National College of Ireland

User experience evaluation can provide an early warning system to identify and treat potential security issues in software development.

MSc Internship
MSc. Cyber Security (Evening)

Raymond O'Brien
Student ID: 17110971

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Raymond O'Brien
Student ID:	17110971
Programme:	MSc. Cyber Security (Evening) **Year:** 2017-2020
Module:	MSc. Internship
Supervisor:	Ross Spelman
Submission Due Date:	17 August 2020
Project Title:	User experience evaluation can provide an early warning system to identify and treat potential security issues in software development
Word Count:	6875
Page Count:	33

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Raymond O'Brien

Date: 14 August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

User experience evaluation can provide an early warning system to identify and treat potential security issues in software development.

Raymond O'Brien
17110971

Abstract

Human error is considered one of the most difficult threats to detect. In 2020, 62% of insider facilitated incidents were the result of human error. This research project explores the gaps in traditional software development methodologies to identity areas of the processes where human error can be better treated to mitigate potential security risks. The Human-Centred Design "heuristic testing" method will be evaluated to identify and treat potential cyber security vulnerabilities and weaknesses in early stages of software development by evaluating the the user's experience before any coding begins.

1 Introduction

In cyber security, a vulnerability is a weakness that can be exploited to gain unauthorised access to, or perform unauthorised actions against a computer system causing unintended results or outcomes. Vulnerabilities can allow unauthorised code to run, access a system's memory, install malware and steal, destroy, or modify sensitive data; with intent or without (Tunggal, 2020). Vulnerabilities in systems can come in many forms but are largely due to the result of complexity and exploitable bugs left in code; often because of poor coding practice such as not utilising secure coding standards or otherwise by an over convoluted design. It is known that complex systems are hard to predict and can lead to unexpected outcomes with a high certainty they will contain vulnerabilities (Pompon, 2017).

Cyber security is consistently seen as a technical problem to be solved by expensive, complex and complicated technical solutions. Albert Einstein said, "If you can't explain it simply, you don't understand it well enough" (Beaver, 2013). To not understand a problem allows for unnecessary complexity and causes vulnerability when developing a solution or a product designed to solve a problem. Allowing vulnerabilities to exist opens the opportunity for a potential security incident to be caused. But what is cyber security?

Security can be defined as "the state of feeling happy and safe from danger or worry" (Oxford University Press, 2020). A sense of security is to feel safe from an attack or wrongdoing. Paying consideration to security in software development, therefore is paying attention to the requirement of ensuring an emotional sense of security of an end-user; or logical understanding of security when applied to an organisation. The end-user's sense of security is the emotional response when using a product; while the sense of security is managed by an organisation using "risk management" (Warsinske *et al.*, p17-30, 2019)

Warsinske *et al.,* (2019) describes an amendment to the traditional software development life cycle (SDLC), a commonly used methodology and workflow for developing software solutions, is to include security as a requirement throughout the process. Known as the Secure Software Development Life Cycle (S-SDLC). The S-SDLC concentrates on

incorporating security from a technical and process perspective; e.g. secure coding, configuration standards, security awareness, security testing, threat modelling. However, apart from functional usability testing in the later stages of development, it does not empathetically consider the end-user and their experience from the initial design stages.

Consider a software product as a solution to a "user's" problem. Why then do we feel a need to exclude the end-user from much of the process? Especially when common end-users or "insiders" are reported as the cause of the most expensive data breaches (Verizon, 2019). It is a widely held opinion that insiders are the biggest threat to an organisation's security. This has been confirmed in the following research by surveying a group of 102 participants across multiple industry verticals. However, it is acknowledged in Verizon (2020) reports that there has been a large increase in the number of incidents caused by human error.

Human error is a fundamental problem that should be addressed continuously in the design and development of systems. Without addressing human error in system design, it is the opinion here that a system cannot claim to be secure by design (Andrews, 2017).

Security by Design is a concept employed to ensure any system or solution is designed from the beginning with security in mind, to reduce the potential risk of an incident or exploitation of a vulnerability. The principles that govern Security by Design primarily concern themselves with employing technical or process-based solutions (Mailloux *et al.,* 2018). This opens the question; how do we focus on the human element while employing Security by Design and ensure the user and the problem are considered from the beginning? Is there a process we can employ to evaluate the human element in solution design while adhering to a secure by design outcome?

By evaluating the user experience, it is possible to solve fundamental problems, instead of treating symptoms of a problem (Norman, 2018). By only treating the symptoms, we run the risk of producing unsuitable products which may unintentionally increase security risk. This concept frames the governing theme for the research conducted here.

Vulnerabilities are a symptom of poor design that greatly impacts security and at best, are often patched or reworked with other complex technical solutions without addressing the fundamental problem (Tsipenyuk, 2005). If we can improve the design process to fix the fundamental problem, we can reduce the symptoms and in turn reduce risk.

Given the issues introduced above, it is intended to explore the Human-Centred Design process as a solution to validate a "usable" product, while reducing the security problem, in turn lowering or mitigating risk.

"User experience evaluation can provide an early warning system to identify and treat potential security issues in software development"

2 Related Work

2.1 Tackling the problem at the source

Software solution design has largely remained a specialism that does not usually consider the end user's true requirements during operation. It is typically based on a primary set of functional and non-functional requirements to allow a user complete a task. The software developer continues to design and develop in how they perceive a user should complete a task using traditional approaches, such as Waterfall; with rapid requirement analysis which are not overly suited to solving/addressing repetitive problems that require an innovative solution (Blosch *et al.,* 2017). The developer is seen as a "superhuman taking care of design, coding, updating

digital products, and also solving all server-related issues" (Oślizło, 2019). With poorly resourced teams and testing primarily evaluated during a single phase in the development lifecycle (towards the end, after coding and prior to release) it is difficult to anticipate and treat coding or design errors that may lead to a security risk. With roughly half of all security defects introduced at the source code level, Tsipenyuk (2005) denotes coding errors as a critical problem in software security.

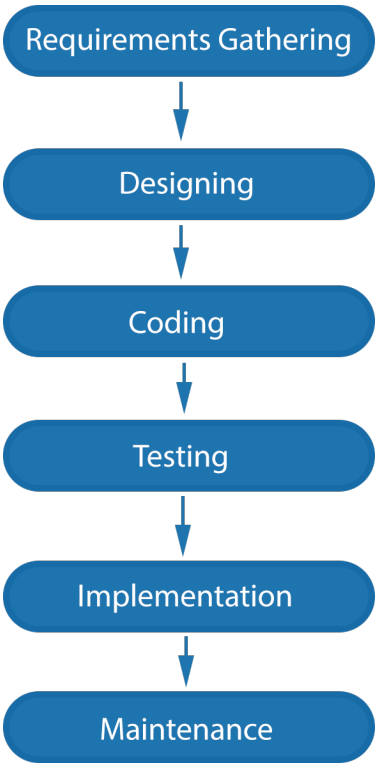


Figure 1: Waterfall Software Development Methodology

With Agile (an iterative rapid development methodology based on shortening time between requirements gathering, feature definition and delivery), "Programmers want to start programming day one, even before we know what we want to build" (Norman, 2019). This can offer an explanation for a large portion of issues experienced by end-users. Not understanding the users and how they will interact with a solution to "their problem" makes it very difficult to anticipate how a system may be misused or abandoned altogether. Agile, as per its 2001 manifesto, is designed to be a flexible improvement on Waterfall, however the adoption of Agile has many issues, including stakeholder buy-in (Miller, 2013), lack of cross-functional teams and the focus placed on technical requirements (Chervenkova, 2019). These processes do not appear compatible with empathetically evaluating and treating user experience issues. This may lead to the inability to identify security vulnerabilities early in the design process derived from human error. Table 1 shows a list of software development models discussed here (Singh, 2020).

Table 1:SDLC's Pro's and Con's[1]

SDLC Method	Pro's	Con's
Agile	Flexible to changes in requirements	Difficult to determine effort required
	Fast development and testing	High-risk due to flexible requirements
	Cost and time effective	Less emphasis on design and documentation
Iterative	Allows developers to test earlier in the process	Not all requirements are gathered at beginning - risk of design issues
	Flexible to scope/requirements change	Iterations are rigid
	Easy to manage	not suitable for small projects
	Less time for documentation more time for design	requires highly skilled resources
Waterfall	All development issues are defined in design phase	Longer time to delivery compared
	Each phase is well defined - easy to manage	Not flexible to requirements change
	Test scenarios are easy specified	Difficult to conceptualise client needs in terms of a functional specification during the requirements phase

2.2 Insider Threats – increase in data breaches due to human error

Insider threat is traditionally associated with malicious employees expressing an agenda to exploit and cause harm (Ponemon Institute, 2018). They often work in collaboration with others to exploit or take advantage of direct flaws or weaknesses in systems to access data for profit, or personal gain. Non-malicious insiders are negligent subjects that can unintentionally cause security incidents due to lack of security awareness or the inability to complete a work instruction safely. These instances can result in an equally high risk of a security breach as the malicious insider (Fortinet, 2019).

Since 2015, security breaches due to human error have seen a consistent increase (Verizon, 2020). Insider threat is an increasing problem with the number of insider-caused cyber security incidents rising by 47% since 2018. As shown in figure 2, negligent actors are the most common category of insider, accounting for 62% of all insider incidents, with an average cost of just over $307,000 per incident. The average cost of these incidents rose by 31% for the same two years (Ponemon Institute, 2020).

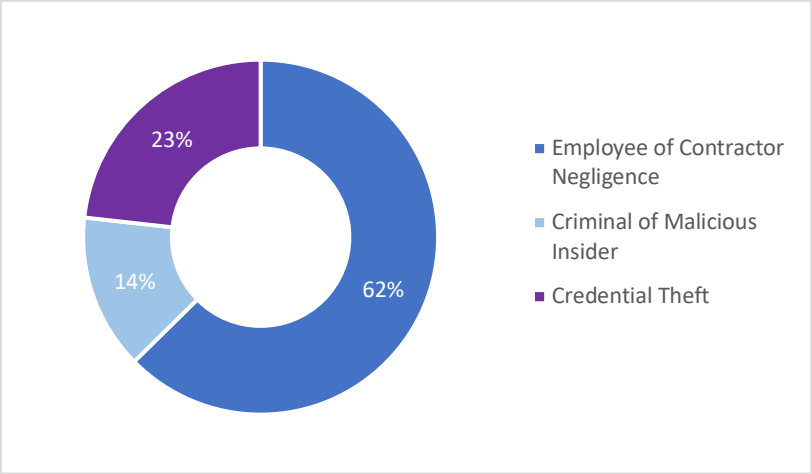


Figure 2: Frequency of incidents between different Insider Threat Profiles – (Ponemon Institute, 2020)

Human error is one of the most difficult risks to mitigate when designing a solution or planning security controls. It is one of the single largest attack surfaces reported (Alashe, 2020) yet solution designers and software developers are often siloed from security experts, user

[1] https://hackr.io/blog/sdlc-methodologies

experience (UX) professionals and the end-user(s) by various agendas, inappropriate processes and a pseudo understanding. Bird (2018) recommends functions should work in unison and operate as a cross-functional team with aligned processes. Neither UX design nor security should be afterthoughts to the development process and should be merged to pertain a holistic approach to solving problems.

