Bilal Naqvi

TOWARDS ALIGNING SECURITY AND USABILITY DURING THE SYSTEM DEVELOPMENT LIFECYCLE

ACTA UNIVERSITATIS LAPPEENRANTAENSIS 933

Bilal Naqvi

TOWARDS ALIGNING SECURITY AND USABILITY DURING THE SYSTEM DEVELOPMENT LIFECYCLE

Dissertation for the degree of Doctor of Philosophy to be presented with due permission for public examination and criticism in the Room 1318 at Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland on the 17th of November 2020, at noon.

Acta Universitatis
Lappeenrantaensis 933

Supervisor Professor Jari Porras
LUT School of Engineering Science
Lappeenranta-Lahti University of Technology LUT
Finland

Reviewers Associate Professor Carmelo Ardito
Department of Electrical and Information Engineering
Politecnico di Bari
Italy

Associate Professor Lene Tolstrup Sørensen
Department of Communication, Media, and Information technologies
Aalborg University
Denmark

Opponent Associate Professor Janne Lindqvist
Department of Computer Science
Aalto University
Finland

ISBN 978-952-335-585-9
ISBN 978-952-335-586-6 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenranta-Lahti University of Technology LUT
LUT University Press 2020

Abstract

Bilal Naqvi
Towards Aligning Security and Usability During the System Development Lifecycle
Lappeenranta 2020
86 pages
Acta Universitatis Lappeenrantaensis 933
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-585-9, ISBN 978-952-335-586-6 (PDF)
ISSN-L 1456-4491, ISSN 1456-4491

Security and usability are considered to be mutually antagonistic goals. Conflicts arise when recommendations from security and usability perspectives contradict each other. Academic research and industrial practices have revealed that conflict management mainly relies on the skill of the developers. Expertise in both security and usability is difficult to find in one person; therefore, there is a need to support developers when they attempt to manage conflicts.

This research investigates the gaps in research and industrial practices concerning the alignment between security and usability. More importantly, this research investigates how conflicts can be effectively managed during the system development lifecycle. This research proposes the use of design patterns to support the developers in management of the conflicts. Besides other information each pattern encapsulates problem statement, suitable trade-off (the solution), and context of use. The work performed during this dissertation led to the creation of different artefacts that enable identification and documentation of design patterns. Each identified artefact has a context in which it can be applied for identification of design patterns.

This research was conducted based on the principles of design science research. The identified artefacts are listed and discussed in the body of this dissertation. Moreover, various data collection methods, including surveys, interviews, and workshops, were utilised to rationalise and validate this research when applicable.

This research contributes to alignment between security and usability in the system development lifecycle. The key findings are as follows: (1) security and usability can be synergised by managing their conflicts during the system development lifecycle as early as possible; (2) the conflicts can be better understood at the level of the sub-characteristics of security and usability; and (3) the artefacts (formulated during this research) can be helpful for developing a catalogue of usable security design patterns, and the patterns can be used to influence the decision-making of developers and designers in similar contexts.

Keywords: design patterns, security, system design, system development lifecycle, usability, usable security

Acknowledgements

I would like to take this opportunity to thank my supervisor, Professor Jari Porras, for his able support and guidance during my doctoral studies. In addition to research, Jari provided excellent support in administrative matters.

I also appreciate the efforts and support of Professor Ahmed Seffah, who admitted me into the doctoral programme and guided me during the initial stages of this dissertation. I also acknowledge and thank Professor Nathan Clarke and Professor Alain Abran for their valuable guidance and feedback on my research.

I would like to express my gratitude to my colleagues in the Software Engineering Department at LUT, especially Professor Kari Smolander. He provided support in matters within his domain, particularly after he became head of the department. I acknowledge and thank Dr. Annika Wolff for her feedback on my articles and survey questionnaires. Special thanks to the department’s secretary, Tarja Nikkinen, for taking care of all administrative matters and enabling me to complete my degree in a timely manner.

I also wish to thank Shola Oyedeji for being there and supporting me through the ups and downs of my doctoral studies. Thanks to my friends, Mehar Ullah, Ali Saud, Omer Sher, and Andrei Saltan, for being there whenever I needed them.

This dissertation would not have been completed without the prayers, love, and support of my exceptional parents. I wish to acknowledge my father, who has followed my academic progress since junior school. I thank my mother for giving meaning to my life and teaching me how to approach challenges. I also thank my brother, who has provided exceptional support throughout my life. I love you all!

Finally, I wish to acknowledge the support of my wife, who has been raising our child back home in Pakistan. I appreciate all your efforts in making this process easy for me and for all your words of encouragement throughout my studies. Last but not the least, I appreciate my daughter, who missed her father during this time. Fatima, baba loves you!

Bilal Naqvi
November 2020
Lappeenranta, Finland

To my love, my wife Sara!

Contents

List of publications

This dissertation is based on following publications, referred in the text as Publications I–V.

I. Naqvi, B., Clarke, N. and Porras, J. (2020). Incorporating the human facet of security in developing systems and services. *Information and Computer Security,* (in press).

II. Naqvi, B., Seffah, A. and Abran, A. (2020). Framework for examination of software quality characteristics in conflict: A security and usability exemplar. *Cogent Engineering*, 7(1), 1788308.

III. Naqvi, B., Porras, J., Oyedeji, S. and Ullah, M. (2020). Towards identification of patterns aligning security and usability. In: Abdelnour Nocera, J. *et al*., eds. *Beyond Interactions: INTERACT 2019 IFIP TC 13 Workshops, Paphos, Cyprus, September 2–6, 2019, Revised Selected Papers*. Lecture Notes in Computer Science. Vol. 11930. Cham: Springer, pp. 121–132.

IV. Naqvi, B. and Porras, J. (2020). Usable security by design: A pattern approach. In: Moallem, A. ed. *HCI for Cybersecurity, Privacy and Trust: Second International Conference, HCI-CPT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings*. Lecture Notes in Computer Science. Vol. 12210. Cham: Springer pp. 609–618.

V. Naqvi, B. and Seffah, A. (2019). Interdependencies, conflicts and trade-offs between security and usability: Why and how should we engineer them? In: Moallem A. ed. *HCI for Cybersecurity, Privacy and Trust: First International Conference, HCI-CPT 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings*. Lecture Notes in Computer Science. Vol. 11594. Cham: Springer, pp. 314–324.

Author's contribution

I am the principal author of Publications I–V. I carried out the literature review, planning and execution of data collection as well as analyses and reporting under the supervision of my supervisor. I was involved in conducting interviews and workshops with the industrial partners, which led to the creation and validation of the framework presented in Publication I. I was also involved in conducting surveys and other evaluations, which led to validation of the framework presented in Publication II. For Publications III–V, in addition to being the principal author, I made presentations at relevant conferences to get feedback on the artefacts.

Nomenclature

CAPTCHA	Completely Automated Public Turing Test to Tell Computers and Humans Apart
DSR	Design science research
DSRM	Design science research methodology
EQ	External quality
HCI	Human–computer interaction
IFUS	Integrated framework for usable security
ISO	International Organization of Standardization
NIST	National Institute of Standards and Technology
PoDF	Pattern-oriented design framework
QinU	Quality in use
SDLC	System development lifecycle
UI	User interface
UX	User experience

1 Introduction

Security and usability are considered essential quality characteristics in today’s software systems (International Organization for Standardization [ISO], 2011). Academic research and industrial practices identify that security and usability are often in conflict, and identifying suitable trade-offs between the two is not an easy task (Caputo *et al*., 2016; Naqvi and Seffah, 2018). A conflict in this dissertation refers to a contradiction between recommendations from a security point of view (in terms of its goals such as confidentiality, integrity and authenticity [CIA]) and recommendations from a usability and user experience (UX) point of view (in terms of their elements such as effectiveness, efficiency and satisfaction, to mention a few). Standards and best practices concerning these two characteristics provide guidance from the perspective of only one, without reference to the possible impact on other, for example, ISO 27001, ISO 9241-210 (ISO 2013, 2019). Consequently, there are recommendations from a security perspective that negatively impact the usability of the system, and vice versa. Examples include the following:

- Text passwords, which feature a conflict between authentication (a security goal) and memorability (an element of usability and user experience). From a security perspective, passwords should be sufficiently long, frequently changed and have different cases and special characters. However, from a user (usability) point of view, such passwords are hard to memorise and place cognitive load on the user. Thus, if the suggested security guidelines are implemented, they have an adverse impact on the usability of the system, and if they are not implemented, the system’s security may be compromised (Garfinkel and Lipford, 2014).
- Password masking, which features a conflict between confidentiality (a security goal) and feedback (an element of usability). Password masking is implemented in most authentication mechanisms to protect against shoulder surfing, but at the cost of feedback. In the case of a mistake, a legitimate user has to re-type the complete password without knowing what was wrong (Sasse *et al*., 2016).

Similarly, some recommendations from the usability perspective negatively impact the security of the system. Examples include the following:

- The location awareness capability of various mobile devices, which features a conflict between desirability, effectiveness (elements of UX and usability, respectively) and privacy (a security goal). Various applications on mobile devices require access to location data to provide personalised and desirable content. However, such features increase the threat of unauthorised dissemination of location information, which could also compromise users’ security.
- Behavioural advertising, which features a conflict between desirability (an element of UX) and privacy (a security goal). Information about user preferences

is recorded so that the user is presented with only the ads they want to view. However, this has security implications because the information gathered in this way could be used for phishing and ransomware attacks, negatively impacting the overall security of the system and infrastructure (Garfinkel and Lipford, 2014).

Having discussed the interplay between security and usability, it is relevant to note that security and usability are handled by different teams during the software development phase of the system development lifecycle (SDLC) (Naqvi and Seffah, 2019). Moreover, it is difficult to find a person who has expertise in both security and usability since these have evolved independently as two different domains. Security engineers and developers are trained to handle security concerns, and their focus is on making the system's security robust against external and internal threats. For them, usability is a minor concern. Similarly, the focus of user interface (UI) and UX designers and developers is on improving users' interaction and experience with the system, which might introduce loopholes from a system security perspective. The domain of research focused on the interdependencies between security and usability and the alignment of these two characteristics in the development of various systems and services is known as *usable security*.

To cater rapidly evolving cyber threats to the infrastructure and dire consequences in terms of money, reputation, and lawsuits, systems and services have security features that are difficult for normal users to comprehend and use. This has caused a greater number of cyber-attacks triggered by humans (IBM, 2018). Therefore, it is relevant to consider the usability elements in security design as key factors affecting cyber hygiene (Kirlappos and Sasse, 2014). Otherwise, even if they are secure against external threats, security systems could be susceptible to:

- user mistakes, ultimately leading to system compromise;
- increased user disengagement and frustration; and
- user-implemented workarounds (Glass *et al.*, 2016). For instance, in the case of a complex password-based authentication mechanism, users may employ unwanted techniques like pretexting and reusing credentials whenever possible.

This research advocates for the concept of usable security by design and is directed towards the creation of artefacts for management of security and usability conflicts within the scope of SDLC. The management of conflicts includes activities such as identifying the conflict, eliciting a suitable trade-off, documenting the problem, context of use, and the solution (suitable trade-off) in form of a design pattern for dissemination among designers and developers to influence their decision-making abilities in similar contexts. This can help avoid the need to repeat work and thus save a significant amount of money and effort in comparison to cases in which security and usability conflicts are identified later during the development life cycle. Moreover, use of patterns for documenting and disseminating suitable trade-offs align with the engineering practice of not reinventing the wheel. Patterns provide proven design solutions and guidance on the context of their usage. They provide real solutions rather than abstract principles by explicitly mentioning the context and problem and summarising the rationale for their effectiveness. Since the pattern provides a generic 'core' solution, its use can vary between implementations (Naqvi and Seffah, 2019). The goals of this dissertation are as follows:

- To design and develop artefacts within the scope of the SDLC to support developers and designers in managing security and usability conflicts.
- To formulate a cross-disciplinary communication mechanism for security and usability developers that influences their decision-making about conflicts in similar contexts of use, thereby supporting the concept of re-usability.

