale) and usability inquiry (i.e., questionnaires) to determine the best simulated environments in which non-experts investigate alerts raised by a cybersecurity defense mechanism.

Other approaches that attempt to evaluate the usability of cybersecurity systems focus on how well the product can be learnt and used to achieve specific goals, as well as how satisfied the users are with the process. For example, a study that compares the usability of multiple online “Capture the Flag” platforms uses a combination of criteria based evaluation and system usability scale [5]. Thus, the use of multiple UEMs is specific for studies in the field. However, there is limited research regarding the application of UEMs for early usability evaluation of cybersecurity web platforms.

Early Usability Evaluation Methods

A systematic review on UEMs used in web development reveals that 80% of the usability evaluations are performed in the implementation phase of web applications development [11]. This approach is seen as inefficient because the integration of extracted insights implies additional workload on developers, who could have used the usability feedback before the actual implementation of the system. In this sense, the use of early UEM is suggested to fill this gap.

One example of an early UEM is to assess the brevity of the conceptual model of a web application [21]. Panach, Condori-Fernández, Valverde, Aquino and Pastor [21] discuss about the brevity of the conceptual model in terms of the amount of cognitive effort required to understand the respective conceptual model. For this purpose, they conduct a usability inquiry method that uses questionnaires to assess the brevity of the conceptual model. However, the authors do not disclose more than a question that assesses the easiness to see information on the conceptual model, measured on a 1 to 7 scale.

Other research efforts [2; 22] that focus on early UEMs tackle the issue of early usability evaluation for the case of automatic generation of web applications. The main criteria introduced for the early assessment of usability of web applications are learnability, understandability, operability, attractiveness, and compliance.

Other approaches [4; 6; 7] focus on the stage following the design of the conceptual model. The authors discuss the application of UEM on mockups of various healthcare products. A mockup is here defined as a model of a product that illustrates how it will look and function after implementation. Yet, there are limited attempts to evaluate the usability of a cybersecurity application based on its mockup. As such, the next section illustrates our approach to evaluate the usability of Yggdrasil GUI mockup.

METHOD

We chose to conduct a usability inquiry approach based on a questionnaire to evaluate the perceived usability of the Yggdrasil GUI mockup. Given that the object of the evaluation was a mockup with a minimum level of interaction, we decided to use an adaptation of the AttrakDiff [3] measurement instrument.

In the AttrakDiff model, pragmatic and hedonic qualities enhance the system attractiveness and, inherently, the frequency of use and satisfaction with using the product [3]. Pragmatic qualities refer to aspects related to the perceived ease of use, whereas hedonic qualities refer to the perceived aesthetic qualities of the product. Despite the purpose of the Yggdrasil application is rather a pragmatic one, it was assumed that hedonic qualities also matter to stimulate the onboarding processes. That is, users are more likely to invest in learning to use the product if it possesses hedonic qualities. Furthermore, the use of a mockup as an evaluation object is especially suited to reflect hedonic qualities of the product. Thus, we also included the hedonic subdimension and, consequently, items related to attractiveness.

The adaptation of the instrument was performed together with the developers of the Yggdrasil GUI. As such, the developers went through the items of the aforementioned instrument and selected those that match the intended qualities of the Yggdrasil GUI. Furthermore, one item was reformulated to match the specificity of the platform. This resulted in 15 items for the usability evaluation questionnaire, organized according to the three dimensions of AttrakDiff. The first item was reformulated starting from the original “technical versus human” item to account for the pragmatic purpose of the product. As such, the item was reframed in terms of “technical vs. non-technical”.

The procedure involved the remote evaluation of the Yggdrasil GUI mockup by an availability sample of 15 individuals. We also included one open-ended question that asked respondents to identify possible improvement opportunities to account for other possible dimensions of usability, besides those covered by the adapted instrument. Furthermore, we also conducted a usability testing procedure based on the think-aloud protocol with 5 respondents.