2.3 Integrating Security into the SDLC

The rise in awareness of cyber security has faced many challenges across multiple domains and struggles to be adopted. The software development lifecycle (SDLC) holds most attention regarding the adoption of security practices (Nguyen *et al.,* 2019). The SDLC is a framework that guides phases of a software development project from inception to implementation. Security engineering teams engage earlier in the SDLC to mitigate security issues which increases efficiency and reduces risk (Warsinske *et al.*, p620, 2019). While this collaboration is improving through various process integrations, there is still a gap between user experience (UX) designers and the software and security engineers. The result of this is "trade-off's" between function, security and usability (Nguyen, 2019). Also, not recognised, is the potential for UX designers to reduce the security overhead during the product lifecycle by addressing potential issues in the earlier design phases. The research of Human Computer Interaction (HCI) and Human-Centred Design processes contribute best practices to evaluate the user experience of a product by iteratively evaluating and evolving the usability design, which is often seen as a contradiction to the goals of cyber security; a discipline to restrict interaction.

The proposal to introduce security earlier in the SDLC is to address the issue of traditionally having security engagements at the end of the cycle. This is seen as too late, as the major architectural and design decisions had already been made. If a security issue is found at the end of the cycle, there is less scope or appetite to have it treated before go-live (Nguyen *et al.,* 2019). If a security bug is found, it will be patched at a later stage. This repositioning of security and embedding it within the SDLC has formed the Secure Software Development Lifecycle (S-SDLC) with a simple objective: to design security into the system, not add it later (Warsinske *et al.*, p620, 2019).

2.4 Security Testing

Security professionals are coming to realise the delusion of the test and patch model that was popular in information security during the 1990's and is still common today (OWASP 2020). Known as the "patch-and-penetrate model", it involves fixing issues once reported, but without proper investigation of the root cause (McGraw, 1998). When an issue is detected early within the SDLC it can be mitigated faster and at a lower cost. To make this possible, development and QA teams should be made more aware of common security issues and the methods to detect and prevent them. There are various tools and methodologies used for security testing applications. Table 2 shows four common security testing methods.

Table 2: Pros and Cons of Security testing methods

Method	Purpose	Techniques	Pro	Con
Static Application Security Testing(SAST)	Analyse application code and design for security vulnerabilities	Technical Reviewes Walk-throughs Static Code Review	Works on any type of application Can Detect Behavioural issues	Unable to find business logic flaws Requires code
Dynamic Application Security Testing(DAST)	Analyse applications by actively exploiting them	Unit Testing Integration Testing System Testing	can run without access to source code High accuracy in detecting server missconfiguration	Unable to find business logic flaws Requires functional application Time consuming High False positive rate
Interactive Application Security Testing (IAST)	Used in DevSecOps for continuous security testing and monitoring	Monitors security using an agent and integration with the application	Fast and accurate Can detect behavioural issues	Unable to find business logic flaws Requires the application to be attacked
Penetration Testing	To assess the architecture, components, and code of the application by simulating an attack	Uses a human approach to security tesing	Finds all forms of security issues Finds business logic flaws	Time comsuming and Expensive Requires functional applicatoin

There are pros and cons to using recommended security testing methods like mentioned in table 2 (Goslin, 2020). However, the inability to determine "business logic flaws" with the exception of penetration testing presents a common trait. According to OWASP, business logic vulnerabilities are weaknesses that can be exploited by using the legitimate processing flow of the application to result in a negative consequence (OWASP, 2020).

Another technique used is use case modelling. Misuse and abuse cases describe unintended and malicious scenarios of an application, commonly using Unified Model Language (UML). The goal is to describe all possible, or the most critical scenarios where a bad actor can misuse or abuse an application (OWASP, 2020). This may be the closest method comparable to the methods used in user experience testing. However, these are based on descriptive scenarios and do not involve observing or evaluating an actor in person or receiving feedback from a subject. Abuse and misuse cases are a hypothetical approach of modelling negative requirements, i.e. behaviours that should not occur in a system. They can be used to model attacks on a system, as well as document the security mechanisms needed to mitigate them. Abuse and misuse cases informally describe requirements at a high-level. This means that, while they can be easy to understand, they do not lend themselves to evaluative testing or analysis (Whittle *et al.,* 2008).

2.5 Human-Centred Design and User Experience

To merge any discipline with another (i.e. S-SDLC) we need to understand its context. Gartner promotes Human-Centred Design (HCD) as a creative approach to innovation that begins with people and shapes ideas that can become practical and attractive propositions based on understanding the user and its context (Blosch *et al.,* 2017). HCD is a Design Thinking methodology that places people at the centre of product design by involving end-users from the very beginning of the process and continuously evaluating their experience by iteratively testing and redesigning a product based on their interaction and feedback (Rizzo *et al.,* 1996*)*. By observing and understanding the user's experience while interacting with a system, developers and system designers are empowered with the knowledge and context of what is "truly" required for a user to satisfyingly complete a task or solve a problem without error, or frustration.

User experience research and design is focused on the understanding of effective design choices, designing for usability, user behaviours, accessibility, motivations and frustrations. The user experience designer specialises in the interaction between real human users and everyday products and services; combining aspects of business, psychology, market research, design and technology (White 2020). A user experience designer is responsible for the journey that a user takes and how the product is structured to facilitate this journey (Babich, 2017).

There are a range of methods to research, evaluate and design, including user personas, user flows, wireframes, prototyping, heuristic evaluation and conducting full scale user testing (Farrell, 2017).

Most research combing user experience and security revolves around the evaluation and design of security features or controls; methods of multi-factor authentication, password policies and encryption (Nguyen *et al.,* 2019). Pain points about the usability of these controls lead to resistance of adoption (Dupuis *et al.,* 2018). Security is an important feature of any product. However, it can be restrictive or cumbersome for groups of users. The balance of usability versus security often comes at the expense of one over the other. It is difficult to achieve effective security while also maintaining usability (Magalhaes, 2018).

Although security controls might make it hard for users to interact with systems, user experience as a process aims to provide positive experiences for the user and identify issues to increase user adoption. User experience is generally excluded from the software design process. Often felt unnecessary or impractical due to project constraints, like: time to market, budget, resourcing (Norman, 2019). However, fixing usability issues is more expensive after a product build or go-live due to many factors; most recognisably "retrofitting" or patching solutions as the issues become known. This is referred to as UX Debt (Kaley, 2018). Kaley (2018) describes UX Debt as the cost of having to address problems that result from usability issues after launch is higher than addressing them in the first place. Coding a user interface correctly in the first instance is much more efficient for developers than having to change shipped code.

2.6 Summary

In almost all cases, user interfaces should be designed iteratively because it is virtually impossible to design an interface without usability issues from the start. Iterative development involves refinement of the design based on evaluation and feedback analysis. This has some similarities with Agile methods. However, instead of building and then testing, UX methodologies allow for evaluating low cost, low fidelity prototypes and paper mock-ups. UX can be very effective in producing good quality in a rapid environment, but given its low-cost user centric methods it can be a very powerful tool to discover creative solutions to problems that may not be obvious until much later in the development process and become too costly to mitigate.

3 Research Methodology

This chapter outlines the deductive and inductive research methodology used to test and evaluate the theory: User experience evaluation can provide an early warning system to identify and treat potential security issues in software development. Figure 3 shows a high-level process derived from Human-Centred Design. It also shows the security assessment approach that will be used to validate the reduction or mitigation of risk through the Human-Centred design process.

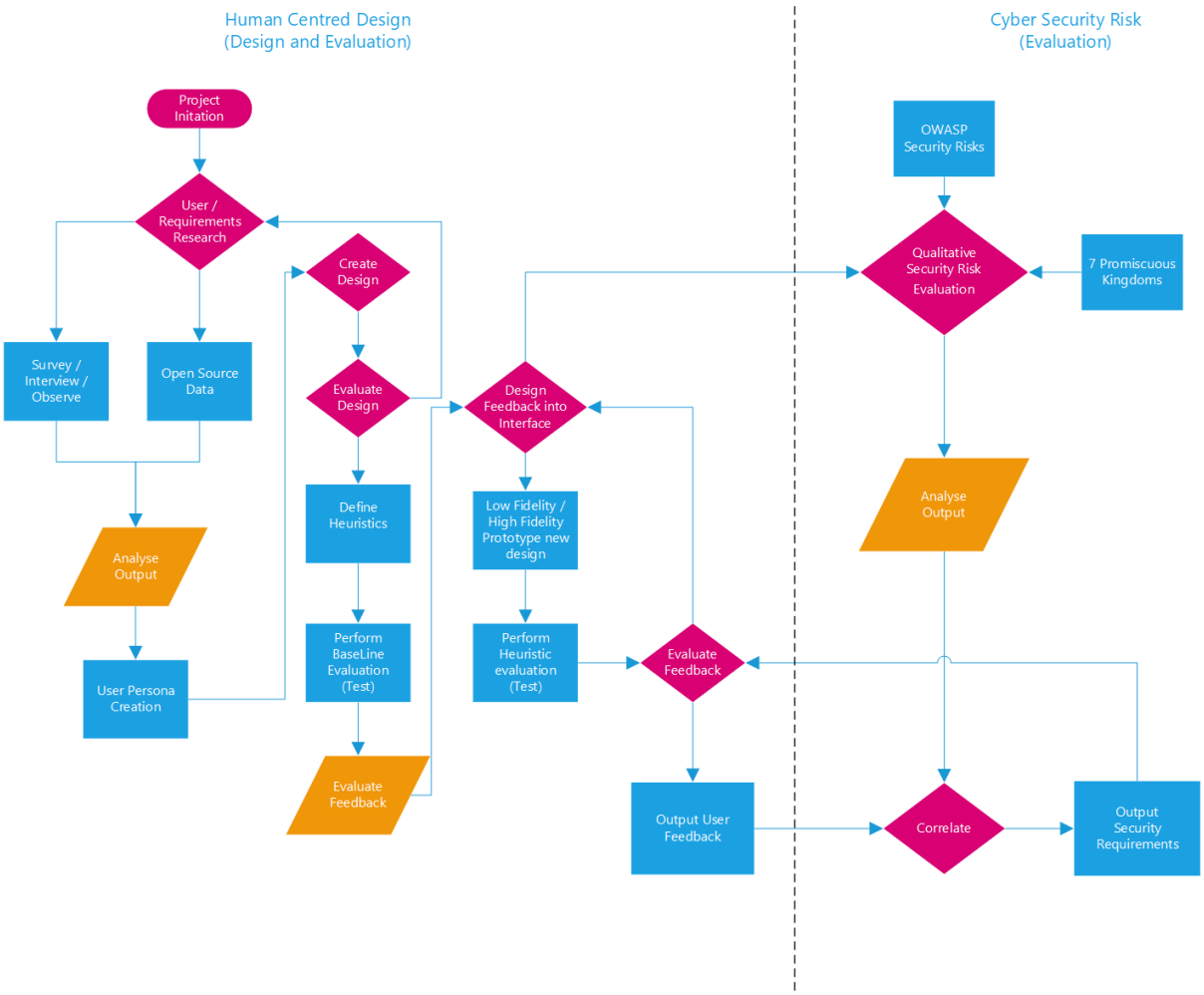


Figure 3: High Level Overview of Process Flow

3.1 Research Strategy

Heuristic tests will be conducted following the Human-Centred Design process and evaluated to address the research question. A cloud-based file transfer application was chosen as a baseline interface for evaluation[2]. Participants chosen, range in demographics. Such variety of perspectives should help gain a broader view of potential user experience issues in conjunction with security threats. A series of user personas will be generated to ensure the anonymity of the participants performing the human evaluation.