1.1 Objectives

The objectives of this dissertation are as follows:

Obj-1: To understand the state of the art concerning the nature of dependencies between security and usability.

Obj-2: To assess the approach of using design patterns to disseminate problem, context, and solution (suitable trade-off) besides other information regarding usage, among developers and designers to support them in managing conflicts.

Obj-3: To formulate artefacts (processes and patterns) based on elements of design science research (DSR) to support developers and designers in managing conflicts.

1.2 Research Questions

The main research question addressed in this dissertation is as follows: *how can the conflicts and suitable trade-offs between security and usability be identified and documented before development?*

The following sub-questions help answer this main question:

RQ-1: *What are the gaps in research and industrial practices that lead to conflicts between security and usability?*

RQ-2: *Can design patterns be used to document and disseminate suitable trade-offs for assisting developers in managing conflicts?*

RQ-3: *How can management of the conflicts be governed from identification of conflicts to elicitation and documentation of suitable trade-offs?*

All these questions are considered while formulating artefacts for aligning security and usability. The processes are aimed at providing different means for managing conflicts in the contexts in which they arise, while the patterns are aimed at supporting common developers and designers in handling security and usability conflicts.

1.3 Methodology

The DSR methodology (DSRM) was applied for this dissertation (Peffers *et al*., 2007). In line with this methodology, different artefacts, including processes and patterns, were created to address the main research question. The work performed as part of this dissertation can be categorised into three stages. Figure 1 maps the research questions, objectives, and various stages of this research.

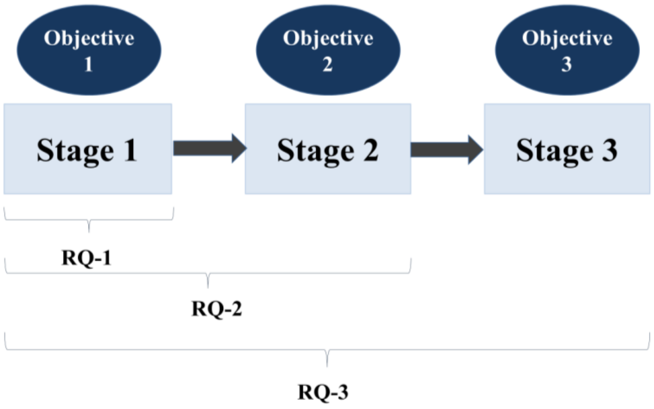


Figure 1. Map of the research question, objectives, and stages

Each stage is described in detail below:

- **Stage 1: Identification of the gaps in research and practice**
  In the first stage of this dissertation, a literature review was conducted to identify the gaps in research, and interviews were conducted with representatives from the industry to identify industrial practices that lead to conflict situations (RQ-1). During the interviews, the roles of (1) project managers, (2) lead architects, (3) security engineers, and (4) UX developers were considered. The data recorded during the interviews were analysed and similar case studies were reviewed to find an answer to vital questions that emerged during this stage. Furthermore, a survey focused on the human aspects of security was conducted to assess the security awareness of the personnel working at a company. Specifically, it aimed

to identify gaps from the perspective of personnel’s knowledge, attitude, and behaviour (KAB) related to different security practices involving humans. The outcomes of this stage were considered while formulating the artefacts to align security and usability during the SDLC. The gaps identified during this stage are reported in Publications I, III, IV and V.

- **Stage 2: Exploring the use of patterns for documenting and disseminating conflicts and suitable trade-offs**

  In the second stage, to investigate the potential use of design patterns for addressing security and usability conflicts (RQ-2), a focused literature review on the use of patterns was conducted. Industry practitioners’ concerns related to this topic were gathered through focused interviews. The use of patterns in this case supports the engineering perspective of not reinventing the wheel.

  Patterns have been shown to be effective as a vehicle to document the best practices for addressing a common design problem. The term ‘pattern’ is considered here as defined by Christopher Alexander, who defined a pattern as ‘each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice’ (Alexander *et al*. 1977, p. x). Patterns support the smooth integration and cross-pollination of communities (Seffah and Javahery, 2004). Patterns are also recommended for improving the communication among team members from different disciplines. They foster the development of a common language or vocabulary for explaining design, and therefore they can be helpful in multidisciplinary fields like usable security.

  A template to document usable security patterns was created during this stage. It was validated by conducting a survey with a group of developers. Details about the pattern’s template and survey are presented in Publication I.

- **Stage 3: Formulation of artefacts for management of conflicts**

  During the third stage, analysis of the data gathered from investigations conducted for RQ-1 and RQ-2 helped to establish a rationale for this research, and enabled formulation of a mechanism to govern the management of conflicts in line with RQ-3. Management of conflicts is a four-step process: (1) identification of the conflict as it arises; (2) modelling of the relationship between characteristics in conflict and illumination of the sub-characteristics; (3) identification of suitable trade-offs by involving practitioners from both domains while conforming to standards and best practices related to security and usability; and (4) dissemination of suitable trade-offs and application of them in similar contexts. Different artefacts formulated during this stage include the following:

- **Integrative Framework for Usable Security (IFUS)**
  The IFUS governs management of security and usability conflicts within the scope of the SDLC and allows security and usability concerns to be incorporated collectively from the requirement engineering stage of the SDLC. The concerns raised after a series of interviews with security and usability practitioners working with our industrial partner served as driving factors in the creation of the IFUS. The key activities of the IFUS are grouped into five distinct phases: analyse, identify, resolve, disseminate and use. During the analyse phase, the requirements are collected and analysed. In the identify phase, goals with respect to both security and usability are identified, leading to the identification of potential conflicts. In the resolve phase, the identified conflicts are resolved and documented as patterns. Then, in the disseminate phase, the patterns are added to the catalogue and disseminated using different mechanisms. In the use phase, the developers use the patterns in similar contexts. The IFUS is presented in Publication I.
- **Pattern Oriented Design Framework (PoDF)**
  The purpose of the PoDF is to assist in the conflict examination of the quality characteristics (in this case, security, and usability). It does so by providing various means for identifying conflicts, modelling conflicts at the level of the sub-characteristics, eliciting a suitable trade-off between conflicting characteristics while documenting suitable trade-offs as patterns for use by developers in the industry. The PoDF has four layers. The first layer, identify, deals with identification of conflicts. Experts can utilise different tools and methods, such as cognitive walkthroughs and surveys, among others for this purpose. Once conflicts are identified, security and usability experts model and quantify the possible relationships between security, usability and their sub-characteristics (i.e. the model and quantify layer). Recommendations from standards on security and usability, internal policies and governmental directives in specific contexts play a key role in the modelling of conflicts. In the build layer, security and usability experts brainstorm and discuss various solutions for eliciting the right trade-offs. Once a suitable solution is identified, it is documented as a pattern. To support reuse, the pattern is added to a catalogue for use by other developers and designers in similar contexts of use. At the apply layer, the software developers apply these patterns to deliver systems that are simultaneously usable and secure. The PoDF is presented in Publication II.

The difference between the pattern-oriented design framework (PoDF) and IFUS lies in the context in which they are applied. The PoDF is applicable to versions of systems and services that have already been deployed, and for which a new version, releases and fixes are desired by the stakeholders. In contrast, IFUS primarily governs the management of conflicts in the development of newer systems.

It is worth noting that one pattern addresses only one conflict in a particular context; therefore, an entire catalogue of usable security patterns needs to be developed to address the various conflicts that appear during the development of systems and services. Thus, during the third stage, an additional question was addressed: what are means for the development of a catalogue of usable security design patterns? Development of such a catalogue can take a lot of time and requires effort from a community. The current dissertation contributes to this cause by presenting different artefacts (methodologies) to assist in development of a catalogue of usable security patterns. For development of a catalogue of patterns, one approach in contrast to IFUS and PoDF is to identify patterns from existing implementations. To do so, two artefacts in the form of methodologies were created based on elements of the DSRM. The prime purpose of these methodologies is to identify usable security patterns from existing implementations that serve as examples of good practices in the field. They are as follows:

- **Three-stage methodology for the identification of usable security patterns**

  This methodology is based on identification of new patterns from existing implementations that serve as examples of good practices in industry. This methodology provides uniform means to identify new patterns and an opportunity for various stakeholders to contribute to identification of the patterns and build a catalogue of usable security patterns. From an industrial perspective, it can enable documentation of new patterns emerging from the implementations of experienced developers, thereby facilitating the learning and training of new developers.

  The first stage involves the selection of a common usable security problem. To do so, experts can utilise, for example, surveys involving end-users, cognitive walkthroughs, or heuristic evaluations. The next step is to identify existing implementations that address the problem. Since implementations can approach a problem in different ways, in order to document the best implementation as a pattern, it is imperative to fulfil the rule of three, according to which there should be at least three instances of similar implementations before a pattern is identified and documented

(Mor *et al.*, 2010). Once three instances of similar implementations for a particular problem are identified, the pattern is documented. The second stage involves review of the newly documented pattern by one or more experts in the field. In the third stage, the expert or experts decide whether to accept, modify or reject the pattern. This methodology is presented in Publication III.

- **Participatory workshop for usable security design patterns**

  To create a validated catalogue of usable security design patterns, a proposal for a participatory usable security design patterns workshop was formulated. The proposed workshop includes two stages. Stage I is focused on identification and documentation of patterns, while Stage II includes validation of the identified patterns. Key activities include (1) distributing the narratives that describe the usable security problem, (2) identifying design patterns using comparative analysis, (3) using scenarios to validate the right context of use for the identified patterns, and (4) documenting the lessons learned and recommendations for future use. These activities are performed in groups (3–5 participants each).

  In stage I, the narratives describing a usable security problem are distributed among the groups, who are tasked with designing their own solutions for the design problem under consideration. The solution developed by each group is subjected to comparative analysis in an attempt to identify instances of good design that can be documented as design patterns. In stage II, the identified design patterns are subjected to validation. The participants are provided with a list of design patterns and the problem scenario. Patterns from the list that are applicable in the context under consideration are selected. The participants are then tasked with documenting a solution derived by applying a particular pattern in a certain context. If the right pattern is applied in the right context, it is validated; otherwise, it is modified. Finally, the lessons learned and recommendations for future use of patterns are documented. The proposal is presented in Publication IV.

1.4 Intended Audience

The intended audiences for this dissertation include the following:

- Young researchers who are interested in the interdependencies between quality characteristics. This dissertation, including the referenced publications, provides knowledge and lessons learned from the example of security and usability, which can be useful for managing conflicts between other quality characteristics.

- Usable security researchers who are trying to investigate the interdependencies, conflicts and trade-offs between security and usability. The dissertation provides ideas that can be built upon as well as opportunities to contribute to the development of a catalogue of usable security patterns.
- Industry practitioners who are involved in conflict management in their day-to-day operations. These practitioners may have different roles, but product owners, security engineers and usability experts may be particularly interested in this topic.
- Reviewers and opponent who are involved in evaluation of the work performed during this dissertation.