The questionnaire also included demographic questions such as the age, gender, the educational and professional background of the respondent. The mean age of the respondents in the sample was of 27 years old; 53.3% of the respondents declared that they have a masculine gender, while 33.3% declared they have a feminine gender. One respondent declared as being non-binary, while another one opted to rather not say. Sixty percent of the respondents owned a master’s degree, while 40% owned only a bachelor’s degree. Except for one respondent who declared a non-technical profession (e.g., human resource expert), all of the respondents declared that they have a technical background (i.e., work in the IT domain).

Proceedings of RoCHI 2021

The questionnaire was created in Google Forms and distributed online to the respondents. The eligibility criterion was that the respondent must not be part of the Yggdrasil project. The opening of the questionnaire first asked the respondents to explore the Yggdrasil mockup for five minutes. Following this, the respondents had to answer one open-question regarding improvement opportunities of the Yggdrasil GUI, 15 items built based on the adaptation of the AttrakDiff [3], and the demographic questions. The order of the AttrakDiff items was randomized in the questionnaire to account for any halo response effect.

The mockup was implemented and distributed online via Adobe XD version 41.1. Using the Adobe XD service, we

created an early prototype with minimal interaction functionalities. The mockup included 21 interactive screenshots of the Yggdrasil main screens (the most representative ones are captured in Figures 1, 2, 3 and 4). Figure 1 introduces the Vulnerability Management Screen in which new emerging vulnerabilities are presented, together with key information. Security analysts can edit each entry and, once all details are properly defined, publish the new vulnerability with an EVE identifier (see Figure 4). Figure 2 introduces a global view of detected vulnerabilities from all sources, together with corresponding filtering criteria. Figure 3 depicts a sample of OSINT data sources, namely news websites and Tweeter accounts that are accessible from the second sub-menu.

| CVSS Base | Attack Vector | Summary | Affected Product | Version | Exploitable | Source | Actions |
|-----------|---------------|--|------------------|--------------|-------------|--------------|------------------|
| 5.8 | Network | In Apache HTTP server 2.4.0 to 2.4.39, Redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an unexpected URL within the request URL. | Apache Server | 2.2.14 | Yes | News article | [Edit] [Publish] |
| 7.5 | Local | An issue was discovered in Django 1.11 before 1.11.22, 2.1 before 2.1.10, and 2.2 before 2.2.3. An HTTP request is not redirected to HTTPS when the SECURE_PROXY_SSL_HEADER and SECURE_SSL_REDIRECT settings are used, and the proxy connects to Django via HTTPS. | Django | 2.1.2 | Yes | Twitter | [Edit] [Publish] |
| N/A | Local | An issue was discovered in Python through 2.7.16, 3.x through 3.5.7, 3.6.x through 3.6.9, and 3.7.x through 3.7.4. The email module wrongly parses email addresses that contain multiple @ characters. | Python | 2.6.2, 2.6.3 | Yes | News article | [Edit] [Publish] |
| 4.1 | N/A | Due to the behaviour of the underlying HTMLParser, django.utils.html.strip_tags would be extremely slow to evaluate certain inputs containing large sequences of nested incomplete HTML entities. | Django | N/A | No | Twitter | [Edit] [Publish] |

Figure 1. Vulnerability Management Screen.

Figure 2. Detected Vulnerabilities Management Screen.

Figure 3. Data Extraction Management Screen.

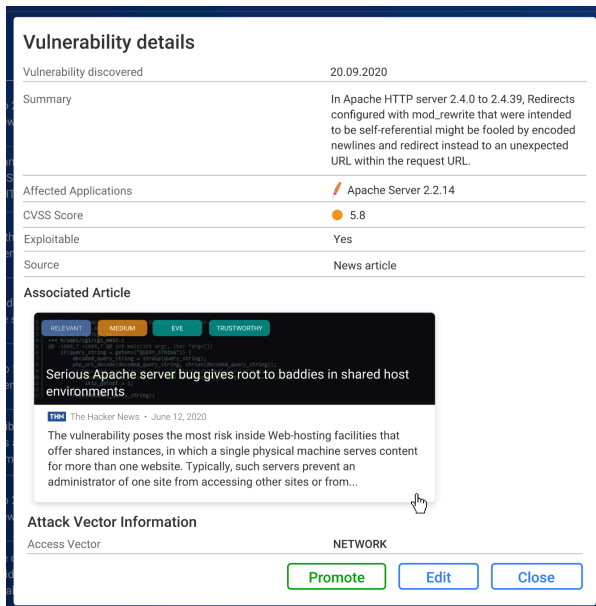


Figure 4. Vulnerability Details Editor.