3.2 Qualitative Research

Science has a considerable reliance on quantitative and experimental methods. However, there are complex, socially based observations in Human-Centred Design that cannot be easily quantified. Qualitative research is preferred, as the emphasis is not on measuring and producing metrics but instead, understanding qualities of a particular solution and how people think and feel about their experience interacting with it. This method combines systematic levels of abstraction to produce theories grounded in multiple types of data gathered, categorised and

[2] Any application requiring user interaction could be used.

coded during the research. This is known as grounded theory (Vassilakaki, Johnson, 2015). Grounded theory supports iterative development, allowing for different, detailed perceptions to be sampled and analysed quickly.

3.3 Quantitative Research

To reduce the possibility of bias within the qualitative research, a series of quantitative outputs from the evaluations will be analysed. This is to code quantitative representations to the qualitative feedback for correlating and trending between the usability and security risk factors.

3.4 Rapid Prototyping

An industry standard high-fidelity prototyping tool, Adobe XD[3] will be used. XD provides a platform/workflow for the design and build of interactive prototypes. This will serve the purpose for replicating a sample application interface for the heuristic evaluation and to rapidly apply user participant feedback into redesigns for further evaluation.

Adobe XD allows for collaboration and handover of prototype designs to development teams. This is in the form of artifact files, i.e. artboard components, HTML/CSS information, design notes, and interactive walkthroughs.

3.5 Data Collection

A digital form-based survey will be created and distributed through a crowdsourcing marketplace. This should maximise diversity in participants to obtain a broad view. Results will be collected and collated into a dataset. Analysis of the data received will be used to compare with statistics regarding "Insider Threat" presented in the literature section.

An iterative heuristic evaluation will be performed using a low/high fidelity prototype design where feedback is collected using a series of digital forms. Each participant will perform a set of scenario-based tasks. The feedback will be compiled into a dataset. This dataset will contain qualitative descriptive responses. Where possible, the responses will be categorised using Nielsen's heuristics (Nielsen, 1994). Severity ratings will also be requested to quantify the level of feeling on a scale of 0 (no issue) to 4 (critical). This will allow for the measurement of change between iterations of design.

A qualitative risk assessment will be performed using OWASP security risk ratings applied to each iteration of the design. This risk data has been compiled based on research conducted by OWASP[4]. OWASP provides a framework for assigning a theoretical risk score to a security attack vector and/or vulnerability.

The aim of the risk assessment is to validate the reduction in risk indirectly resulting from iterative usability design decisions. The risk assessment will be conducted in isolation as not to directly influence the changes in design, i.e. the design decisions to address usability issues should not be influenced by the risk ratings of each iteration. The design changes will be based on user feedback in relation to the heuristics evaluated and a visual walk through of the prototype.

---

[3] https://www.adobe.com/ie/products/xd.html

[4] https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_Details_About_Risk_Factors

3.6 Personas

Personas will assist in the use case and scenario design and provide a visual representation of the end-user. A persona is not a description of a single person. Instead, it is an aggregate of a range of interviews and other information from a specific group of users with similar goals and backgrounds. The personas will be aggregated from information collected in the initial survey and merged with observations made during the user testing.

3.7 Data mapping

As mentioned, Nielsen's heuristics will be used to guide the user experience evaluation. A mapping exercise will be attempted to correlate a sample of Nielsen's heuristics with OWASP security risks. To assist this mapping, categorisation will be influenced by the 7 Pernicious Kingdoms (Tsipenyuk, 2005) [5] and where necessary the Common Weakness Enumeration database by The Mitre Corporation[6]. Although the databases are concentrated on coding and technical vulnerabilities, meaning will be inferred from the narrative of each to extend their attribution.

3.8 Analysis

All data received from the initial survey will be collated and prepared in Microsoft Excel. The data will be transfer to R-Studio for quick analysis and findings presented in section 6.

All feedback data received from the heuristic evaluation will be collated and prepared in Microsoft Excel. A data model will be prepared for coding the raw data into a usable format. The formatted data will be catagorised using the mapping in section 3.7. This will allow for the review and analysis.

3.9 Limitations

This research is based on human participation and feedback. Some limitations have been considered. Surveys and forms collect "point-in-time" responses. It is accepted that responses may vary and may change due to increased cognition from repetitive use of sample interface.

The initial survey is limited to 102 participants. By using online surveys, there is a moderate risk of abuse and receiving unusable data. To help minimise this, the survey is structured to use a token to ensure a participant can only complete once. This token also ensures the survey is completed fully before being committed.

The heuristic evaluation is limited to between 3 and 5 participants. It is recommended that larger populations do not add value. "Testing with 5 people lets you find almost as many usability problems as you'd find using many more test participants" (Nielsen, 2012).

Adobe XD is a digital interactive prototyping tool. It does not produce a fully functional product. However, it will serve as a sound method for the purpose of this research.

---

[5] https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/8013/33104/1556543/1556543-table-1-source-large.gif

[6] https://cwe.mitre.org/

3.10 Ethical Approval

The methods considered in this research, by nature has a high reliance on human participation. However, careful consideration has been taken to exclude recording of any personal data, or the involvement of vulnerable and risky demographics. Responses were captured electronically and anonymously. A disclaimer is to be provided to request consent, informing the participants no personally identifiable information will be collected.

Where personas appear to contain identifiable information, this is inferred from the anonymous data collected based on language, occupation and career position.

4 Design Specification

A sample mapping will be created of known standards and taxonomies for Secure Coding and Human-Centred Design process as shown in figure 4 below: 7 Pernicious Kingdoms (4), OWASP Top Ten Application Security Risks (6), Jakob Nielsen's 10 Golden Usability Heuristics (4).

As the secure coding taxonomies are reflective of coding techniques, technical vulnerabilities and controls, where a direct mapping is not possible, attributes and suggestive meaning derived from the descriptive narrative of each will be used. The goal is to create a relationship between the taxonomies for communication purposes.

The purpose of mapping is to produce a hypothetical risk score for the security risk types as found in the application interface through the evaluation of usability heuristics. For the purpose of this research we will concentrate on Nielsen's heuristics related to input, feedback and handling errors.

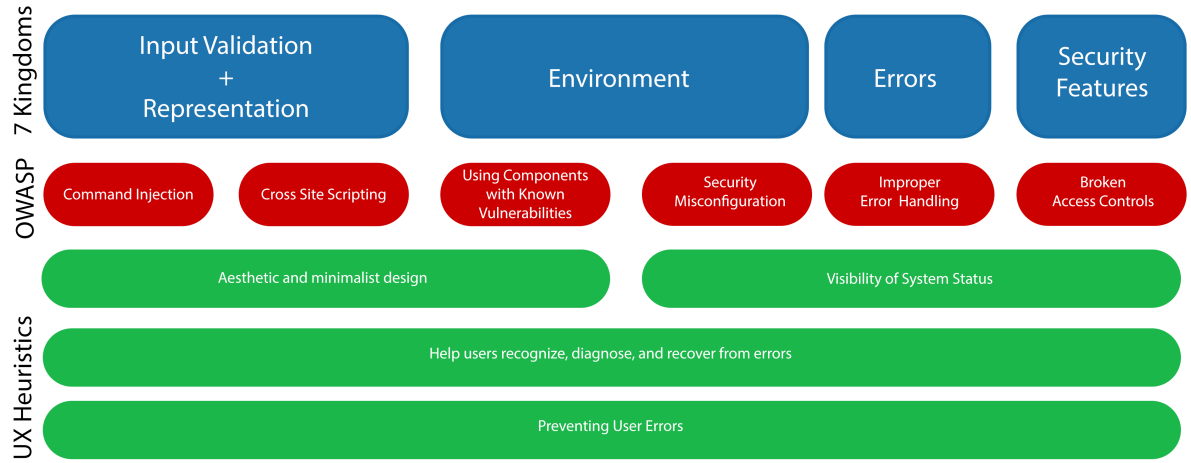


Figure 4: Mapping of OWASP/ 7 Pernicious Kingdoms and Usability Heuristics

A web-based file transfer application will be used as a template. The basic requirements are to allow a user share files by email and/or a URL. The application will be replicated using Adobe XD with branding removed. This will be our prototype. An assumption will be made, that no security controls have been applied to the application. The prototype will be evaluated using the scenarios in table 3 that are commonly expected with a web-based file transfer application. A spreadsheet will be used to collate the responses.

Table 3: Sample set of usability test scenarios

Task Number:	Scenario Description:
task 1	Make an assumption of what the function of the application is
task 2	Send a file from your desktop to the email recipient
task 3	Send a new file from your desktop to the email recipient ray.obrien@gmail.com
task 4	Send 3 files from your desktop to the email recipient - verify you have uploaded the correct files before sending
task 5	Send a password protected file to the email recipient - assume the file is highly sensitive
task 6	Send a file from your desktop to the email recipient - and set the expirery to 15 days
task 7	Send a file from your desktop to the email recipient - and set the expiry to 10 days
task 8	Send a file from abc@gmail.com using a custom link
task 9	Send a file from abc@gmail.com using a custom link + add special characters (e.g. *&<>?/) - you can use e.g. <script>alert("hello")</script> in the file name
task 10	Send a file from abc@gmail.com using a custom link you used in task 8

A risk assessment will be done of the application to produce a baseline risk score of the interface for each reported usability issue by the participant.

Feedback will be redesigned into the prototype application using Adobe XD and be prepared for the second round of heuristic testing.

For the purpose of theory demonstration only a sample of feedback will be incorporated for each iteration. This is to demonstrate the variance in user experience and its effect on potential risk where risk identified.

5 Implementation

The Human-Centred Design process discussed in literature and methodology sections, was followed as this is the primary process being used to validate the research theory.

5.1 Participants

Four participants were recruited for the iterative evaluation of prototype interface. It was important to maintain anonymity for the participants. To support this, the personas (see Appendix C) outlined have been altered using data collected from an initial online survey conducted. This method for personas provides a true, non-assumed depiction of some of the prototype's target-audience.

Each user participant walks through a set of scenarios and records feedback. Feedback will be collected using a digital form (available in the Appendix A) automatically submitted to the researcher for review. This process was iterated 3 times.

5.2 Researcher – Heuristic Testing

The researcher will introduce the project and provide instructions:

1. For accessing the prototype application to be tested
2. For accessing the digital form to record feedback
3. Guidelines for the participants to follow should any severe issues arise preventing the user from completing a scenario other than the usability of the interface[7].

5.3 Researcher – Design and prototype

[7] Note: The researcher will attempt to limit interaction with the participants to a minimum.

The researcher will prepare an interactive prototype using Adobe XD to replicate a chosen interface for the purpose of the heuristic and security testing. The prototype will be suitable to replicate most functions available and will be redesigned to interrupt feedback from the heuristic testing.

5.4 Researcher – Security testing

The security researcher will apply a security context to the participant's feedback, qualitatively assessing and assigning security risk attributions where possible using recommended conventions discussed previously.

5.5 Researcher – Analysing data

The researcher pools all data to be coded, manipulated and analysed using both R-Studio and Microsoft Excel. Excel is preferred due to its usability and more appealing visualisation outputs to be discussed in the evaluation section.

5.6 Design the prototype and test.

The interface was prototyped using Adobe XD and shared to the participants using a publicly accessible URL hosted by Adobe Creative Cloud[8]. A high-fidelity prototype was used to allow for interactions to simulate real-world user inputs and actions, as shown in figure 5.