1.5 Expected Contributions

This research explores a way to manage conflicts between security and usability by identifying and using design patterns. Though the concept of design patterns is not particularly new, the novelty lies in the application of the DSRM for formulating artefacts (i.e. processes) that can be used to create and identify usable security patterns. Significant contributions of this dissertation include:

- a body of knowledge discussing various gaps in academic research and industrial practices;
- identification of the nature of relationships between security and usability from the perspective different quality views;
- recommendations for handling conflicts at the level of sub-characteristics;
- formulation of different artefacts for management of conflicts from identification of conflicts to their documentation and dissemination as suitable trade-offs;
- formulation of methodologies for creating a catalogue of usable security design patterns; and
- a template to standardize the documentation of usable security patterns.

2 Background and Related Work

Security and usability have evolved independently as different domains. The ISO 25010 standard defines security as ‘degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization’ (ISO, 2011). It is pertinent to state that the perception of security is consistent among all series of standards and all communities with CIA as its main goals. However, usability has been defined differently by different communities (i.e. software engineering and human–computer interaction [HCI]) and by different standards, such as ISO 9126 and ISO 9241-11. There are also two viewpoints on usability in ISO 25010 that are relevant to product quality and quality in use models. However, the definition of usability considered in this dissertation is the ‘degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use’ (ISO, 2011).

Initially, human aspects or the usability of security were considered as limited to the usability of the security interfaces, but over time, it was realised that elements of usability (as identified and defined by standards such as ISO 25010) and UX must be incorporated into security (Morville, 2004). Mahlke (2007) presented a framework for the integration of non-instrumental qualities, symbolic aspects, and emotional user reactions to traditional approaches of interaction. Zagouras *et al*., (2017) presented concepts and definitions regarding the incorporation of elements of UX into security. The authors assert that UX needs to be considered in security design, as this dimension has an impact on the way the user interacts with the system and influences the way it behaves.

To cope with the challenge for incorporating elements of usability and UX into security, researchers studying usable security started to investigate several avenues involving (1) usability issues that arise from user authentication (Zhang *et al*., 2010; Florêncio *et al*., 2014; Kelley *et al*., 2012), (2) usability issues arising from email security and public key infrastructure (Ruoti and Seamons, 2019; Ramsdell and Turner, 2010; Gaw *et al*., 2006), (3) anti-phishing efforts (Lapsey, 2013; Sheng *et al*., 2010, Hong, 2012), and (4) web-privacy and fair information practices (Ur *et al*., 2012; Tsai *et al*., 2011). Garfinkel and Lipford, (2014) list and discuss other avenues of usable security research. With a broader scope of security services (ranging from authentication to email security) and web-based services (such as behavioural advertising), the need for a generalised solution within the scope of SDLC was identified. This dissertation is a step in this direction.

Furthermore, usable security challenges generally take the form of conflicts. It is relevant to note that there are conflicts in each of the areas identified earlier: (1) the study of text passwords features a conflict between authentication (a security mechanism) and memorability (a usability element) (Yildirim and Mackie, 2019; Naqvi and Seffah 2018); (2) various graphical password schemes, in which authenticating the user takes longer than inputting text passwords, feature a conflict between authentication (a security mechanism) and efficiency (a usability element) (Suru and Murano, 2019; Garfinkel and

Lipford, 2014), (3) email security and PKI-based systems feature a conflict between confidentiality (a security mechanism) and understandability (a usability element) (Reuter *et al*., 2020; Garfinkel and Lipford, 2014). However, more conflicts arise in the industry during the development of state-of-the-art systems and services.

An important aspect to note with regard to the challenges posed by usable security is that, as a multi-dimensional field, usable security is being studied by the following communities and interest groups in silos:

1. The usable security community, which is a small community exploring usability/HCI approaches to security and privacy technologies, including access control, authentication, and identity management.
2. The traditional computer security community, which deals with a broad scope of services related to computer and communication technologies. Usability is a minor concern addressed at a surface level within this community.
3. The software engineering community, which has defined usability and security as two of the eight major quality characteristics. Usability is a characteristic of user interfaces, and security is a characteristic of functionality (ISO, 2011).
4. The HCI community, where HCI and UX researchers try to apply a cognitive perspective to explain how users make poor security decisions when faced with non-usable user interfaces. They have concluded that in order for users to make the right security choices, they need user interfaces that are not just usable but that also promote pro-secure behaviour.
5. The information security governance community, which introduces guidelines and policies for the auditing and establishment of usable security at the organisational and managerial levels.
6. The human factors and ergonomics interest group, which places focus on the role of human users in achieving systems' security goals and how the issues in security design can lead to bad security practices on the part of the user.

The integration of findings from different communities leading to a joint effort towards aligning the principles of security and elements of usability still poses a challenge.

2.1 Understanding the Interdependency between Security and Usability

Despite the recognition that security cannot truly be achieved unless it is usable by users (Garfinkel and Lipford, 2014), the state of the art concerning alignment between security and usability has some catching up to do. Some of the key factors associated with security and usability conflicts include the following:

- **Lack of focus on user-centric design:** Green and Smith, (2016) note that despite the argument that security engineers should not see users as a problem, the current practice is that security engineers consider usability and UX to be a significant threat to robust security. Consequently, usability is a minor concern for them, leading to a lack of user-centric design in the implementation of security features.
- **An interdisciplinary job:** Gorski *et al*., (2019) state that security and usability are handled independently, usually in different teams, and the focus is on delivering specific features. For example, the team working on usability is focused on enhancing the UX and making the UI more attractive. In contrast, the team working on security is focused on protecting the system, and thus they may overlook aspects like memorability, learnability, and findability during the development of security services.
- **Varying types of users:** While discussing the challenges posed by usable security, Garfinkel and Lipford (2014) identify the customer challenge, which refers to the fact that the opinions and requirements concerning security differ within the community of users of the same mobile device or application. It is difficult to cater to the requirements of a diverse category of users, which further complicates the tasks of finding common ground between security and usability and delivering a usable, secure system (Publication III).
- **Lack of measures and methods for assessing adequacy:** To assess robustness of security, techniques like vulnerability scanning and penetration testing can be employed to check the robustness of security features. Similarly, for usability, HCI techniques like user feedback on user interfaces can be used to evaluate the appropriateness of usability features. However, there is no such technique for evaluating the adequacy of usable security (Garfinkel and Lipford, 2014). This was identified not only in prior literature but also in the interviews conducted during this dissertation.
- **Ad-hoc nature of the task:** Feth (2015) notes that managing usable security is an ad-hoc task that is performed at different levels. Specifically, it is performed by system engineers during development and by system administrators (in the case of mobile devices, users) during runtime. Feth (2015) further states that at the time of development, user testing helps to understand the needs of the user, but the outcomes are mostly project specific.
- **Constraint to a constraint:** The requirement engineering community defines security as a constraint to a system's functional requirements (Haley *et al*., 2006). This raises a question: if security is a constraint to system requirements, then could the usability of security be considered a constraint to a constraint?

- **Reliance on developers' skills:** Even in an organisational setting, the handling of usable security is dependent upon the skill of developers (Caputo *et al.*, 2016). In their widely cited paper, Whitten and Tygar (1998) identified five properties that make usable security problematic. The authors suggested that there is a need for developers to think from the user's perspective. In addition, designers of the system should not assume that the user will read the manuals for configuration; instead, security features should be easy to use (Whitten and Tygar, 1998).
- **Short development cycles:** In systems like social media services and interactive learning applications, developers face pressure due to short development cycles and an increased demand for usability. Often, this situation results in a compromise regarding security and privacy. Management of the conflict between security and usability is a task in itself, but current practices and development cycles offer no or less focus on management of conflicts; individual requirements, and not their interdependencies, are being considered (Garfinkel and Lipford, 2014).

Having discussed the factors associated with security and usability conflicts, it is worthwhile to discuss gaps in the state of the art. As stated in Chapter 1, these gaps were identified during the literature review and validated during interviews with industry representatives.

2.1.1 Gaps in the state of the art

There are some cases in which security and usability can be enhanced by modelling their mutual relationships. Typical examples include online payment and e-banking services, supervision of critical industrial infrastructures and crisis management. However, the following are key gaps (Publication V):

- **Failure of security specialists to address usability:** One gap advocates the failure of security specialists to address usability as perceived and defined by the HCI community. Historically, security and usability have evolved independently or have been considered two opposite factors. Another historical explanation is that researchers were more driven by technology than by user problems and perceptions of security. For example, the development of identity management technologies was so demanding in terms of security that it left little time and monetary resources to cater to usability and human factors in general (Publication V).
- **Bug-fix-driven behaviour:** The industry's behaviour has been more driven by bug fixing than by trying to consider the context and user experiences in which bugs occur. Therefore, most industry efforts have been focused on automating the

process of reporting and handling bugs rather than looking for human experiences and how they can promote more secure operations overall.

- **Lack of a user-centric approach to security design:** Another gap that demonstrates the lack of alignment between security and usability is the design and innovation approach applied to develop new security technologies. Most often, innovation is initiated by a company developing an in-house technology to address a specific problem that occurs in a specific project. Other groups in the same company or other companies may develop their own versions of these solutions. However, when the original context of applicability is changed, it is difficult to ensure the usability of these in-house solutions and their different versions.
- **Lack of applicability of existing HCI techniques to security:** The lack of applicability of existing HCI techniques to conduct effective user studies for security systems and services is a serious obstacle. Moreover, it is difficult to conduct user studies because there are usually regulations governing the use of human subjects in experiments related to the safety and security of systems and services. Garfinkel and Lipford (2014) discuss the ecological validity and adversary modelling challenges, which relate to the fact that existing HCI techniques do not apply to security.
- **Study of conflicts by different communities in silos:** As discussed earlier, different communities and interest groups, including the usable security, cybersecurity, software engineering and HCI communities, have been studying usable security independently from each other in silos. There is no medium for collaboration that enables views from different communities and perspectives to be integrated. Moreover, due to the prevalence of independent activities, there is a lack of joint efforts concerning usable security. It remains challenging to integrate findings from different interest groups and communities to be able to develop a strategic vision for usable security (Publication III).
- **Lack of a strategic approach:** Much of the work related to usable security suffers from a surface-level approach, meaning that the solutions are limited to specific problems and do not contribute to the management of the conflicts in general (Garfinkel and Lipford, 2014). For example, there was a perception that the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) poses readability problems for users, and therefore, new CAPTCHAs were developed that allow the user to select relevant images in response to the challenge. However, a question that must still be addressed by the community is whether CAPTCHAs are needed at all. The prime purpose of CAPTCHAs is to protect against denial of service attacks, which is the responsibility of the service provider. However, why should the user bear the

burden of dealing with CAPTCHAs, especially when they cause deviations from the users' primary task? Likewise, the majority of the work on usable security remains at the operational and tactical levels and therefore only has surface-level effects on the usable security problem (Publication III).

2.1.2 Relationship between security and usability with respect to quality views

To understand the conflict between security and usability, it is vital to consider the quality views concept as identified by the ISO 25010 standard. ISO 25010 identifies two models for categorising quality characteristics: (1) the product quality model and (2) the quality in use (QinU) model. The product quality model has eight characteristics and focuses on conforming to the stated product requirements (Rivera *et al*., 2016), whereas the QinU model has five characteristics and focuses on meeting users' expectations while using the product. ISO defines quality in use as 'the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use' (ISO 2011).