Three tasks that could be completed using the Yggdrasil mockup were also designed for the think-aloud procedure. The first task asked respondents to attach an EVE identifier to the first vulnerability from the “Vulnerability Manager” tab. This task required the user to click the first “pencil” button illustrated in Figure 1. and then click the “Promote” button presented in Figure 4. The second task required respondents to display only relevant articles in the “OSINT” tab. This task required clicking the “OSINT” tab and selecting only relevant articles by means of the second filter presented in Figure 2. Finally, the respondents were asked to find and open the “Twitter accounts” subtab from the “Sources” tab presented in Figure 3.

The think-aloud protocol involved the recruitment of 5 available respondents. The eligibility criterion was that the respondents must not be part of the Yggdrasil project and that they have an IT background (either have an engineering degree or work in a software development company). One of the respondents was a quality assurance engineer. Two respondents are developers in two different cybersecurity organization. Another respondent occupies a human resources position in a cybersecurity organization. The final respondent is a consultant in software development.

After the respondents were given the tasks, they were recorder while remotely using the mockup to achieve the tasks. Uttered perceptions of the respondents, as well as issues and difficulties encountered while attempting to achieve the tasks were listed and reported in the following section. Additionally, perceptions of the respondents regarding the overall GUI were elicited and reported.

RESULTS

Table 1 presents the mean scores, with corresponding standard deviations, obtained for each of the selected items of AttrakDiff. For each item, a mean score of 1 would illustrate that all the respondents perceived the mockup as having the first characteristics of the item (e.g., technical, familiar, unpleasant), while a mean score of 7 would illustrate that all the respondents perceived the mockup as having the second characteristic of the item (e.g., non-technical, unfamiliar, pleasant).

Table 1. Scores for the selected criteria from AttrakDiff [3].

| Selected criterion | Mean (Standard deviation) |
|---------------------------------|---------------------------|
| Pragmatic quality | |
| technical vs. non-technical | 2.60 (1.21) |
| confusing vs. structured | 6.00 (1.06) |
| complicated vs. simple | 5.67 (0.97) |
| hard to use vs. easy to use | 5.93 (0.59) |
| discouraging vs. motivating | 5.20 (1.01) |
| Hedonic quality | |
| familiar vs. unfamiliar | 2.80 (1.26) |
| conservative vs. innovative | 4.80 (1.01) |
| dull vs. captivating | 5.26 (1.03) |
| old vs. new | 5.26 (1.22) |
| boring vs. interesting | 5.60 (1.05) |
| Attractiveness | |
| unpleasant vs. pleasant | 5.93 (1.38) |
| cheap vs. premium | 5.20 (0.94) |
| unpresentable vs. presentable | 6.00 (0.84) |
| repelling vs. attractive | 5.66 (1.04) |
| unprofessional vs. professional | 6.20 (0.86) |

The mean scores presented in Table 1 highlight the following traits for the mockup in terms of the three dimensions: pragmatic qualities (technical, structured, simple, easy to use and motivating), hedonic qualities (familiar, innovative, captivating, new and interesting design), and attractiveness (pleasant, premium feel, presentable, attractive, with a professional design).

We also decided to test the reliability of the adapted instrument. In this sense, we ran a series of Cronbach’s Alpha measurement procedures on the pragmatic and hedonic items of the adapted AttrakDiff dimensions. For the

pragmatic quality dimension, we obtained an acceptable Cronbach Alpha (.735) after we eliminated the “technical vs. non-technical” item. This suggests that our reformulation and adaptation of the item was done poorly. The same procedure revealed an acceptable Cronbach Alpha (.755) for the items in the hedonic quality dimension.