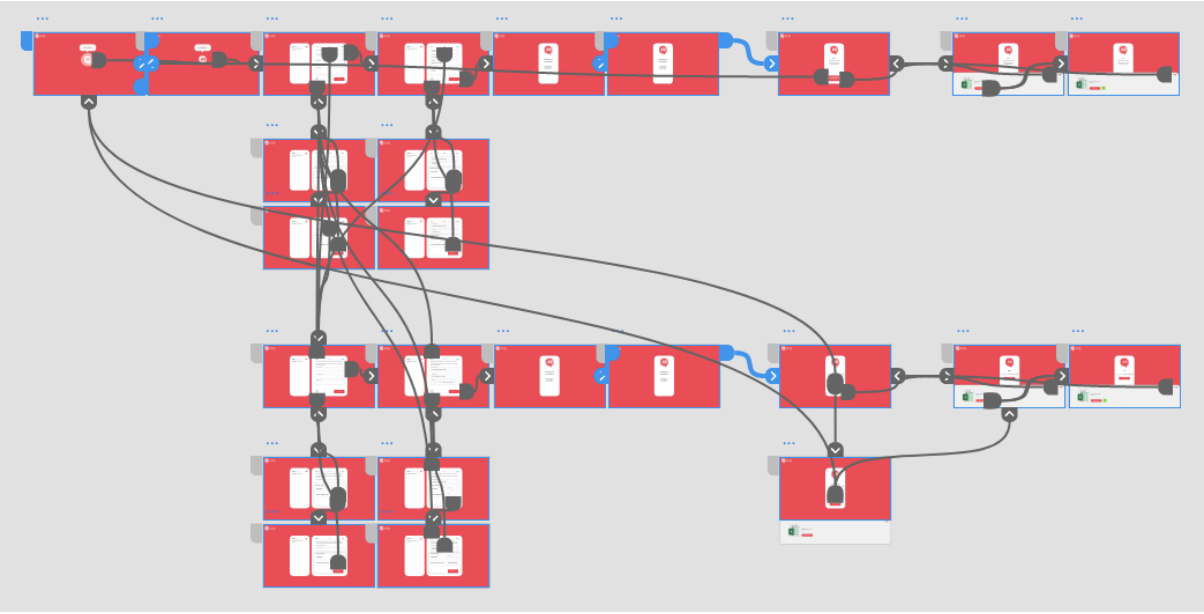


Figure 5: Interaction Storyboard

The same process can be applied to low-fidelity prototypes using paper mock-ups and wireframes. The decision is made based on a number of factors: time, budget, available resources, technical ability.

[8] https://www.adobe.com/ie/creativecloud.html

6 Evaluation and Results Analysis

6.1 User Error

A short analysis of the initial survey conducted on AmazonMTurk[9] was completed to complement a running theory expressed in the literature review. Respondents were primarily from design industries, healthcare, finance and customer service. Of 102 participants, 86% expressed their organisation considers users to be the highest security risk with majority as depicted in figure 6, admitting to sourcing "other" solutions to gap functionality missing from their business approved toolsets which may result in security risks, like Shadow IT[10].

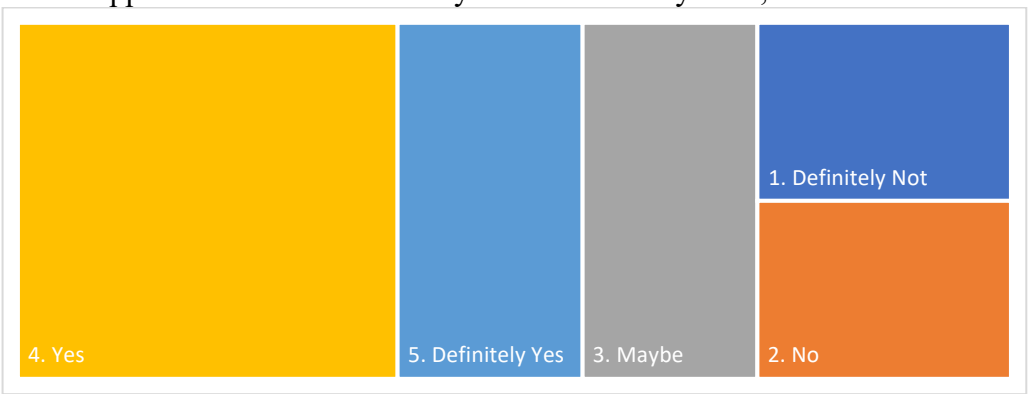


Figure 6: Use of other applications or services to gap missing functionality

When errors occur or the users experience is an issue, 66% claim the reason for the error is not easy to understand with only 57% reporting errors to their I.T. department. This is interesting when referring back to the 2020 Verizon report where negligent actors account for 62% of insider data breaches. Perhaps the 62% were not aware of the impact from the errors they were making and failed to report to I.T. as a result.

Some of the information in this survey data was taken and inferred to generate the user personas for the heuristic evaluation of the file sharing application commonly used across a range of industries. 4 users represent the top industries presented. Please see Appendix C for user personas.

6.2 Online web application – File sharing (MyApp)

Of the 4 participants, it was discovered that most issues reported were the result of error and/or interrupted experience of completing the scenario. An aggregate of 51 issues were experienced of which 64.7% were directly related to receiving an error or the participant not being able to recover from the error. Other minor issues were discovered related to the general usability of the interface represented in figure 7.

[9] https://www.mturk.com
[10] https://www.forcepoint.com/cyber-edu/shadow-it

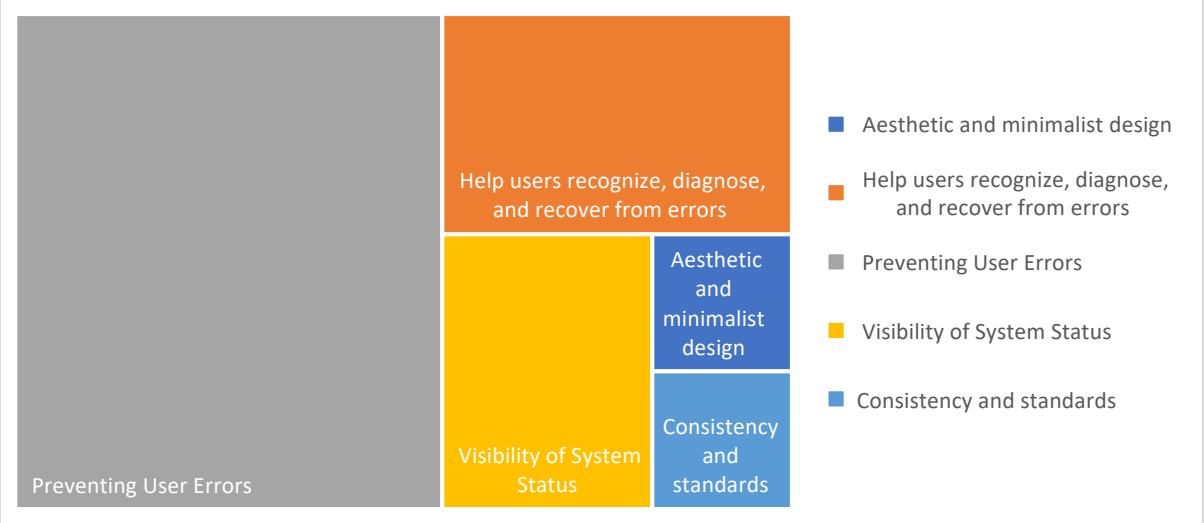


Figure 7: Frequency of issues found by heuristic

6.3 Preventing Input/Injection vulnerabilities before coding begins

A comparative analysis was performed between the feedback received from each iteration. It was determined, participants experienced most errors while trying to share a file using a custom URL (Link). This is interesting as the custom link feature was designed to allow users to create their own HTTP/URL string to share with a recipient.

It was found, the user enters the string into an input field that updates the URL "Preview" field (also found to be editable based on the template application). By evaluating the feedback, an input validation vulnerability was highlighted. The user was able to input dangerous special characters or worse in the preview field, e.g. enter a cross site scripting (XSS) request string, or a string that exceeds the safe HTTP/URL request size of the application and cause it to crash.

The consequence of allowing "code" injection[11] can result in severe risks which may impact confidentiality, availability and integrity[12] depending on the attack scope. For example, this may result in a format string attack, like buffer overflow. If this was discovered later in the SDLC testing phase through standard security testing discussed in literature previously reviewed, it would require implementation of controls, like input sanitisation or cause a break in the development process to jump back and patch issues before proceeding. However, by using design methods discussed, it was possible to reduce the risk by evaluating the "experience" issues to redesign the interface feature and user journey. Figure 8 shows the feature was redesigned to prevent users editing the URL preview field. It was felt unnecessary to be able to edit both fields. A custom error was added to the design to inform the user. This will be documented as a requirement for developers to implement in production.

[11] https://owasp.org/www-community/attacks/Code_Injection
[12] https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf

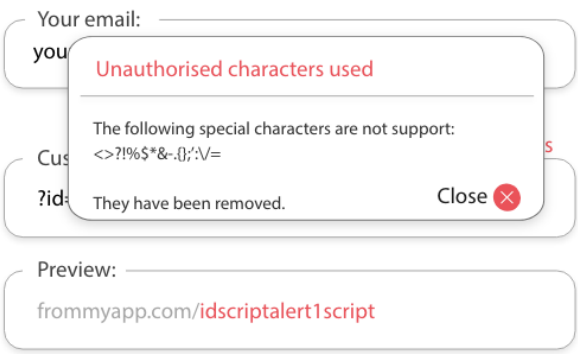


Figure 8: Input Sanitisation

6.4 Potential violation of privacy security features

A second prominent issue exposed the possibility for information disclosure and depending on the content shared, could be another user's sensitive information. Users discovered when trying to create a custom URL, if the URL had been used before, they would receive a feedback error message explicitly notifying them of this as shown in Figure 9.

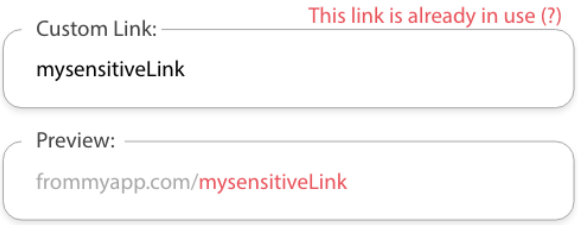


Figure 9: Privacy violation - This link is already in use

This could allow an attacker or curious individual, access a document(s) shared by another user. Feedback suggested to change the error message to something less exposing. An alternative message was proposed "You cannot use this URL, please try again", however this alternative was thought to offer a curious but less risky question, why can I not use this URL?!

Further analysis of feedback suggested to remove the feature. It was felt the application should enforce users to generate a random URL by clicking on an icon shown in Figure 10. This would remove the privacy concern, address the previous input errors discussed and also simplify the process for the user.

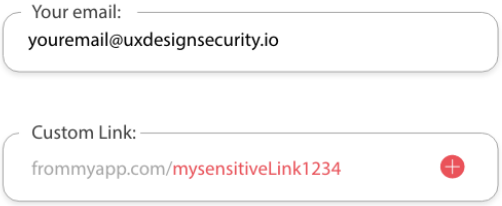


Figure 10: Generate random URL

Discovering these errors during the heuristic evaluation and treating them as "user experience" errors iteratively, quickly facilitated the rapid redesign of the interface to reduce human error and other potential weaknesses, validating the positive increase in user experience and consequently reducing risk before the developers start coding. Figure 11 visually represents the success of treating all users feedback in design versus the rate of risk decreasing between the iterations.