The ISO 25010 standard also identifies three quality views: the internal quality view, the external quality (EQ) view and the QinU view (Lew *et al*., 2010). The internal quality view is specified by the product quality model and can be evaluated using static attributes (e.g. requirement specifications, architecture, pieces of code). The EQ view is specified by the product quality model and can be measured and evaluated by dynamic attributes (e.g. running the code in a simulated environment). The QinU view is specified by the QinU model and can be measured and evaluated by the degree to which the product meets the user's needs and expectations during actual use in an operating environment. The ISO standard also identifies the relationships (using the terms 'influences' and 'depends on') between these views (Lew *et al*., 2012).

As stated earlier, security has been defined in a consistent way and has the same meanings among different communities and standards. However, this is not the case for usability, making it a very confusing quality characteristic. Despite listing usability as one of the eight characteristics in the product quality model, the ISO 25010 standard defines usability as 'a subset of quality in use consisting of effectiveness, efficiency and satisfaction, for consistency with its established meaning' in the QinU model (ISO, 2011). Therefore, to distinguish between the two perceptions about usability, usability in use is referred to as actual usability (Lew *et al*., 2010).

In the context of usable security and the existence of a relationship between internal/EQ and QinU views, there exists an 'influences/depends on' relationship between security and actual usability (see Figure 2).

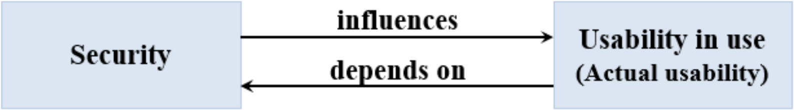


Figure 2. Relationship between security and usability in terms of quality views

Figure 2 clarifies the existence of dependency and the nature of the relationship between security and usability. The way security procedures are implemented as internal qualities/EQs of the system determines and influences the level of usability that the end user experiences. In the case of complex security systems, there is less usability (specifically, less actual usability). Therefore, the conflicts between security and usability are mostly concerning the interdependencies between security and usability in use (actual usability).

2.2 Perceptions concerning the dependency between security and usability: Trade-offs or no trade-offs?

As stated earlier, the study of security and usability dependencies by different communities and interest groups has led to inconsistent perceptions about the dependencies between two (Naqvi and Seffah, 2019). Although these perceptions are known to exist, they have never been explicitly highlighted. Analysis of existing research on usable security identifies two perceptions concerning the dependencies between security and usability. One perception is more traditional, positioning security and usability as conflicting characteristics for which there are trade-offs. However, some research on usable security considers trade-offs and conflicts to be mere myths.

2.2.1 Trade-offs

Most of the work on security and usability dependencies advocates the existence of trade-offs. A case study of iOS and Android was conducted to determine ‘what is more important: usability or security’ (Garg *et al*., 2017). The results show that the importance of security and usability is purely situation-based and that trade-offs sometimes favour security and vice versa. Furthermore, comparison of the two platforms revealed that Android has better usability compared to iOS, but security is more prioritised in iOS devices.

Irrespective of the type of system under consideration, there is evidence of the existence of trade-offs between security and usability (Bai *et al*., 2017; Wang *et al*., 2017). Bai *et al*., (2017) revealed that, when making a choice regarding a preferred system, the participants deliberately made trade-offs between security and usability. Another case study for handling security and usability in database systems determined that systems designed with tight security have limited usability; in other words, robust security comes at the cost of usability (Wang *et al*., 2017).

Researchers have extended the argument for trade-offs to propose that quantification of trade-offs can help effectively achieve balance between security and usability (Nwokedi *et al*., 2016). Kulyk *et al*., (2017) conducted a study to test and quantify possible usability and security trade-offs using three different schemes for e-voting systems. The results reveal that voters were in favour of more secure systems and were willing to sacrifice a maximum of 26 points (on a scale of 0 to 100) for usability in order to achieve a system that provides higher security.

2.2.2 No trade-offs

Other research classifies usability and security trade-offs as mere myths and argues that security and usability are not inherently in conflict. It proposes that researchers have to go beyond adopting human-centred design principles and consider involving the user in the decision-making process (Cranor and Buchler, 2014).

A special issue, ‘The Security–Usability Trade-off Myth’, presents a discussion of usable security researchers and practitioners on this topic (Sasse *et al*., 2016). The participants argued that decreasing usability can lead to less security. As an example of a false trade-off, they discussed two-factor authentication involving a one-time password and the consequences if its length is increased from 6 to 8 characters. There are also cases in which increased usability can lead to increased security. For example, making security functionalities more understandable can lead to improved user decision-making and increased security. Overall, the participants were of the view that ‘security experts simply invoke the myth of trade-off between usability and security and use this as cover to avoid the exercise of saying precisely what security benefit in precisely what scenarios this usability burden is going to deliver’ (Sasse *et al*., 2016, p.36) .

Effective usable security must incorporate aspects of user-value-centred design (Lazaro *et al*., 2017). For this purpose, a framework to identify users’ values associated with security systems and services is required. Specifically, there is a need to shift the approach towards ensuring that users are able to use security. Incorporating value-sensitive design can help in this regard. It requires the following actions: (1) identifying and documenting drivers, trends, and patterns of user behaviour, which might conflict with security mechanisms, and (2) conducting value-sensitive conceptual and empirical analyses for security applications. Lazaro *et al*., (2017) state that ‘identifying the root causes of disengagement can only be done by studying users’ rationales for not using a security mechanism, not by studying how they, or others, fail to use it when they already want to’.

Based on the above discussion of the two perceptions of the dependencies between security and usability, the existence or non-existence of trade-offs is purely situation-based. Thus, there is a need to study the dependencies between security and usability at the level of sub-characteristics. For example, the password-masking example discussed earlier features a trade-off between the security mechanism of authentication and the usability element of feedback. In contrast, enhancing the usability element of understandability positively affects security characteristics such as confidentiality in the context of secure messaging or encrypted emails. Moreover, where there are trade-offs, there is a need to find an effective balance between security and associated usability elements.

2.3 Related Work

Some related work to cater the challenges posed by usable security is presented here, and other relevant literature is presented in Publications I–V. The work presented here can be classified into two categories: (1) literature on the existence of relationships between quality characteristics, including security and usability, and (2) literature presenting frameworks and approaches for handling security and usability conflicts.

2.3.1 Existing literature on the existence of dependencies between quality characteristics

Feitosa *et al.*, (2015) investigated the trade-offs between sub-characteristics concerning the safety of a critical embedded system. Their empirical investigation shows that trade-offs usually favour critical quality characteristics. However, their work is limited to identification of conflicts.

Zhu *et al.*, (2012) proposed a model of fuzzy soft goal interdependency graphs. This model uses qualitative and quantitative approaches to describe, analyse and evaluate the alternatives to certain quality characteristics (sometimes referred to as non-functional requirements [NFRs]) and the relationships among them. It facilitates trade-off decisions among the competing NFR alternatives, and it can help when studying, or at least documenting, conflicts.

Other researchers have investigated the use of design patterns for prioritisation and conflict resolution between quality characteristics. For example, Mehta *et al.*, (2013) introduced a pattern-based approach to analyse the dependencies among selection alternatives that may affect the quality characteristics. They classify possible dependencies into various types, such as partial vs. total and mandatory vs. optional. They argue that their approach could help to make better selections. Supakkul *et al.*, (2010) presented four kinds of NFR patterns for capturing and reusing knowledge about NFRs. These patterns enable visualisation of NFRs and manage synergy and conflict among them. Aldaajeh *et al.*, (2012) reported that the relationship between quality characteristics is a critical aspect for formulating suitable trade-offs and achieving quality. However, the authors extend their argument to claim that, ‘unfortunately, quality attributes relationships’ nature is poorly explored’ (Aldaajeh *et al.* 2012, p.102).

During their research on the establishment of guidelines for the selection of appropriate software architecture, Haoues *et al.*, (2017) found that relationships and dependencies exist between quality characteristics. The authors categorised relationships into four categories:

- positive, or ‘+’ (e.g. security and reliability);
- negative, or ‘–’ (e.g. security and performance efficiency);
- positive-negative, represented by ‘±’ (e.g. usability and performance efficiency), ‘+’ in the case of appropriateness recognizability (a usability sub-characteristic) and time behaviour (a performance efficiency sub-characteristic) or ‘–’ in the case of user error protection (a usability sub-characteristic) and resource utilisation (a performance efficiency sub-characteristic); and
- independent, or ‘0’ (e.g. performance efficiency and functional suitability).

Neri *et al.*, (2018) identified that there is empirical evidence supporting a multidimensional linkage between software quality characteristics and that adopting a one-dimensional perspective limits the use of these characteristics in continuous software engineering environments. More details on the existence of interdependencies between quality characteristics are presented in Publication II.

2.3.2 Existing literature on approaches to handle security and usability conflicts

Al-Darwish and Choe (2019) presented a framework for integrating security with human factors. The framework provides means for classifying and holistically viewing challenges with respect to human aspects in security systems. It also provides a mechanism to evaluate the behaviour of personnel and the adequateness of existing security measures. The framework does not contribute to the development of simultaneously secure and usable systems; rather, it is limited to evaluating the appropriateness of security measures with respect to direct and indirect human factors.

Mujinga *et al.*, (2019) proposed the socio-technical information security framework, which was designed with consideration to both the technical and social aspects of information security. The authors claim that the development of security applications can be improved by applying the 12 design principles included in the framework. However, it is worth noting that the framework is more focused on providing a list of usable security design principles than on contributing to the improvement of industrial development processes.

Parveen *et al.*, (2014) presented a process-oriented approach for aligning security and usability during the SDLC. In this approach, all requirements are assessed, and security requirements are extracted from these. For each security requirement, possible vulnerabilities are identified. The next phase involves identification of usability requirements based on a specific set of characteristics. Finally, security and usability analysis tests are performed to determine which outcomes of the requirement engineering process are highly secure and highly usable. An aspect that remains unaddressed is the practicality of adopting such a methodology in real industrial contexts.

Hausawi and Allen (2014) presented an assessment framework for usable security, which works by filtering and merging security and usability requirements and then applying utility functions for risk analysis. Decision trees are generated to calculate the weight and utility of each characteristic of security and usability. The weights determine the relative importance of characteristics for the requirement specification of software. The authors claim that requirements specified after employing the framework strike a balance between usability, security, and usable security.

Based on a literature review, Gorski *et al.*, (2019) presented principles, guidelines, and patterns for usable security. The authors also identify that many usable security problems are not being addressed by the list of usable security patterns presented in the paper, and such patterns need to be identified. Their work is mainly based on that of Garfinkel (2005), who identified patterns for usable security. However, neither of these studies present a mechanism for identification and standardised documentation of the design patterns. The current dissertation fills this gap. The mechanisms for identification of patterns are presented in Publications I–V, and the template for documentation of patterns is presented in Publication V.

2.4 Is the field catching up?

Dhillon *et al.*, (2016) identify that although usable security has been recognised as one of the top challenges for implementing effective security, little has been accomplished for two reasons: (1) security and usability are afterthoughts during the development lifecycle of systems and services; and (2) security and usability are not integrated into strategic plans for system development. Overall, the state of the art concerning usable security may be improving.

2.4.1 Towards more usable security guidelines

The US National Institute of Standards and Technology (NIST) Special Publication 800-63B states that ‘evaluating the usability of authentication is critical, as poor usability often results in coping mechanisms and unintended work-around that can ultimately degrade the effectiveness of security controls’ (Grassi *et al*. 2017, p, 50). In the revised policy document for memorised secrets, NIST recommends more simplified guidelines for passwords and periodic changes: ‘Do not require that memorized secrets be changed arbitrarily (e.g., periodically) unless there is a user request or evidence of authenticator compromise’ (Grassi *et al*. 2017, p. 54). This is an improvement from the perspective of usable security, as in most organisations users must periodically change their passwords.