The Cronbach Alpha procedures suggest that our attempt to select only items applicable to the intentions of the GUI developers was successful. However, our reformulation of the original term “human” from AttrakDiff into “non-technical” seems to be problematic. As such, we decided to exclude the item from the procedure used to label the mockup in a two-dimensional space, given the average scores on the hedonic and pragmatic qualities.

According to the AttrakDiff single evaluation method, the Yggdrasil mockup can be labeled as being task-oriented. This label is attached based on the average scores of the pragmatic quality items (5.7) and the hedonic quality items (4.78). Given the purpose of the Yggdrasil, such a label is rather satisfactory. However, these results reveal that improvements can be made.

Furthermore, we used the open-ended question and a usability testing procedure based on the think-aloud protocol to identify specific improvement opportunities for the Yggdrasil GUI. The resulting feedback from the open-ended question and the think-aloud protocol were classified into the following categories: chromatics, understandability, and operability.

In terms of chromatics, the analysis of the open-ended questions suggests that the color scheme makes the text and edit options difficult to follow. Suggestions to change the blue background color, the dark background color of the cards in Figure 2, and to add more contrast between elements were provided. Similarly, suggestions include the usage of brighter colors with no gradients. Furthermore, feedback obtained from the think-aloud protocol suggests that the green color of the “Promote” button in Figure 4 is aesthetically unpleasant.

With regards to understandability, the analysis of the open-ended questions revealed minor problems. First, the name of the button “Promote” in Figure 4 is confusing. The same feedback was obtained from the analysis of the think-aloud protocol. Another problem identified by means of the open-ended question is represented by the lack of a comprehensive description of the “Edit” function in Figure 4. Briefly, few respondents did not understand why they would edit a vulnerability identified by Yggdrasil; however, such perceptions were expected as this functionality is intended for a security analyst who curates the automated rankings.

Another identified understandability issue is that the information in the screen presented in Figure 2 is cluttered. Furthermore, the text in the Figure 2 cards was declared as being rather imprecise in describing the vulnerability. Additionally, a respondent involved in the think-aloud

protocol did not understand what “OSINT” means and declared the screen in Figure 2 as being confusing. Suggestions to increase the overall size of the text or to change the display structure of the screen presented in Figure 2 were provided in the open-ended question and in the think-aloud protocol, respectively.

Moreover, the analysis of the think-aloud protocol results suggests that the name of part of the filters in Figure 2 are confusing (e.g., “all terms”, “all tags”). Similarly, a respondent involved in the think-aloud protocol argued that he could not understand what “Relevant” means.

Further insights obtained via the think-aloud protocol suggest the use of a trustworthiness gradient to explain the three different tags of the sources of Yggdrasil sources (see Figure 3). Additionally, suggestions to apply bold text that corresponds to the subtabs (e.g., Twitter Accounts in Figure 3) were provided.

With regards to operability, we could identify two types of feedback in the open-ended question results. On one hand, we received feedback that addressed the Adobe XD framework rather than the Yggdrasil GUI (e.g., limitations of transitions between pages). On the other hand, we received feedback that suggested the addition of further filtering criteria in Figure 2.

Additionally, a suggestion to include the option to edit sources from the cards in Figure 2 was provided. The reason behind this suggestion was to allow users to quickly eliminate sources that provide irrelevant data. Furthermore, a suggestion to create a filter that incorporates all the other filters in Figure 2 was provided during the think-aloud protocol. Finally, one of the respondents involved in the think-aloud protocol suggested the use of the Jira software (<https://www.atlassian.com/software/jira>) as an inspiration for the logical flow of our product.

DISCUSSION

The use of the AttrakDiff instrument limited our view on usability as a multi-facet concept. To account for this limitation, we added the open-ended question which resulted in relevant actionable insights, especially for the understandability dimension. Furthermore, we conducted the think-aloud procedure for the same purpose. This latter procedure also provided additional improvement directions.