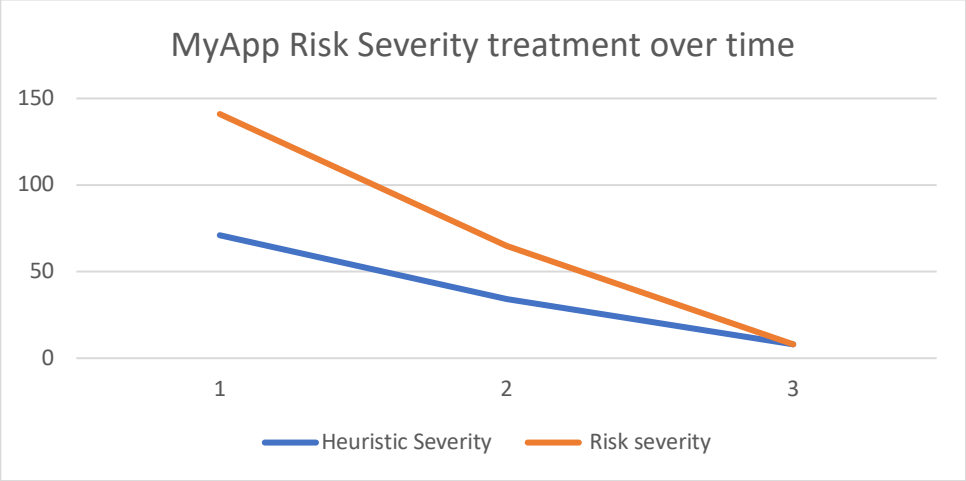


Figure 11: Total Risk Severity treatment over time

6.5 Present a hypothesis

Taking the reduction in heuristic severity represents an increase in user satisfaction, the theory can be expressed as:

$$H_0 \text{ where } r^{x+1} < r^x \text{ while } u^{x+1} > u^x$$

Where r = risk, u = user satisfaction, x = iteration

This can be demonstrated per user participant.

Shown in figure 12, User 2 suffered the worst experience and discovered the most potential risks. This is due to the number of issues reported per scenario completed.

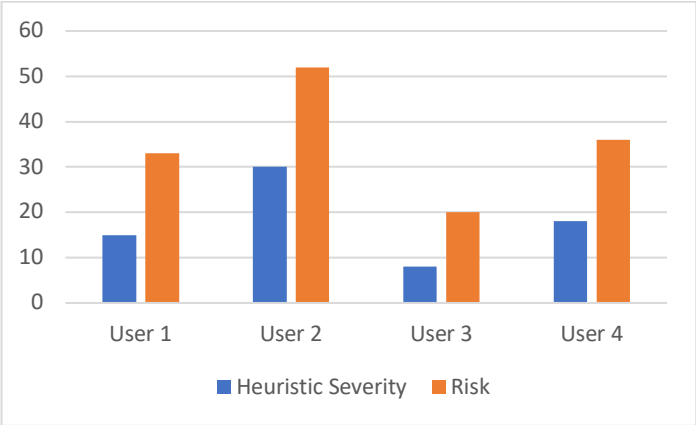


Figure 12: Participants risk versus heuristic severity

Figure 13 highlights a break down for User 2. The experience suggests there should be more consideration for preventing user error.

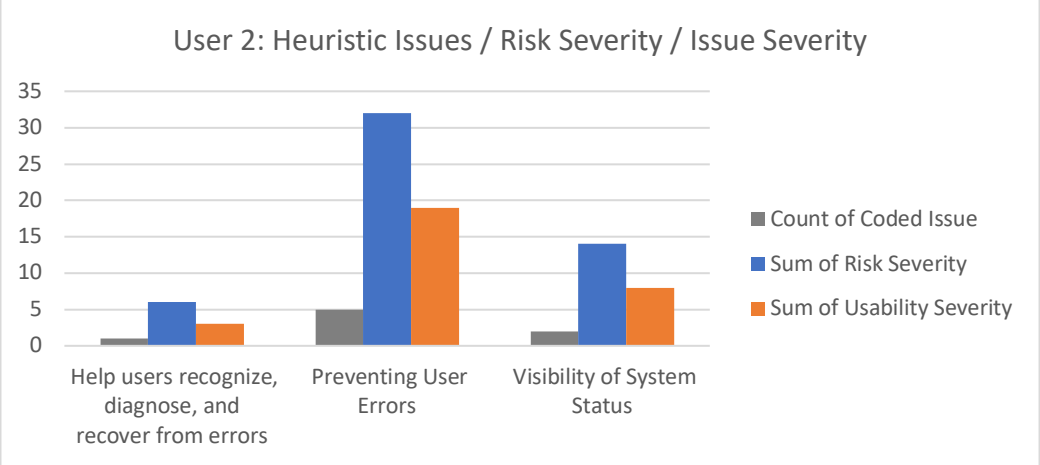


Figure 13: User 2's security risk versus heuristic severity

By treating the feedback presented from the user it was possible to improve its experience and also treat the potential risk between iterations before moving from design to implementation.

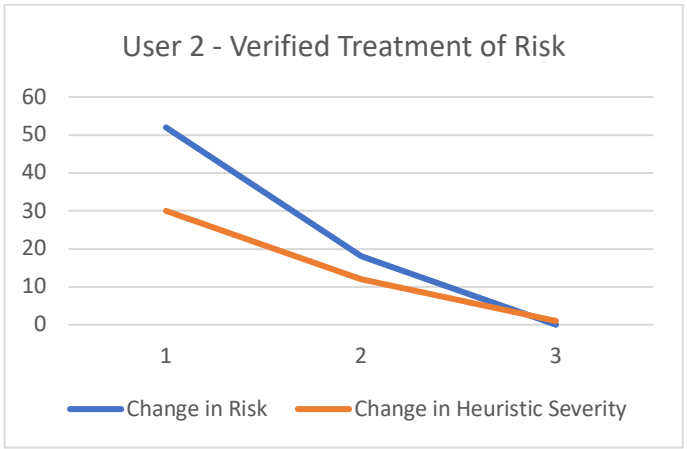


Figure 14: User 2 Reduction in risk vs Heuristic violation severity

Figure 14 demonstrates User 2's 60% improvement in satisfaction (represented by the decrease in heuristic severity) produced a 65% reduction of security risk between iteration 1

and iteration 2. Iteration 3 showed further improvement however the user has provided subsequent feedback recording cosmetic severity (1) due to the removal of the custom link feature.

Comparatively User 4 appeared to have demonstrated a more subtle improvement in satisfaction between the iterations. However, analysing the data shows a 61% increase in satisfaction with a 78% reduction in risk for the design.

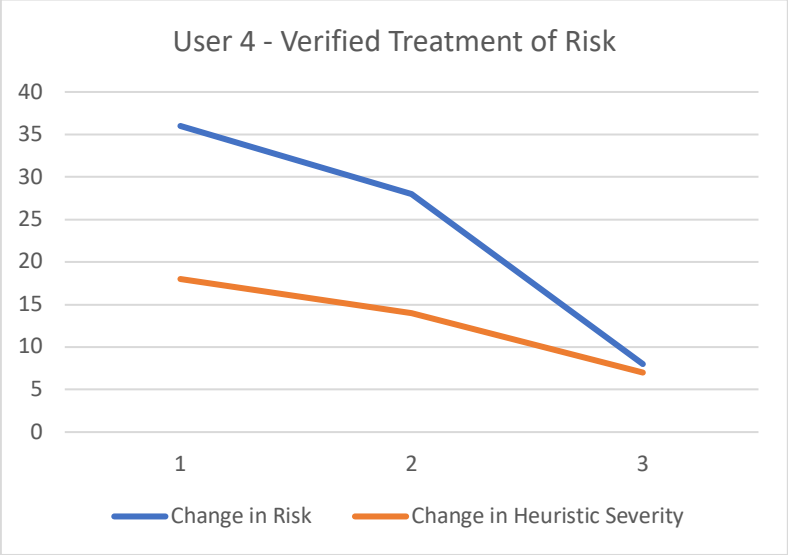


Figure 15: User 4 Reduction in risk vs Heuristic violation severity

The residual risk of 22% shown in figure 15 reflects feedback not treated in the redesign. For further comparisons please refer to Appendix B.

6.6 Discussion

As the evaluation and analysis have shown, it is evident an increase in positive user experience reflected by the decrease in heuristic severity results from incorporating the user's feedback between each iteration of the process. It also demonstrates that by not incorporating feedback leaves the user feeling negative to the experience. Listening to users works, the feedback may not always be in the designer's interest, however, the evaluation allows for an informed decision to be made.

More interesting is the impact this improvement in experience reflected upon security. The negative experience related to errors and error handling particularly exposed the potential for bad design to flow downstream to development. Mitigating potential risk by design that would otherwise not be detected until later in the process; depending on where security testing and evaluation is integrated. Meaning it was possible to quickly validate the user's response after each design change while also verifying the risk posture. However, this did require a stakeholder versed in security practices to substantiate.

The user experience researcher/designer alone, may not be in a position to assess the security risk independently, despite being able to treat "risk" through alternative requirements. However, if the user experience researcher/designer is upskilled to be sufficiently security aware, this process and its inclusion in the S-SDLC proves to be a sound solution for identifying potential risk before development in an efficient and effective manner given the rapid prototyping and accessibility of evaluation methods. Alternatively, the previously mentioned "cross-functional team" of user experience, security and software development will produce the desired results thus improving security by design.

Issues were found in creating the alignment in taxonomies rendering the activity difficult and required some minor changes. The purpose was to create a mapping to build a relationship between Nielsen's heuristics and a theoretical vulnerability or risk. This was trivial for "input validation" and "error handling", however "Environment" and "Security Features" were less successful due to ambiguity of interpretation. Although "all" are relatable to error conditions and error handling in some form, the issue experienced was the attributes behind the categories are very concerned with coding and security configuration from a technical perspective despite Tsipenyuk's (2005) attempt to build a flexible language open to adaptation. To overcome this, a narrative meaning was interrupted from the categories descriptions.

For example, "Using components with known vulnerabilities"[13]. This narrative suggests to remove unused dependencies, features and components. This was mapped to the heuristic "Aesthetic and Minimalist Design"[14] recommending interfaces should not contain elements which are not relevant or necessary to the user. Tsipenyuk's (2005) mapping of this risk and also security misconfiguration to the category "environment" proved the best fit.

However, the impact of Broken Authentication[15] as per OWASP is the compromise of data that should be protected, suggesting to use strong authentication controls and password systems. Feedback from the heuristic evaluation defined password protection and trust issues against the heuristic Visibility of System Status[16]. The user was not very trustful of how password protection features were operating and communicated within the interface. The heuristic promotes the concept of providing visibility of all system status and states to the user allowing a sense of trust while completing their tasks. Mapping visibility of system status to the 7 Pernicious Kingdoms (Security Features) through broken authentication did not appear confident, however served our purpose for this research objective and may benefit from further work. As the risk score used in the evaluation was not defined by the mapping to "Security Features" it was felt it did not directly affect the outcomes of the primary research goal.

7 Conclusion and Future Work

As discussed in this research, human error is a problem that is not going away, with consistent increase in human error threats and associated costs. Common software development methodologies were evaluated to identify gaps in the process. 102 participants were surveyed and 4 participants performed a usability test to further substantiate that user experience testing in software design can be used to mitigate potential software security risks.

Understanding the end-user facilitates solution designers and software developers to concentrate on what is truly required by the end-user to complete a task or function; reducing the impact of unnecessary features, distractions or, interface elements that hinder a user's experience.

The object of this research was to identify a gap in software development where design can assist in reducing potential weaknesses or vulnerabilities.