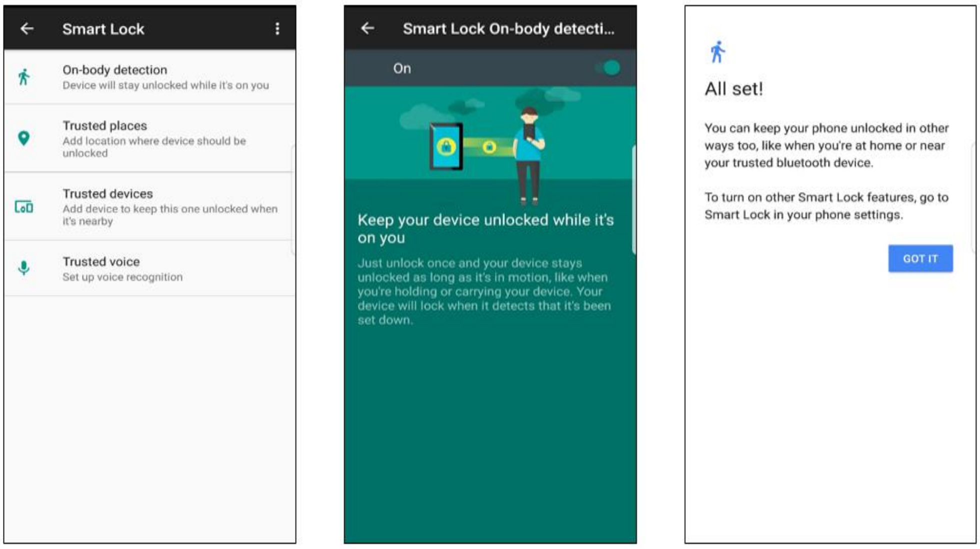
2.4.2 Understanding the context

When developing security solutions, it is vital to understand the context in which these solutions will be deployed (Sasse *et al*., 2016). For example, password-protected user accounts can be subjected to three types of attacks: (1) password stealing, (2) online attacks, and (3) offline attacks. In password-stealing attacks, the attacker tries to obtain a password by using malware, such as keystroke loggers, or by using social engineering techniques. In an online attack, the attacker tries to log in as the user, but this is usually defended against by account lockouts. In offline attacks, the attacker tries to access the password file of the system. There are three ways in which passwords can be stored in the file: (1) without encryption, (2) with reversible encryption or (3) with one-way hashes. In the case of password-stealing attacks that involve vectors, such as social engineering or keystroke loggers, the strength of the password does not matter. Likewise, for online attacks, account lockouts serve as a defence rather than strong passwords. Moreover, if the password file is compromised and not encrypted, the strength of the password is irrelevant. When the password file is protected with reversible encryption, resistance to the attack relies on the encryption key, not the length of the password. The strength and length of passwords are relevant only when passwords are stored as one-way hashes. This raises a question: why must users deal with complex passwords?

Government Communications Headquarters, a British agency, developed more system-centred authentication guidelines that avoid placing burden on the user, for example, such as monitoring logins to detect unusual use, not imposing password changes at regular intervals, employing automated controls to defend against guessing attacks, using account lockouts and using hashing and salting to store passwords. However, the practical implementation of these policies and guidelines remains a challenge.

2.4.3 Smart lock by Android

One positive development with regard to usable authentication is the smart lock feature offered by Android, which allows the user to access the device without authentication if one of the following options is met: (1) the device has been detected on the body, (2) device is in a trusted place, (3) the device is near trusted devices, or (4) the user’s voice is recognised. This helps users a great deal while they are at home and at trusted locations, as they do not have to provide authentication each time, they wish to use the device.



On-body detection
Device will stay unlocked while it's on you

Trusted places
Add location where device should be unlocked

Trusted devices
Add device to keep this one unlocked when it's nearby

Trusted voice
Set up voice recognition

Keep your device unlocked while it's on you

Just unlock once and your device stays unlocked as long as it's in motion, like when you're holding or carrying your device. Your device will lock when it detects that it's been set down.

All set!

You can keep your phone unlocked in other ways too, like when you're at home or near your trusted bluetooth device.

To turn on other Smart Lock features, go to Smart Lock in your phone settings.

Figure 3. Smart lock by Android

Moreover, there are usability improvements in other areas, such as active security warnings, that have led to the development of guidelines for designers. For example, the guidelines for designing security warnings include following a consistent layout; being concise, accurate and encouraging; and offering meaningful options (Garfinkel and Lipford, 2014).

3 Research Methodology

This research applied the DSR method, which involves the design and investigation of artefacts in a particular context (Wieringa, 2014). In this context, artefacts are the design patterns and the processes (frameworks and methodologies) for the creation and identification of the design patterns. However, the following research methods were considered at the beginning of this dissertation.

- **Action Research:** Action research is a type of qualitative research that involves improvement of practice and generation of theory with the researcher acting on or in the social system (Mohajan, 2018). It can be viewed as a cyclic process with phases such as identification of the problem, action planning, action taking, evaluating the action, and specifying learning. This research method could have been suitable if this research was initiated by a company and focused on the problems faced by them. This means active collaboration from the beginning and would have allowed the researcher to access the company premises for execution of the phases of this method. However, there was no active collaboration in the very beginning of this dissertation and the focus of this research was on creation of artefacts that were validated by involving practitioners from companies, therefore, action research was not applied in this dissertation.

- **Grounded Theory:** This methodology was developed by two sociologist Anselm Strauss and Barney Glaser in late 1960s. The methodology is based on development of a theory grounded in data which is systematically gathered and analysed (Glaser and Strauss, 1967). New theory is created through collection and analysis of the data. This method is not appropriate for this research since the main objective was creation of new artefacts for management of the conflicts, not new theories.

- **Design Science Research:** Design science research is a method focused on the development of artefacts with the intention to solve existing problems. DSR has a dual mandate, (1) it attempts to generate new knowledge, insights, and theoretical explanations, and (2) it allows utilization of existing knowledge to solve problems and improve existing solutions (Baskerville *et al*., 2015). The following arguments present rationale for selecting the DSR method for this dissertation.

    - Design science attempts to create artefacts that serve human purposes (Peffers *et al*., 2007). Since the main issue addressed in this dissertation is handling security and usability conflicts and assisting humans (software developers and software architects and designers), it was a natural choice.

    - According to Hevner *et al*., ‘design science…creates and evaluates IT artefacts intended to solve identified organisational problems’ (Hevner *et al*. 2004, p.77). From the perspective of this dissertation, the DSR method guided the creation of IT artefacts to solve the security and usability conflicts (problem) identified in line with one of the research questions. The artefacts were also evaluated and communicated to the relevant audience.

- It is helpful to use the DSR method in domains that lack an existing body of knowledge. It also supports an iterative model of development, which involves creating new and evolved artefacts after the communication phase of the last completed iteration (Peffers *et al.*, 2007). An essential aspect to consider during new iterations is the feedback recorded during the communication phase, as this must be reflected in the evolved processes and artefacts.

3.1 Design Science Research Cycles

The DSR process contains three cycles: the relevance cycle, the design cycle, and the rigor cycle (Hevner, 2007). Each cycle can be described as follows:

1. **The relevance cycle:** The motivation underlying this cycle is to improve the environment (software ecosystem) through the introduction of new artefacts. During this research, artefacts were formulated to help software developers and designers manage conflicts. The problem considered during the relevance cycle is the conflict between security and usability, and the evaluation criterion are meant to expose these artefacts to enable researchers and practitioners in the domain to comment on and review them. It is vital to demonstrate how the artefact would address a real-world usable security problem. This cycle involves as much iterations as required.

2. **The design cycle:** This cycle is iterative and involves a build-and-evaluate loop for the design of artefacts as both a product and a process. Iteration is performed until the item is validated and new knowledge can be added to the knowledge base. The artefacts added to the knowledge base after this cycle include different frameworks, methodologies, and patterns.

3. **The rigor cycle:** This cycle includes the selection, application, and evaluation of knowledge bases to build and evaluate artefacts. Knowledge bases include theories, experiences, experts and existing artefacts and processes. In the context of this dissertation, the knowledge base includes personal experiences, existing case studies, existing frameworks, surveys, and interviews with experts.

Each of the DSR cycles is presented in Figure 4.

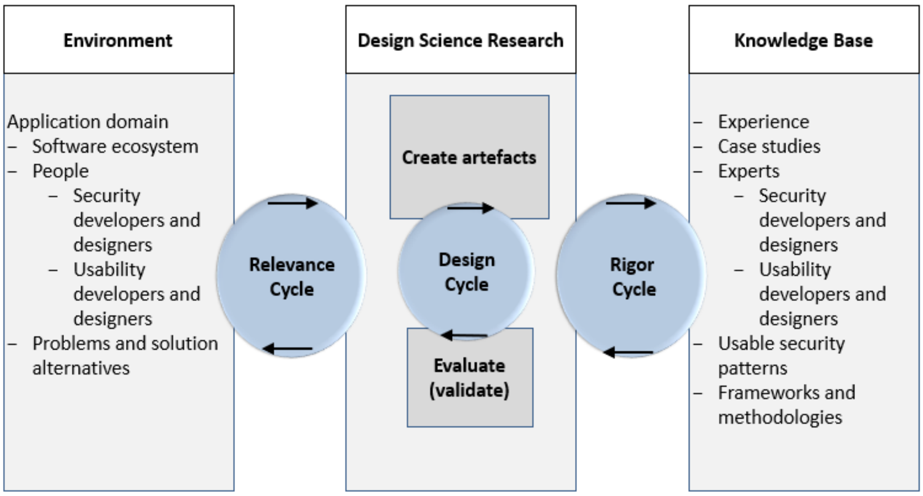

Environment
Application domain
– Software ecosystem
– People
– Security developers and designers
– Usability developers and designers
– Problems and solution alternatives
Relevance Cycle
Design Science Research
Create artefacts
Design Cycle
Evaluate (validate)
Rigor Cycle
Knowledge Base
– Experience
– Case studies
– Experts
– Security developers and designers
– Usability developers and designers
– Usable security patterns
– Frameworks and methodologies

Figure 4: DSR cycles (adopted from Hevner, 2007; [re-drawn in the context of this research])

3.2 Design Science Research Procedure

In addition to the DSR cycles, Peffers *et al*., (2007) presented a DSRM that incorporates principles and practices from existing DSR literature while adding procedures to conduct DSR for various real-world research problems. The methodology includes six steps: (1) problem identification and motivation, (2) definition of the objectives for a solution, (3) design and development, (4) demonstration, (5) evaluation, and (6) communication.

Moreover, the DSRM provides four possible entry points for initiating the process, including: (1) problem-centred initiation, (2) objective-centred solutions, (3) design- and development-centred initiation, and (4) client/context-centred initiation. It is worthwhile to state that during this dissertation, problem-centred initiation was used to initiate the research process. The steps of the DSRM are presented in Figure 5.