It is worth noting that our study is limited by the characteristics of the sample. Given that the sample was obtained via an availability procedure, this study did not address exclusively potential end-users. Yet, as most of the respondents have a technical background, it may be argued that the current study provides relevant insights and guidelines for the future development of the Yggdrasil GUI. This is due to the fact that Yggdrasil is addressed to cybersecurity experts who, most often, have a technical background.

Given the results of our study, it seems the open-ended question and the think-aloud procedure were crucial for obtaining actionable insights with regards to the improvement of the Yggdrasil GUI. While the adaptation of the AttrackDiff instrument also provided a reference point for further evaluations procedure, the think-aloud procedure in which respondent record themselves or a task-based questionnaire that uses open-ended questions to collect written feedback may be the most suited approaches for the early evaluation of GUI mockups.

CONCLUSIONS

This study provides a demonstration for the early usability evaluation of Yggdrasil cybersecurity mockup. We used existing literature to illustrate that there are limited attempts to evaluate the usability of a cybersecurity GUI based on mockups. Afterwards, we conducted a usability evaluation of the Yggdrasil mockup. In this sense, we conducted both a usability inquiry procedure based on the adaptation of the AttrackDiff questionnaire together with an open-question, as well as a usability testing procedure based on the think-aloud protocol.

With regards to the usability inquiry procedure, the Yggdrasil GUI was described as being task-oriented. Furthermore, the open-ended question pointed towards specific improvements to make in terms of chromatics, understandability, and operability. Finally, the think-aloud procedure also pointed towards specific improvements in terms of understandability and chromatics.

Future work may involve repeating the usability evaluation procedures after the implementation of the Yggdrasil GUI interface. Another direction for future work refers to adapting other usability methods for the early usability of cybersecurity mockups. Yet, another direction for future work is to compare the feedback obtained in the current study with feedback from potential end-users. Moreover, future research should also seek to identify the most adequate approaches to develop an iterative early usability evaluation cycle to match the agile development process.

ACKNOWLEDGMENTS

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number 2PTE/2020, YGGDRASIL – “Automated System for Early Detection of Cyber Security Vulnerabilities”.

REFERENCES

1. Abraham, S.A., 2021. Usability Problems in GI Web Applications: A lesson from Literature. *AGILE: GIScience Series 2*, 1-7.
2. Abrahao, S. and Insfran, E., 2006. Early usability evaluation in model driven architecture environments. In *Proceedings of the 2006 Sixth International Conference on Quality Software (QSIC'06)*IEEE, 287-294.
3. AttrackDiff, 2013. Work Model. Retrieved 24 June 2021 from <http://www.attrackdiff.de/science-en.html#messen>.
4. Biesbroek, S., Shultz, J., Kirkpatrick, A., and Kortbeek, J., 2012. Human factors evaluation of an interventional trauma operating room mock-up. In *Proceedings of the 2012 Symposium on Human Factors and Ergonomics in Health Care*, 75-78.
5. bin Noor Azam, M.H. and Beuran, R. Usability Evaluation of Open Source and Online Capture the Flag Platforms.
6. Boman, I.-L., Lundberg, S., Starkhammar, S., and Nygård, L., 2014. Exploring the usability of a videophone mock-up for persons with dementia and their significant others. *BMC geriatrics* 14, 1, 1-11.
7. Chisholm, S., Shultz, J., Caird, J., Lord, J., Boiteau, P., and Davies, J., 2008. Identification of intensive care unit (ICU) system integration conflicts: Evaluation of two mock-up rooms using patient simulation. In *Proceedings of the Proceedings of the Human Factors and Ergonomics Society Annual Meeting*SAGE Publications Sage CA: Los Angeles, CA, 798-802.
8. Georgsson, M. and Staggers, N., 2016. Quantifying usability: an evaluation of a diabetes mHealth system on effectiveness, efficiency, and satisfaction metrics with associated user characteristics. *Journal of the American Medical Informatics Association* 23, 1, 5-11.
9. Grobler, M., Gaire, R., and Nepal, S., 2021. User, usage and usability: Redefining human centric cyber security. *Frontiers in big Data* 4.
10. Hartson, H.R., Andre, T.S., and Williges, R.C., 2001. Criteria for evaluating usability evaluation methods. *International journal of human-computer interaction* 13, 4, 373-410.
11. Insfran, E. and Fernandez, A., 2008. A systematic review of usability evaluation in web development. In *Proceedings of the International Conference on Web Information Systems Engineering*Springer, 81-91.
12. Iorga, D., Corlatescu, D.-G., Grigorescu, O., C., S., Dascalu, M., and Rughinis, R., 2021. Yggdrasil – Early Detection of Cybernetic Vulnerabilities from Twitter. In *Proceedings of the 23rd Conference on Control Systems and Computer Science (Bucharest, Romania (Online))*, IEEE.
13. Iorga, D., Corlatescu, D.-G., Grigorescu, O., Sandescu, C., Dascalu, M., and Rughinis, R., 2020. Early Detection of Vulnerabilities from News Websites using Machine Learning Models. In *Proceedings of the 19th RoEduNet Conference: Networking in Education and Research (Bucharet, Romania (Online))*, IEEE.
14. Ivory, M.Y. and Hearst, M.A., 2001. The state of the art in automating usability evaluation of user