This research has concluded that the Human-Centred Design methodology when applied, assists in the design of a solution to reduce potential weaknesses or vulnerabilities in software. It does this by mitigating, through the redesign of, or removal of elements from the interface when proven not to add value to the user's experience. It was observed that security risks can

[13] https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities
[14] https://www.nngroup.com/articles/ten-usability-heuristics/
[15] https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication
[16] https://www.nngroup.com/articles/visibility-system-status/

be uncovered and treated by evaluating the users experience and paying attention to the feedback reported.

This research recommends organisations to employ Human-Centred Design in their processes, particularly during the early stages of software solution development. It is not seen as a replacement for other security testing methods. However, it will expose the potential for human error early in the lifecycle which will be beneficial for mitigating these types of risks.

Its ease of use and accessibility clearly renders it a useful tool. This research should be further explored to identify implications of onboarding the process in established practices. This should involve the social economic and financial impacts. A starting point would be to investigate an efficient communications method or a platform to automatically detect and document common security issues in interface design while rapid prototyping and suggest recommendations to improve the user experience.

A comprehensive taxonomy should be compiled to align Nielsen's heuristics to common scenarios, vulnerabilities and attack categories. This may require broadening categories as defined by OWASP and other existing taxonomies; by introducing new terminologies to facilitate this. Finding common meaning and narrative language would benefit the user experience designer in communicating with a security audience.

References

Verizon (2019) *2019 Data breach investigations report*. Available at: https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf [Accessed 17 September 2019].

Verizon (2020) *2020 Data breach investigations report.* Available at: https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf [Accessed 19 April 2020].

Andrews, G. (2017) 'Better security by design', *Thoughtworks.com*, 24 November. Available at: https://www.thoughtworks.com/insights/blog/better-security-design? [Accessed 18 April 2019].

White, C. (2020) 'What does a UX designer actually do?, *Careerfoundry.com*, 27 January. Available at: https://careerfoundry.com/en/blog/ux-design/what-does-a-ux-designer-actually-do/ [Accessed 18 May 2020].

Fortinet (2019) *2019 Insider Threat report*. Available at https://www.fortinet.com/content/dam/fortinet/assets/threat-reports/insider-threat-report.pdf [Accessed 19 January 2020].

Singh, V. (2020) 'Top 7 SDLC Methodologies', *Hackr.ie*, 09 April. Available at: https://hackr.io/blog/sdlc-methodologies [Accessed 15 May 2020].

Norman, D. (2018) *Principals of Human-Centered Design (Don Norman*). Available at: https://www.youtube.com/watch?v=rmM0kRf8Dbk [Accessed 12 July 2019].

Rizzo, A., Parlangeli, O., Marchigiani, E. and Bagnara, S. (1996) 'The Management of human errors in user-centered design', *ACM SIGCHI Bulletin*, 28 (3), pp. 114-118.

Vassilakaki, E. and Johnson, F. (2015) 'The use of grounded theory in identifying the user experience in search.', *Library and Information Science Library and Information Science Research*, 37(1), pp. 77-87.
Oxford University Press (2020) 'Security'*, Oxford Learner' s Dictionaries.* Available at: https://www.oxfordlearnersdictionaries.com/definition/american_english/security [Accessed 27 June 2020].

Tunggal, A. (2020) 'What is a vulnerability?', *UpGuard.com*, 28 May. Available at: https://www.upguard.com/blog/vulnerability [Accessed 29 May 2020].

Pompon, R. (2017) 'Where do vulnerabilities come from?', *F5 Blog*, 26, September. Available at: https://www.f5.com/labs/articles/cisotociso/where-do-vulnerabilities-come-from [Accessed 1 April 2019].

Beaver, K. (2013) 'If you can't explain it simply, you don't understand it well enough', *Rapid7 Blog*, 15 July. Available at: https://blog.rapid7.com/2013/07/15/if-you-can-t-explain-it-simply-you-don-t-understand-it-well-enough/ [Accessed 15 April 2020].

Blosch, M. and Burton, B. (2017) *Take a Human-Centered Approach to Digital Design*. Available at: https://www.gartner.com/document/3614217 [Accessed 10 December 2019].

Norman, D. (2019) *Observe, Test, Iterate, and Learn (Don Norman)*. Available at: https://www.youtube.com/watch?time_continue=5&v=JgPppwsocRU&feature=emb_logo [Accessed 10 December 2019].

Nguyen, J. and Dupuis, M. (2019) 'Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations.*' In The 20th Annual Conference on Information Technology Education (SIGITE '19)*, pp. 93–98.

Syrossian, D. (2018) 'Security by Design principles according to OWASP', *Threatpost.com*, 27 March. Available at: https://blog.threatpress.com/security-design-principles-owasp/ [Accessed 23 January 2020].

Mailloux, L., Beach. P. and Span, M. (2018) 'Examination of Security Design Principles from NIST SP 800-160', in *2018 Annual IEEE International Systems Conference (SysCon),* Vancouver, British Columbia, Canada, 24-26 April 2018, Available at: https://www.researchgate.net/publication/325495586_Examination_of_security_design_principles_from_NIST_SP_800-160 [Accessed 24 January 2020].

Warsinske, J., Graff, M., Henry, K., Hoover, C., Malisow, B., Murphy, S., Oakes, C., Pajari, G., Parker, J., Seidl, D. and Vasquez, M., (2019) *The Official (ISC)2 Guide to the CISSP CBK Reference.* 5th ed. New Jersey: Wiley.

M. Dupuis and F. Khan, "Effects of peer feedback on password strength," *2018 APWG Symposium on Electronic Crime Research (eCrime)*, San Diego, CA, 15-17 May 2018, pp. 1-9, doi: 10.1109.

Goslin, H. (2020) ‘Weighing Pros and Cons to Select AppSec Testing Types’, *Intro to AppSec*, 2 March. Available at: https://www.veracode.com/blog/intro-appsec/weighing-pros-and-cons-select-appsec-testing-types [Accessed 10 April 2020].

Babich, N. (2017*) What Does a UX Designer Actually Do?*. Available at: https://adobe.com/what-does-a-ux-designer-actually-do/ [Accessed 1 May 2019].
Kaley, A. (2018) ‘UX Debt: How to Identify, Prioritize, and Resolve’, *Nielsen Norman Group*, 11 November. Available at: https://www.nngroup.com/articles/ux-debt/ [Accessed 7 November 2019].

Nielsen, J. (2012) ‘How Many Test Users in a Usability Study?’, *Nielsen Norman Group*, 3 June. Available at: https://www.nngroup.com/articles/how-many-test-users/ [Accessed 7 November 2019].

Kaley, A. (1994) ‘10 Usability Heuristics for User Interface Design’, *Nielsen Norman Group*, 24 April. Available at: https://www.nngroup.com/articles/ux-debt/ [Accessed 7 November 2019].

Magalhaes, M. (2018) ‘Security vs. usability: Does there have to be a compromise?’, *Techgenix.com*, 20 December. Available at: http://techgenix.com/security-vs-usability/ [Accessed 7 April 2019].

Tsipenyuk, K., Chess, B. and McGraw, G. (2005) ‘Seven pernicious kingdoms: a taxonomy of software security errors’, *IEEE Security & Privacy*, 3(6), pp. 81-84.

McGraw, G. (1998) ‘Testing for security during development: why we should scrap penetrate-and-patch’, *IEEE Aerospace and Electronic Systems Magazine*, 13(4), pp. 13-15.

Bird, J. (2018) ‘Building truly cross-functional teams’, *uxdesign.cc,* 5 March. Available at: https://uxdesign.cc/back-to-the-basics-cross-functional-teams-7fa12ba6f5ed [Accessed 20 November 2019].

Oślizło, D. (2019) 'What Is UX Design and What Are the Benefits for Your Business', *Netguru.com*, July 16. Available at: https://www.netguru.com/blog/ux-is-not-just-about-user-experience-popular-misconceptions-about-ux-design [Accessed 20 July 2019].

Miller, G. J. (2013) ‘Agile problems, challenges, & failures’, in *PMI® Global Congress*. New Orleans, Louisiana, Available at: https://www.researchgate.net/publication/335475075_Agile_problems_challenges_failures [Accessed 22 November 2019].

Chervenkova, M. (2019) ‘When Does Agile Fail? Challenges, Problems, and Issues with Agile’, *Lean/Agile*, December 4. Available at: https://kanbanize.com/blog/agile-challenges/ [Accessed 18 January 2020)].

Ponemon Institute (2020) *2020 Cost of Insider Threats Global Report.* Available at: https://www.observeit.com/ponemon-report-2020-cost-of-insider-threats-global-cyberwire/ [Accessed 30 July 2020].

Ponemon Institute (2018) *2018 Cost of Insider Threats Global Report*. Available at: https://www.observeit.com/ponemon-report-2018-cost-of-insider-threats-global-cyberwire/ [Accessed 30 July 2020].

Alashe, O. (2020) 'Major data breaches are far too common, with human error often being a cause', *Silicon Republic*, 7 April. Available at: https://www.siliconrepublic.com/companies/oz-alashe-cybsafe-data-breaches [Accessed 20 April 2020].

Farrell, S. (2017) 'UX Research Cheat Sheet', *NNGroup.com*, 12 February. Available at: https://www.nngroup.com/articles/ux-research-cheat-sheet/ [Accessed 18 July 2019].

OWASP (2020) *Business logic vulnerability.* Available at: https://owasp.org/www-community/vulnerabilities/Business_logic_vulnerability [Accessed 5 May 2020].

OWASP (2020) *The OWASP Testing Project*. Available at: https://owasp.org/www-project-web-security-testing-guide/latest/2-Introduction/ [Accessed 10 July 2020].

Whittle, J., Wijesekera, D. and Hartong, M. (2008) 'Executable misuse cases for modeling security concerns', in *2008 ACM/IEEE 30th International Conference on Software Engineering*, Leipzig, Germany, 10-18 May 2008, pp. 121-130, doi: 10.1145/1368088.1368106.

Appendix A – User/Participants feedback form

Interface Usability Heuristic Evaluation Form

This form will be completed by a participant and used for every Issue experienced. The participant is to summarise the issue name, give a severity rating based on how the issue effects its experience, assign a heuristic category and provide a short qualitative description of the experience offering suggestive feedback.