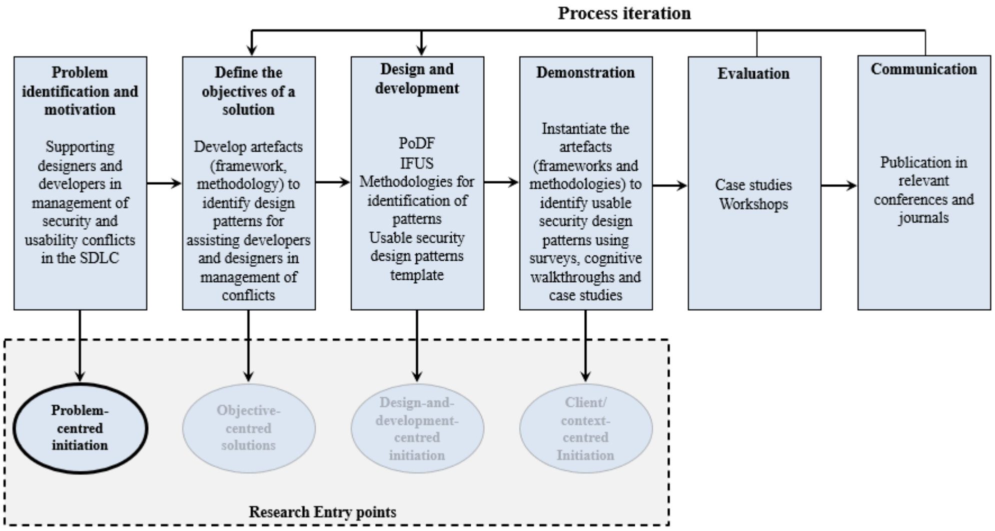

Process iteration
Problem identification and motivation
Supporting designers and developers in management of security and usability conflicts in the SDLC
Define the objectives of a solution
Develop artefacts (framework, methodology) to identify design patterns for assisting developers and designers in management of conflicts
Design and development
PoDF
IFUS
Methodologies for identification of patterns
Usable security design patterns template
Demonstration
Instantiate the artefacts (frameworks and methodologies) to identify usable security design patterns using surveys, cognitive walkthroughs and case studies
Evaluation
Case studies
Workshops
Communication
Publication in relevant conferences and journals
Problem-centred initiation
Objective-centred solutions
Design-and-development-centred initiation
Client/ context-centred Initiation
Research Entry points

Figure 5. Procedure for conducting DSR (adopted and re-drawn from Peffers *et al*., 2007)

Each step of the DSRM in the context of this dissertation can be described as follows:

1. **Problem identification and motivation:** This step involves identification of the research problem and justification of the value of a solution. The research problem investigated during this research is the conflict between security and usability. Conflicts and trade-offs between security and usability can have consequences ranging from monetary losses to human safety, and therefore, it is worthwhile to design a solution that helps software developers and designers manage those conflicts, thereby delivering solutions that are simultaneously usable and secure. A solution for managing conflicts needs to govern aspects from identification of conflicts to their elicitation as suitable trade-offs and documentation as design patterns. Design patterns can prove helpful in positively influencing the decision-making abilities of other designers and developers in similar contexts.
2. **Define the objectives of a solution:** This step involves explicitly identifying the objective of the solution, which in this case was to formulate artefacts (frameworks and methodologies) that can be used to identify usable security design patterns. Different frameworks and methodologies were formulated to cover all possible contexts for identification of design patterns. All the artefacts contribute to the solution identified earlier: to assist developers and designers in handling security and usability conflicts.

3. **Design and development:** This step involves the creation of artefacts, which could take the form of processes, constructs, models, methods or instantiations (Hevner *et al.*, 2004). Different artefacts created during this research include the following:
    a. ***Processes*****:** During this dissertation, various processes (frameworks, methodologies) were formulated to cover all possible contexts for the identification of design patterns. Each of the processes was driven by a context, and the reason for their creation was to develop and identify artefacts (usable security design patterns). Some of the processes included the following:

        i. **IFUS:** The IFUS governs the management of security and usability conflicts within the scope of the SDLC and allows security and usability concerns to be incorporated collectively from the requirement engineering stage of the SDLC.

        ii. **PoDF:** The purpose of the PoDF is to assist in examining conflicts between quality characteristics. It does so by providing various means for identification of conflicts, modelling conflicts at the level of sub-characteristics, eliciting suitable trade-offs between conflicting characteristics and documenting suitable trade-offs as patterns for use by developers in the industry.

        iii. **Three-stage methodology for the identification of usable security patterns:** As stated earlier, this three-stage methodology is based on the identification of new patterns from existing implementations that set good practices in the industry. It provides uniform means to identify new patterns and an opportunity for various stakeholders to contribute to the identification of patterns and building of a catalogue of usable security patterns.

    b. ***Patterns*****:** The other artefacts in this case are usable security design patterns. Security developers are trained to handle security concerns, and usability is a minor concern for them. Similarly, usability developers are trained to ensure elements of usability and positive UX, and they consider security procedures to be a hindrance. Therefore, usable security design patterns could provide an opportunity for security developers to assess the usability of their security options and vice versa. Patterns have shown their effectiveness to document the best practices for addressing a common design problem.

        Each pattern expresses a relation between three things: context, problem, and solution. The patterns have three dimensions: descriptive, normative, and communicative (Mor *et al.*, 2010). From the perspective of usable security, the communicative dimensions of the patterns enable different communities to discuss design issues and solutions. Patterns also prove effective in domains that lack an existing body of knowledge; in such cases, patterns assist in identifying effective practices as they emerge and capture them as objects for discussion, scrutiny and modification (Mor *et al.*, 2010).

4. **Demonstration:** This step involves instantiation of the solution. During this dissertation, frameworks and methodologies identified during the previous stage were instantiated to identify different usable security design patterns. The method used for demonstration varied for each artefact.
5. **Evaluation:** This step involves evaluation of whether the artefact meets the objectives identified during the second step. Evaluations can be both qualitative and quantitative. For processes created during this dissertation, the limitations identified during the evaluation step led to the creation of new processes that are applicable in other contexts. Moreover, the patterns are under continuous evaluation (monitoring and review) by the developers and designers who use them. Requests for modifications identified with time are subject to consideration and can lead to modification of existing patterns or creation of new ones.
6. **Communication:** This step involves communicating the problem and solution among researchers and practitioners to gain their feedback on artefacts regarding aspects such as utility, novelty, and the rigor of design. In this dissertation, publications in conferences and journals served as means for communicating the problem and solution to other researchers and practitioners to record their feedback.

3.3 Data Collection and Evaluation Methods

To meet the objectives of this dissertation, different data collection and evaluation methods were used.

3.3.1 Surveys

A survey is a data collection method used to gain information and insight into topics of interest from a group of respondents. During this dissertation, surveys were conducted to (1) understand respondents' feedback on the usability of security, (2) identify potential conflicts, and (3) identify the sub-characteristics of usability that are affected by the security of state-of-the-art devices. Participation in the surveys was voluntary and due ethical concerns were considered while handling the data recorded from the participants. More details about each survey applied in this dissertation are presented below:

- **Survey I (Usability of Security in Mobile Devices):** This survey was conducted online to record respondent's feedback on usability of security of their mobile devices. One objective in this regard was to identify potential security and usability conflicts. The participants in the survey were users of mobile devices (smartphones, tablets) mainly based in Lappeenranta, Finland. Participation in the survey was specified by an inclusion criterion that allowed participants using the mobile devices for their personal and work-related purposes to participate in the study irrespective of their age, gender, and educational background. The participants were recruited mainly through social media. To ensure that the respondents were from the same city, the responses were filtered based on the IP address from which the survey link was accessed. Demographic details about the respondents were collected to ensure that the responses represent a diverse cross-section of the target population. For instance, out of the total 75 respondents, the majority of the respondents (62.7%) were from the age group 22-34 years, however, the other respondents belonging to different age groups (8% from 35-44 years of age, 16% from 45-54 years of age, 10.7% from 55-64 years of age, and 2.7% more than 65years of age) constituted the remaining 37.3% of the respondents. Among the respondents, 32% had a computer science background, and the remaining 68% belonged to other fields of study, including engineering, business, medicine, etc. Moreover, to ensure that the participants are not forced to respond to the questions they are not sure about, a 'Neutral' option was provided as the midpoint of the scale. The purpose of this survey was to demonstrate and instantiate PoDF. The survey questionnaire is presented in Appendix A and more details about the survey are presented in Publication II.
- **Survey II (Usability of Security for Smartphone Users):** As a step further from Survey I, an exploratory survey was conducted to identify the sub-characteristics of usability considered while implementing the security features in state-of-the-art smartphones. Participation in the survey was specified by an inclusion criterion that allowed smartphone users to participate in the study irrespective of their age, gender, educational background. The participants included students and staff at LUT University. The participants were recruited by posting the survey link on the LUT intra portal. The survey was conducted over a period of two weeks in the English and Finnish languages simultaneously. Two hundred and two (202) respondents completed the survey. Demographic details about the participants were collected to ensure that the responses represent a diverse cross-section of the targeted population. Moreover, the survey questionnaire also included questions concerning the usage of smartphones in addition to the questions concerning user's feedback on the implementation of various elements of usability in the security features of today's smartphones. To assist the

respondents in understanding various terms and definitions used in the survey, a description of each term as identified and defined by the ISO 25010 standard was included. The English version of the survey questionnaire is presented in Appendix B.

- **Survey III (Information Security Awareness of Employees):** This survey was conducted as part of a project funded by Business Finland, and the respondents included personnel of Finnish IT companies (Visma and other companies that are part of the Finnish Information Security Cluster). The inclusion criterion in the survey was to include respondents with at least basic knowledge of their company's information security policy and using a computer or portable device as part of their daily work. The survey was conducted online, and a link to the survey was disseminated via email to the focal person who disseminated it using internal email lists among the personnel. However, only 15 respondents from different Finnish companies completed the survey. The survey focused on human aspects that are relevant to information security, and the purpose was to identify security and usability conflicts in the day-to-day organizational practices. The survey questionnaire is presented in Appendix C.

It is relevant to mention that ethical concerns were duly considered during the surveys. Personal information (if any) recorded during the surveys, was not and will not be disclosed at any time. The respondents were also provided a copy of the survey results upon request.

3.3.2 Interviews

An interview is a data collection method used to identify respondents' opinions, thoughts, and experiences on a topic of interest. A series of interviews was conducted with personnel from the Finnish IT industry to understand their perspective on security and usability conflicts, identification of conflicts based on their experiences and, most importantly, to understand the state of the art concerning security and usability conflicts. The interviews were audio-recorded, and due ethical concerns were followed. The interviewees' consent was recorded before the interview.

Personnel in different roles were interviewed to understand different perspectives. The interviewees included: (1) IT managers, (2) lead architects, (3) security engineers, and (4) usability/UX developers. The interviews enabled validating the perceptions developed after analysis of existing literature. The interviews also served as a means for establishing a rationale for the development of artefacts and potential use of design patterns for assisting the developers in handling security and usability conflicts.

3.3.3 Workshops

Two workshops were conducted online with objectives as discussed below.