- interfaces. *ACM Computing Surveys (CSUR)* 33, 4, 470-516.
15. Kabil, A., Duval, T., and Cuppens, N., 2020. Alert Characterization by Non-expert Users in a Cybersecurity Virtual Environment: A Usability Study. In *Proceedings of the International Conference on Augmented Reality, Virtual Reality and Computer Graphics* Springer, 82-101.
 16. Kane, D., 2003. Finding a place for discount usability engineering in agile development: throwing down the gauntlet. In *Proceedings of the Proceedings of the Agile Development Conference, 2003*. ADC 2003IEEE, 40-46.
 17. Lyzara, R., Purwandari, B., Zulfikar, M.F., Santoso, H.B., and Solichah, I., 2019. E-government usability evaluation: Insights from a systematic literature review. In *Proceedings of the Proceedings of the 2nd International Conference on Software Engineering and Information Management*, 249-253.
 18. Nielsen, J., 1994. Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. *Cost-justifying usability*, 245-272.
 19. Nielsen, J., 1994. *Usability engineering*. Morgan Kaufmann.
 20. Nurse, J.R., Creese, S., Goldsmith, M., and Lamberts, K., 2011. Guidelines for usable cybersecurity: Past and present. In *Proceedings of the 2011 third international workshop on cyberspace safety and security (CSS)IEEE*, 21-26.
 21. Panach, J.I., Condori-Fernández, N., Valverde, F., Aquino, N., and Pastor, O., 2007. Towards an early usability evaluation for web applications. In *Software Process and Product Measurement* Springer, 32-45.
 22. Panach, J.I., Condori-Fernández, N., Valverde, F., Aquino, N., and Pastor, Ó., 2008. Understandability measurement in an early usability evaluation for model-driven development: an empirical study. In *Proceedings of the Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, 354-356.
 23. Rogers, Y., 2012. HCI theory: classical, modern, and contemporary. *Synthesis lectures on human-centered informatics* 5, 2, 1-129.
 24. Wronikowska, M.W., Malycha, J., Morgan, L.J., Westgate, V., Petrinic, T., Young, J.D., and Watkinson, P.J., 2021. Systematic review of applied usability metrics within usability evaluation methods for hospital electronic healthcare record systems: Metrics and Evaluation Methods for eHealth Systems. *Journal of Evaluation in Clinical Practice*.
 25. Zwakman, D.S., Pal, D., and Arpnikanondt, C., 2021. Usability Evaluation of Artificial Intelligence-Based Voice Assistants: The Case of Amazon Alexa. *SN Computer Science* 2, 1, 1-16.