* Required

1. Participant user number *

Enter your answer

2. Task *

- 1 Make an assumption of what the function of the application is
- 2 Send a file from your desktop to the email recipient
- 3 Send a new file from your desktop to the email recipient
- 4 Send 3 files from your dsktop to the email recipient ray.obrieen@gmail.com - verify you have uploaded the correct files before sending
- 5 Send a password protected file to the email recipient ray.obrien@gmail.com - assume the file is highly sensitive
- 6 Send a file from your desktop to the email recipient ray.obrien@gmail.com and set the expirery to 15 days
- 7 Send a file from your desktop to the email recipient ray.obrien@gmail.com and set

the expiry to 10 days

- 8 Send a file from abc@gmail.com using a custom link
- 9 Send a file from abc@gmail.com using a custom link + add special characters (e.g. *&<>?/) - you can use e.g. <script>alert("hello")</script> in the file name
- 10 Send a file from abc@gmail.com using a custom link you used in task 8

3. Issue: *

Enter your answer

4. Severity Rating (0. None, 1. Cosmetic, 2. Minor, 3. Major, 4. Critical *

	0	1	Minor 2	Major 3	Critical 4
Severity Rating	○	○	○	○	○

5. Heuristic *

- Aesthetic and minimalist design
- Visibility of System Status
- Help users recognize, diagnose, and recover from errors
- Preventing User Errors
- User control and freedom
- Recognition rather than recall
- Flexibility and efficiency of use
- Help and documentation
- Match between system and the real world

Consistency and standards

6. Description *

Enter your answer

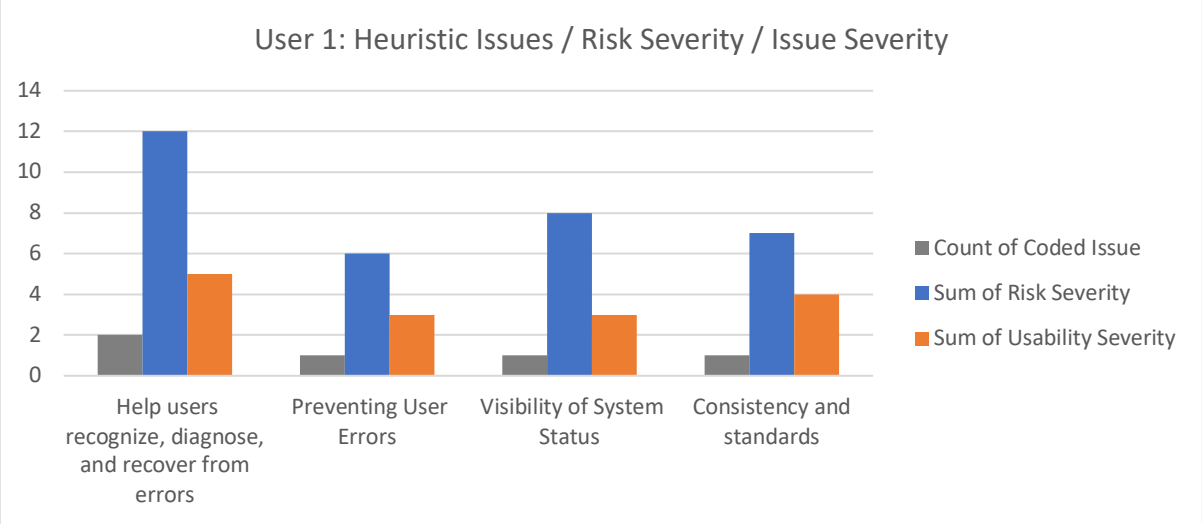
Submit

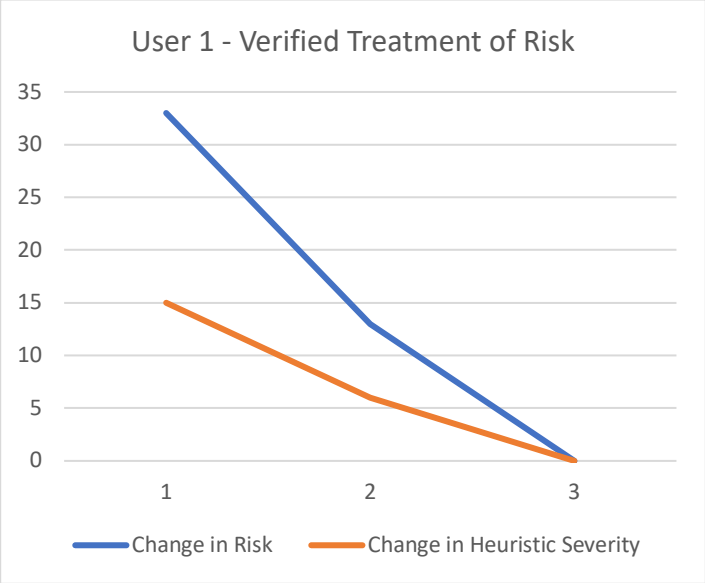
Never give out your password. Report abuse

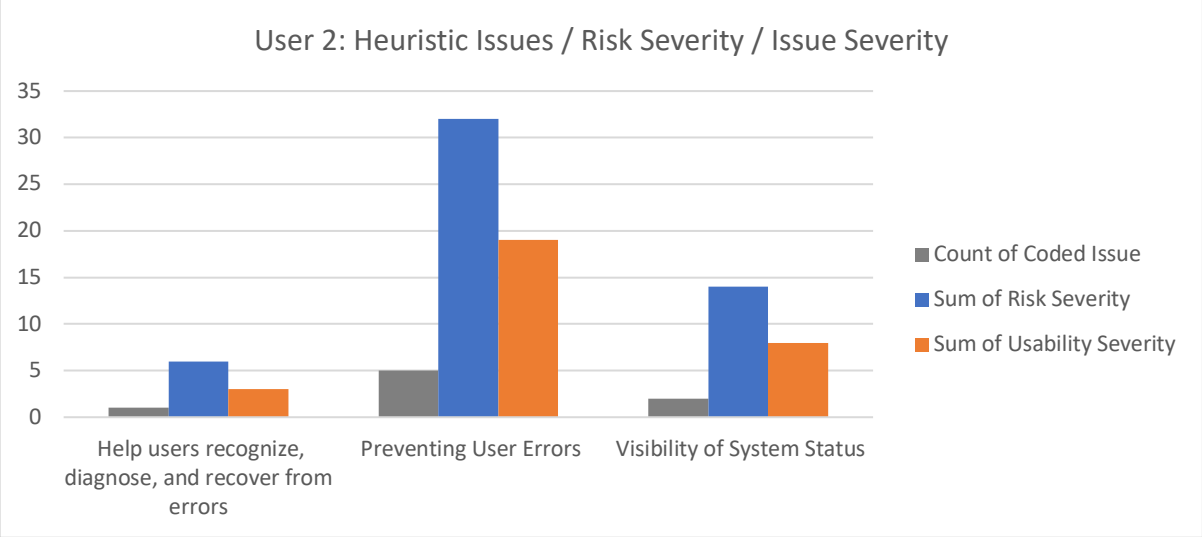
This content is created by the owner of the form. The data you submit will be sent to the form owner.

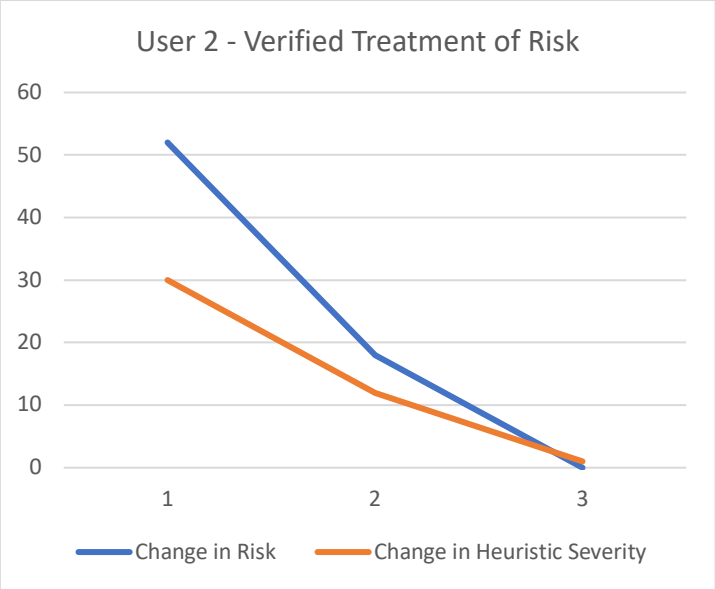
Powered by Microsoft Forms | Privacy and cookies | Terms of use