- **Workshop I (*Objective*: To evaluate and validate the artefacts created using the DSRM):** An online workshop was conducted to evaluate and validate the artefacts formulated for addressing the usable security concerns raised after the literature review and interviews. The participation in the workshop was specified by an inclusion criterion which allowed to include only the personnel (working for the industrial partner) who had participated in the interviews. 10 personnel fulfilling the inclusion criteria joined the workshop. The concerns raised after the interviews played a key role in the formulation of the artefacts, therefore, it was an obvious choice to present the artefacts for review and comments by the personnel who had participated in the interviews. Requests for modifications were considered and discussed during the workshop and incorporated into the final version of the artefacts. Moreover, to instantiate the methodology presented in Publication IV the participants were provided a usable security problem, and a usable security design pattern was identified.
- **Workshop II: (*Objective*: To identify the challenges and opportunities relevant to usable security):** An online workshop was conducted to identify the challenges and opportunities relevant to usable security from the Finnish IT industry perspective. The call for participation in the workshop was open for security and usability practitioners working for Finnish Information Security Cluster member companies. 13 participants from 10 different companies joined the workshop. The main goal was to discuss and brainstorm the challenges posed by usable security. However, it was also intended to identify state-of-the-art ways in which these challenges are tackled in the industry. Different usable security challenges and improvement opportunities were identified and listed during the workshop, which will be considered as future work.

4 Overview of Publications

In line with the research questions, five publications are included in this dissertation. The objective of each publication along with the research question addressed and relevance to the dissertation is also discussed.

4.1 Publication I – Incorporating the Human facet of Security in Developing Systems and Services

4.1.1 Objective

The objective of this publication is to formulate a mechanism for managing conflicts during the SDLC. The framework presented in the publication governs the development of systems and services while identifying conflicts and eliciting suitable trade-offs. In line with the approach advocated in this research, the outcomes of employing the framework are documented as usable security patterns. The patterns can assist other developers and designers in managing conflicts.

4.1.2 Relevance to the dissertation

The publication is related to RQ-2 and RQ-3, identified in Chapter 1 of this dissertation. In line with RQ-2, the publication illustrates how to use design patterns for documenting and disseminating suitable trade-offs. Furthermore, addressing RQ-3, the publication presents IFUS, which can be seen as a mechanism to govern the management of conflicts between security and usability during the SDLC.

4.1.3 Output and contribution

Figure 6 presents the IFUS's three layers, different elements and activities and the participants. A bottom-up approach was applied to construct the elements of the framework. The participants in activities related to the IFUS are system designers and developers from security and usability domains. The framework is adopted during the requirement engineering phase of the SDLC. The outcomes of employing it are documented as usable security design patterns and disseminated among the community of developers for use in similar contexts. The publication also presents a validated template for documentation of usable security patterns.

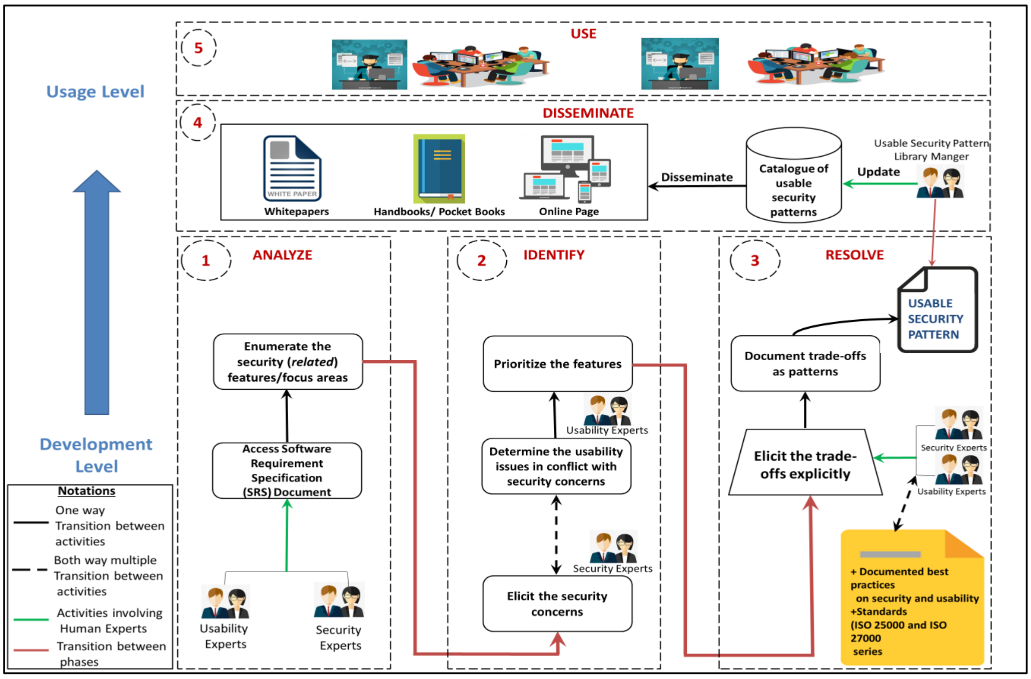

Usage Level
Development Level
5
USE
4
DISSEMINATE
WHITE PAPER
Whitepapers
Handbooks/ Pocket Books
Online Page
Disseminate
Catalogue of usable security patterns
Update
Usable Security Pattern Library Manger
1
ANALYZE
2
IDENTIFY
3
RESOLVE
USABLE SECURITY PATTERN
Enumerate the security (related) features/focus areas
Prioritize the features
Document trade-offs as patterns
Usability Experts
Access Software Requirement Specification (SRS) Document
Determine the usability issues in conflict with security concerns
Elicit the trade-offs explicitly
Security Experts
Usability Experts
Security Experts
+ Documented best practices on security and usability +Standards (ISO 25000 and ISO 27000 series)
Elicit the security concerns
Usability Experts
Security Experts
Notations
One way Transition between activities
Both way multiple Transition between activities
Activities involving Human Experts
Transition between phases

Figure 6. IFUS (Naqvi *et al.*, 2020; Publication I)

In the analyse phase of the IFUS, requirements are collected and analysed. In the identify phase, goals related to both security and usability are set, leading to the identification of potential conflicts. In the resolve phase, the identified conflicts are resolved and documented as patterns, which are disseminated in the disseminate phase. In the use phase, developers use the patterns in similar contexts.

4.2 Publication II – Framework for Examination of Software Quality Characteristics in Conflict: A Security and Usability Exemplar

4.2.1 Objective

The objectives of this publication are twofold: (a) to argue for the importance of handling conflicts between quality characteristics in general, and (b) to formulate a framework for examining conflicts between software quality characteristics, using the specific case of security and usability.

4.2.2 Relevance to the dissertation

The publication is related to RQ-2 and RQ-3. In line with RQ-2, the publication illustrates how to use design patterns for documenting and disseminating suitable trade-offs. Furthermore, in line with RQ-3, the publication presents the PoDF, which can be seen as a mechanism to govern the management of conflicts between security and usability. What distinguishes PoDF from the IFUS presented in Publication I is that the latter uses system requirement specifications for identification of conflicts, which are mostly applicable to the development of newer systems or system updates, while the former is based on using different methods (as shown in Figure 7) for identification of conflicts in extant systems.

4.2.3 Output and contribution

The main output of the publication is the PoDF, which has four layers and governs the management of conflicts from identification of conflicts to their dissemination as suitable trade-offs. The context in which the PoDF is applied are systems that have already been deployed and for which identified usable security problems must be fixed in newer versions. The PoDF is presented in Figure 7.

The PoDF also utilises a methodology presented in (Naqvi *et al*., 2018) to assign severity ratings to conflicts. Conflicts with higher ratings must be fixed in upcoming releases of the system. The lessons learnt from the identification and documentation of conflicts between security and usability could be useful for adapting the PoDF and identifying patterns that address the trade-offs between characteristics such as performance efficiency and maintainability.

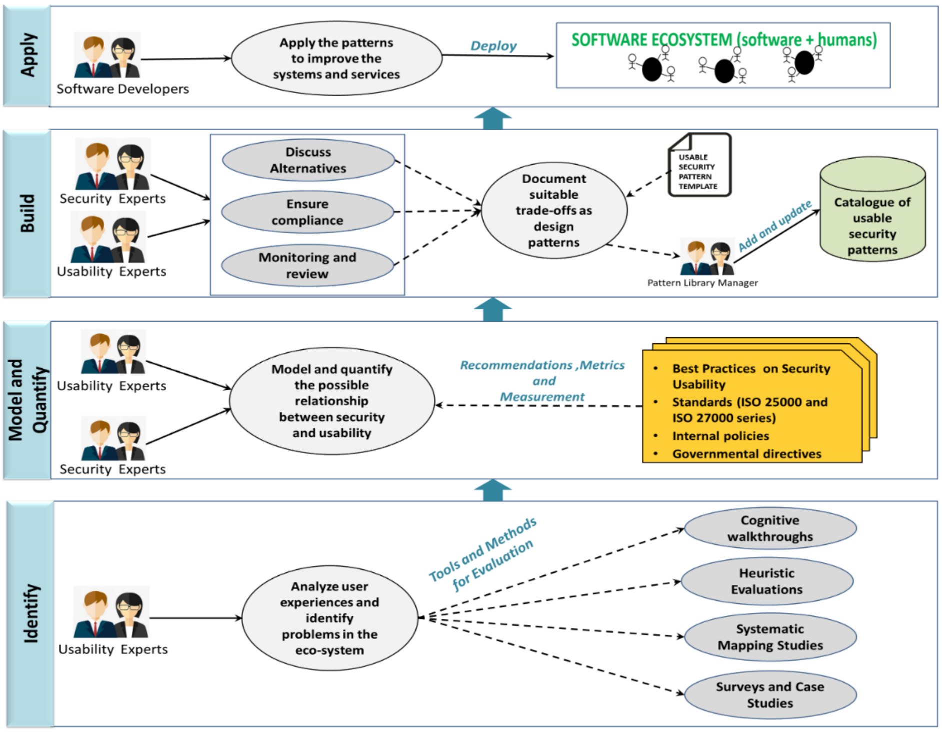

Apply
Software Developers
Apply the patterns to improve the systems and services
Deploy
SOFTWARE ECOSYSTEM (software + humans)
Build
Security Experts
Usability Experts
Discuss Alternatives
Ensure compliance
Monitoring and review
Document suitable trade-offs as design patterns
USABLE SECURITY PATTERN TEMPLATE
Add and update
Catalogue of usable security patterns
Pattern Library Manager
Model and Quantify
Usability Experts
Security Experts
Model and quantify the possible relationship between security and usability
Recommendations ,Metrics and Measurement
Best Practices on Security Usability
Standards (ISO 25000 and ISO 27000 series)
Internal policies
Governmental directives
Identify
Usability Experts
Analyze user experiences and identify problems in the eco-system
Tools and Methods for Evaluation
Cognitive walkthroughs
Heuristic Evaluations
Systematic Mapping Studies
Surveys and Case Studies

Figure 7. Pattern Oriented Design Framework (PoDF) (Naqvi *et al*., 2020; Publication II)

4.3 Publication III – Towards Identification of Patterns Aligning Security and Usability

4.3.1 Objective

The publication presents a three-stage methodology for the identification of usable security patterns from existing implementations that set good examples in the industry. This methodology follows a bottom-up approach in which patterns from existing implementations are identified. In contrast, in Publication I and II, new patterns are identified first and implemented later.

4.3.2 Relevance to the dissertation

The methodology presented in the publication was formulated during Stage III, when the additional question 'what are means for the development of a catalogue of usable security design patterns?' was identified. The publication aligns with RQ-1 and RQ-2 of this dissertation. In line with RQ-1, the publication identifies some of the gaps in the state of the art, but in line with RQ-2, the publication presents arguments to justify the use of patterns as a way to assist developers in handling security and usability conflicts.

4.3.3 Output and contribution

The contributions of this publication to the dissertation are twofold. First, it provides arguments (justified by existing literature) regarding the use of patterns for handling conflicts between security and usability. Second, based on the need (identified during Stage III) to develop a mechanism to identify a catalogue of usable security patterns, this publication presents a three-stage methodology.