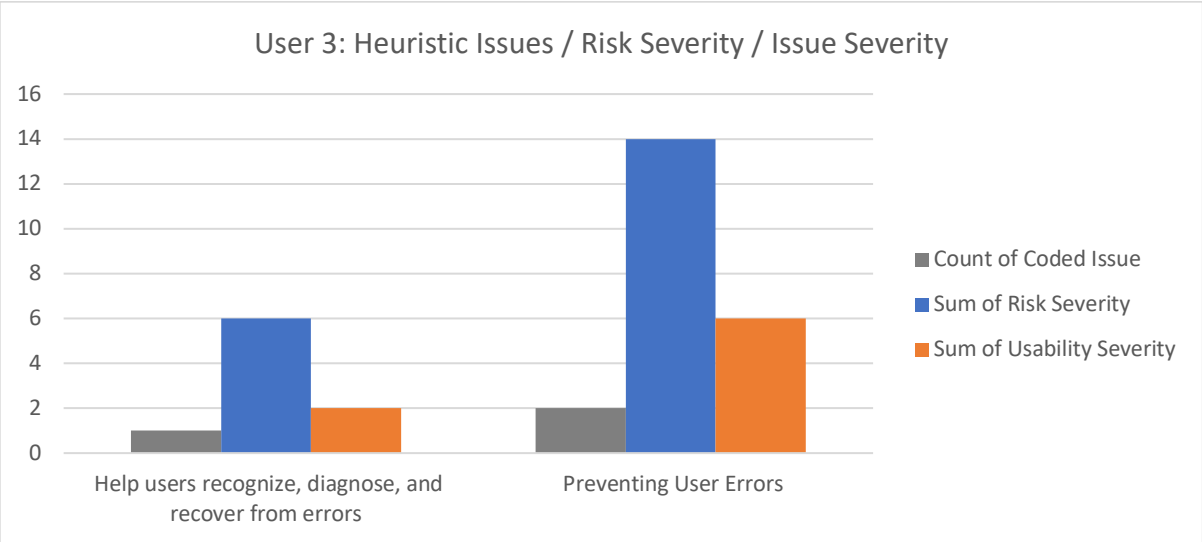
Appendix B – Evaluation graphs

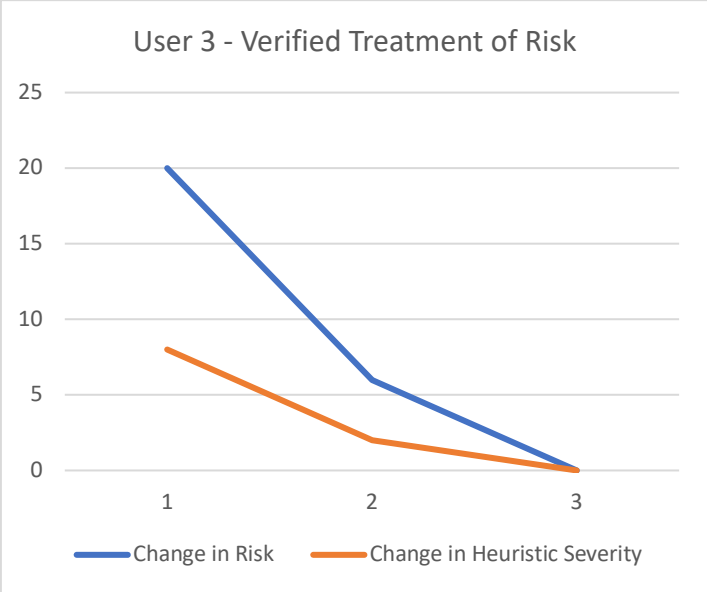


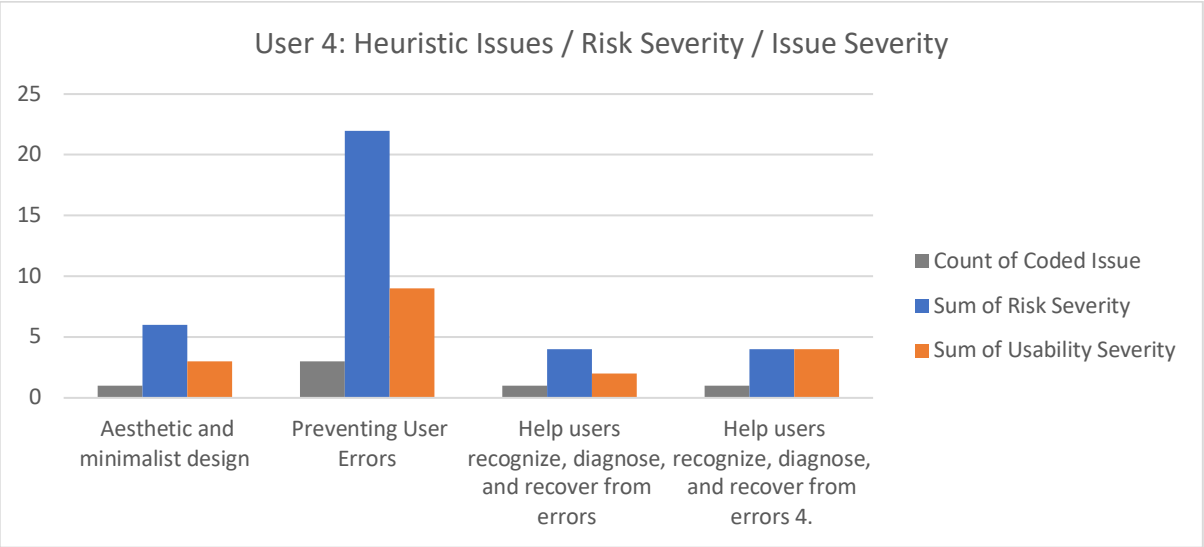


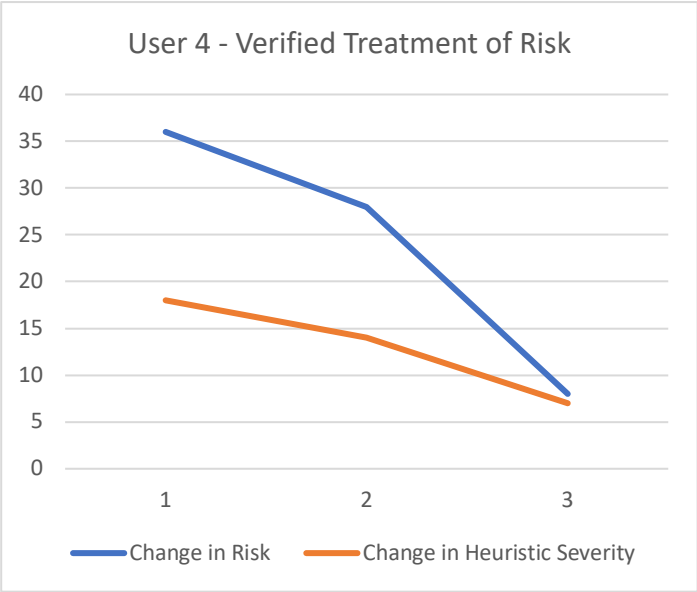

User 2: Heuristic Issues / Risk Severity / Issue Severity
35
30
25
20
15
10
5
0
Help users recognize, diagnose, and recover from errors
Preventing User Errors
Visibility of System Status
Count of Coded Issue
Sum of Risk Severity
Sum of Usability Severity


User 2 - Verified Treatment of Risk
60
50
40
30
20
10
0
1
2
3
Change in Risk
Change in Heuristic Severity

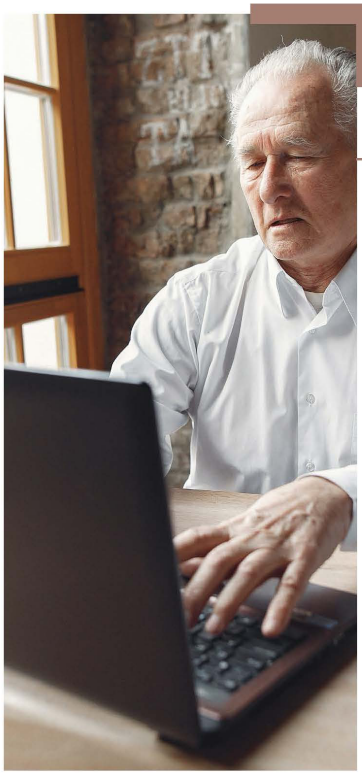

User 3: Heuristic Issues / Risk Severity / Issue Severity
16
14
12
10
8
6
4
2
0
Help users recognize, diagnose, and recover from errors
Preventing User Errors
Count of Coded Issue
Sum of Risk Severity
Sum of Usability Severity


User 3 - Verified Treatment of Risk
25
20
15
10
5
0
1
2
3
Change in Risk
Change in Heuristic Severity


User 4: Heuristic Issues / Risk Severity / Issue Severity
25
20
15
10
5
0
Aesthetic and minimalist design
Preventing User Errors
Help users recognize, diagnose, and recover from errors
Help users recognize, diagnose, and recover from errors 4.
Count of Coded Issue
Sum of Risk Severity
Sum of Usability Severity


User 4 - Verified Treatment of Risk
40
35
30
25
20
15
10
5
0
1
2
3
Change in Risk
Change in Heuristic Severity

Appendix C – User Personas



Persona 1: Christopher Donovan

"sharing data securely is of major importance"

BIO

Christopher is the CFO for one of Ireland's top pharmaceutical manufactures. Working in this role for the last 13 years he knows the high stakes involves with sharing accurate information with external auditors and regulators.
With such a large volume of data to share, Christopher must depend on alternative services. While these services are recommended internally by his company, his paranoia about the service's security remains.

DEMOGRAPHICS

- Male
- 62
- Chief Financial Officer
- Computer proficiency: High
- Dublin, Ireland

FRUSTRATIONS

- Cannot send files directly through his email - must rely on another service
- He is extremely nervous about sending sensitive financial information

NEEDS & WANTS

- A reliable and secure service to share sensitive financial documentation
- To be reassured that the files being sent can only be viewed by the receiver
- To ensure documentation sent cannot be tampered with

Persona 2: Jane Blakely

"Reviewing files is a must - but is never an option.
Double checking requires starting all over. it's very inefficient! "

BIO

As a graphic designer for a global manufacturing company, Jane is well-versed in a multitude of different softwares. Her daily workload is fast paced and demanding; having requests flung her way from various stakeholders, both internal and external to the organization.
File sizes tend to be large and can greatly vary in sensitivity levels.
Jane's I.T. support is limited and colleagues are not I.T. savvy; leading her to depend on the most straightforward file share method she can find.

DEMOGRAPHICS

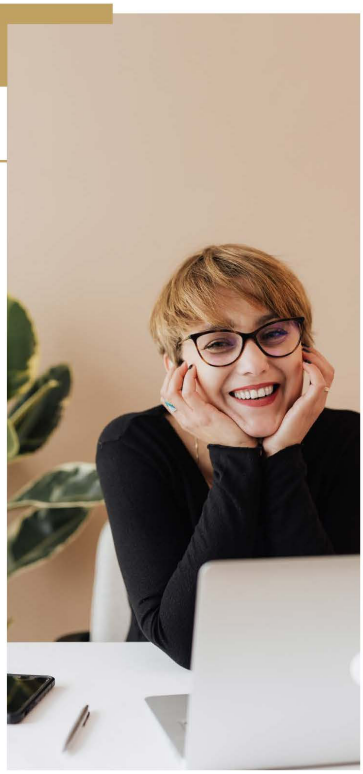
- Female
- 28
- Graphic Designer
- Computer proficiency: High
- Brighton, England

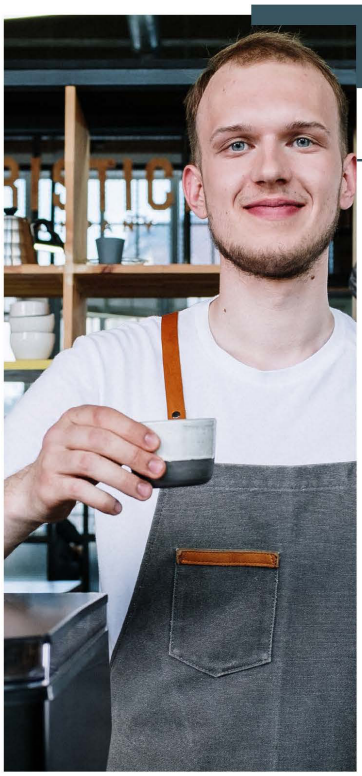
FRUSTRATIONS

- Non-intuitive designs
- With multiple iterations of designs and files to share, hates that there is no reviewing of files available before sending

NEEDS & WANTS

- Share information securely
- Requires straightforward process
- Wants the ability to preview files easily before sending to ensure corrected documentation is attached





Persona 3: Ivan Wilkins

"I hate unfamiliar practices - especially when they could be to my detriment"

BIO

Ivan is a devout coffee lover. Dedicating his life to coffee beans, he established his cafe in his home city, Galway, 4 years ago.
While Ivan is computer literate, he spends his days away from the screen. Preferring the outdoors when he gets the free time.
Since the lockdown due to covid-19, the HSA have imposed regulations to follow in order for him to reopen his doors and stay open.
The stress of potentially losing his business, now combined with the 'dreaded' new practice of continuously providing the authorities with paperwork have Ivan nervous of making any errors.

DEMOGRAPHICS

- Male
- 34
- Cafe owner
- Computer proficiency: Medium
- Galway, Ireland

FRUSTRATIONS

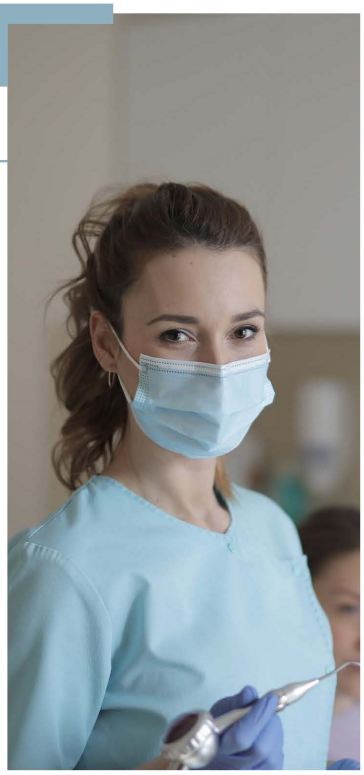
- Prefers hands on activities - finds documentation work tedious.
- Does not like interacting with computers.
- Already burdened with new regulations - is worried about the reliability (security) of chosen sharing platform.

NEEDS & WANTS

- Send compliance reports to the HSA securely and regularly
- Wants an uncomplicated interface so he 'knows' he can't make a mistake.
- Wants a time efficient procedure

Persona 4: Sara Mullens

"Clients privacy has always been a priority for the practice"



BIO

Sara is local dentist in the Carlow area.
Knowing her patients on both a personal and professional level has made her a favoured dentist within the practice.

On somewhat of a regular basis Sara is required to transfer patient files between consultants. Sending highly sensitive files always makes Sara nervous as one wrong move would lose her patients trust and ripple through the community.

DEMOGRAPHICS

- Female
- 41
- Dentist
- Computer proficiency: Medium
- Carlow, Ireland

FRUSTRATIONS

- As files are patient sensitive, Sara hates not having the ability to review documents before she shares them
- Sara finds it frustrating that she cannot retract a file share if necessary

NEEDS & WANTS

- To share medical information between consultants and patients as required.
- Wants to review files before sending
- Wants to be able to cancel link to shared documents if required