The first stage involves the selection of a common usable security problem. Existing implementations to address the problem are assessed. To ensure that the best implementations are documented as a pattern, it is imperative to fulfil the rule of three, according to which at least three instances of similar implementations are needed before a pattern can be identified and documented.

The second stage involves review of the newly documented pattern by one or more experts in the field. During the third stage, accepted patterns are added to the catalogue, patterns with modify recommendations are referred back to the security and usability experts who identified the need for modification, and rejected patterns are discarded.

Figure 8 presents the details of this methodology. From an industrial perspective, it can enable documentation of new patterns from implementations by experienced developers, thereby facilitating the learning and training of new developers.

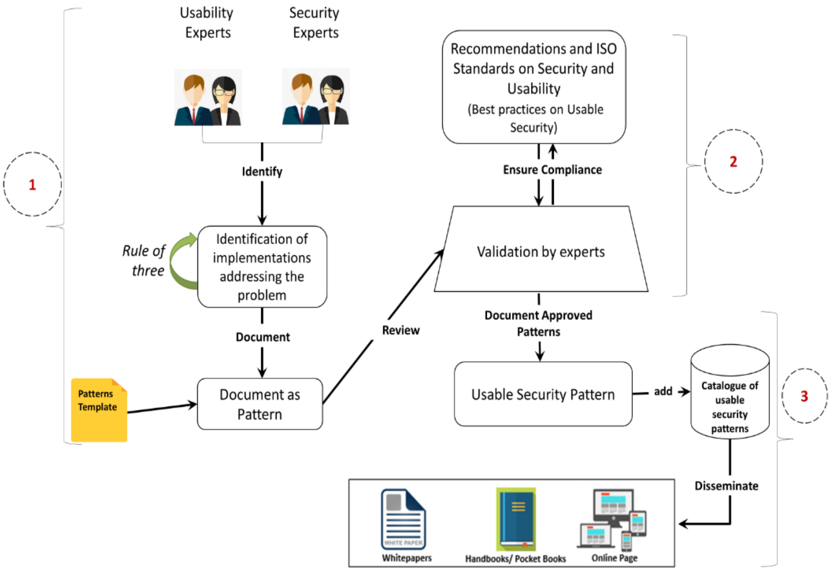


Figure 8. The three-stage methodology for identification of usable security patterns (Naqvi *et al*., 2020; Publication III)

4.4 Publication IV – Usable Security by Design: A Pattern Approach

4.4.1 Objective

This publication presents a proposal for a participatory workshop for identification of usable security design patterns. The workshop would provide a forum for researchers and practitioners to participate in identifying the catalogue of usable security patterns. Moreover, the workshop could provide a forum for discussing a variety of issues concerning security and usability conflicts while documenting conflicts and suitable trade-offs as design patterns for use by other designers and developers.

4.4.2 Relevance to the dissertation

The methodology presented in this publication was formulated during Stage III to address the question of 'what are means for the development of a catalogue of usable security design patterns?' In line with RQ-1, the publication presents some of the challenges in the state of the art, and in line with RQ-2, it rationalises the use of patterns to handle security and usability conflicts due to their descriptive, normative and communicative abilities.

4.4.3 Output and contribution

The publication presents lists of activities to be applied when conducting developer workshops to develop a catalogue of usable security design patterns (Figure 9). First, narratives describing a usable security problem are distributed among the participants. The groups are tasked with designing their own solutions to the problem. Afterwards, the solutions from each group are subjected to comparative analysis in an attempt to identify instances of good design. In accordance with the rule of three, once three instances of similar implementations for a problem are identified, the pattern is documented on a standard template.

The next step involves validation of the patterns. The participants are provided with a list of design patterns (that have already been identified) and a problem scenario. The problem scenario used during this stage involves a set of problems, and the task involves the selection of the patterns (from a list) that are applicable in the context under consideration. If the right pattern is applied in the right context, it is validated; otherwise, it is modified to ensure the use of the right patterns in the right scenarios.

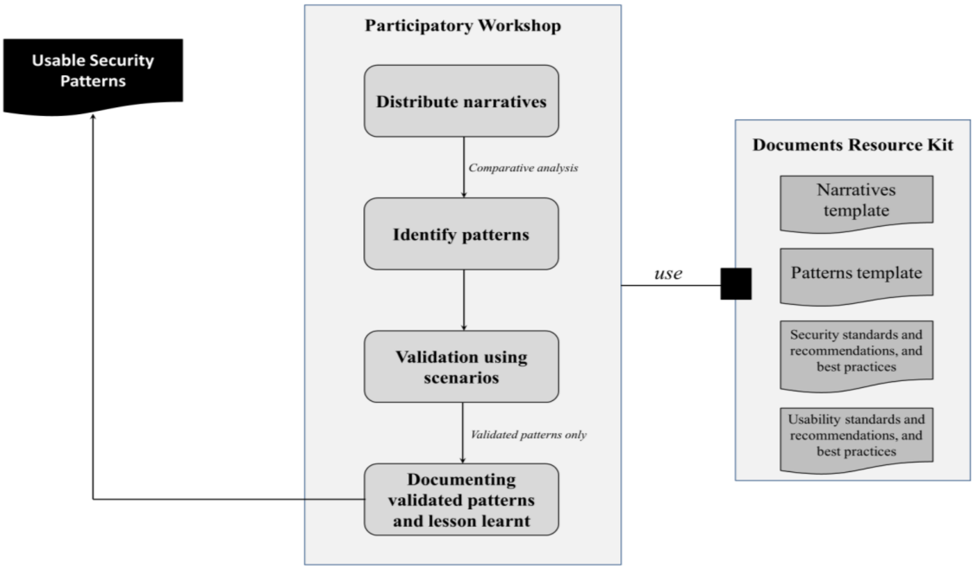

Usable Security Patterns
Participatory Workshop
Distribute narratives
Comparative analysis
Identify patterns
Validation using scenarios
Validated patterns only
Documenting validated patterns and lesson learnt
use
Documents Resource Kit
Narratives template
Patterns template
Security standards and recommendations, and best practices
Usability standards and recommendations, and best practices

Figure 9. Participatory workshop for identification of usable security design patterns (Naqvi and Porras, 2020; Publication IV)

4.5 Publication V – Interdependencies, Conflicts and Trade-Offs Between Security and Usability: Why and How Should We Engineer Them?

4.5.1 Objective

The objectives of this publication are twofold: (1) to identify gaps in the state of the art while discussing why it is important to handle security and usability conflicts and (2) to discuss how to handle security and usability conflicts.

4.5.2 Relevance to the dissertation

This publication aligns with RQ-1 and RQ-2. Addressing RQ-1, the publication presents some gaps in the state of the art concerning usable security, and in line with RQ-2, it lays the foundation for use of the pattern approach for management of the conflicts. As this dissertation applied an iterative research methodology, Publication II represents an evolved and improved version of this publication. The feedback recorded during the communication phase of Publication V was considered when formulating the framework presented in Publication II.

4.5.3 Output and contribution

Figure 10 portrays the four-stage process-oriented framework proposed in this publication. The framework includes a sequence of activities to be followed in order to address the conflict. The first stage involves analysis of the diverse human experiences and tasks of stakeholders and end-users associated with security technologies; modelling of the interaction between stakeholders and users to accomplish those tasks; and quantification of possible usability problems. The second stage involves modelling of the relationship between security and usability using the descriptions of human experiences, tasks and usability problems identified in the previous stage as inputs. The third stage involves the development of solutions and their documentation in the format of patterns. In the last stage, patterns are applied in the software ecosystem.

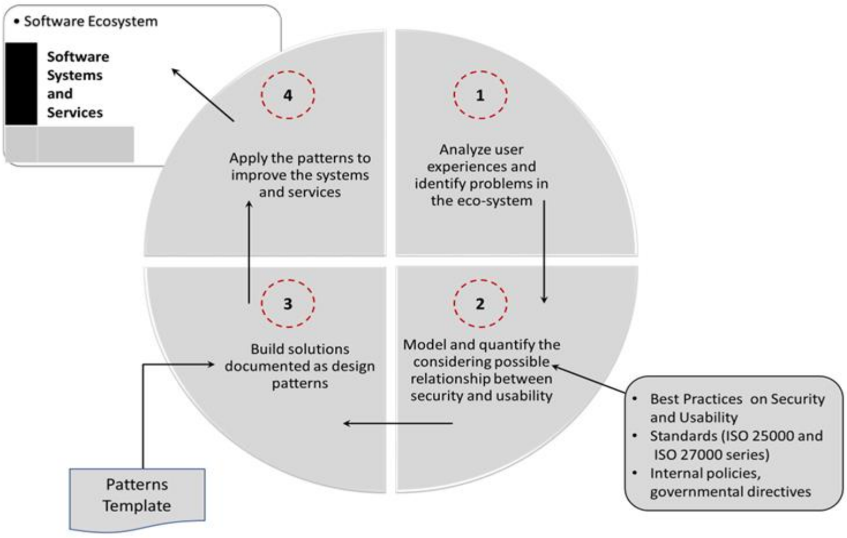

• Software Ecosystem
Software Systems and Services
4
Apply the patterns to improve the systems and services
1
Analyze user experiences and identify problems in the eco-system
3
Build solutions documented as design patterns
2
Model and quantify the considering possible relationship between security and usability
• Best Practices on Security and Usability
• Standards (ISO 25000 and ISO 27000 series)
• Internal policies, governmental directives
Patterns Template

Figure 10. The proposed process-oriented framework for engineering conflicts between security and usability (Naqvi and Seffah, 2019; Publication V)

5 Discussion and Limitations

The work performed during this dissertation led to some findings, which are discussed in this chapter. The limitations of the current work and avenues for further research are also presented.

5.1 Discussion

The findings of this dissertation are discussed in the subsequent sub-sections.

5.1.1 Context-based dependency

Based on the investigation of the interdependency between security and usability during this research, it was revealed that the existence of conflicts and trade-offs is completely dependent on the context. In most instances, security and usability are in conflict, and in some cases, they could positively influence each other. For example, enhancing the usability element of understandability positively affects security characteristics, such as confidentiality, when using mechanisms such as secure messaging or encrypted emails. However, there is a need to identify the contexts where security and usability negatively influence each other and model the interdependencies in a way that causes minimal trade-offs. The frameworks presented in this dissertation could be useful for this purpose.

5.1.2 Handling the interdependencies at the level of sub-characteristics

The identification and modelling of the interdependencies between security and usability could be aided by handling their mutual relationship at the level of sub-characteristics. Considering the context-based dependency between security and usability, it is imperative to handle interdependencies at the level of sub-characteristics; rather than declaring that security and usability are in conflict, there is a need to identify specific contexts in which their sub-characteristics are in conflict or positively influencing each other.

5.1.3 The earlier, the better

Management of conflicts between security and usability could be less problematic when handled earlier in the SDLC. Therefore, for newly developed systems, security and usability concerns should be considered during the requirements and design phase of the SDLC. However, for a system already in the production environment, it is vital to fix all reported usable security issues in upcoming releases of the system. This approach is also advantageous from an economic perspective. Management of usable security issues from the start of the SDLC can help to avoid conflict situations, thereby circumventing the costs and efforts associated with redoing work due to changes in the system design at later stages of the development lifecycle. The same was identified during the interviews with practitioners from the industry.

5.1.4 Efficacy of using patterns

This dissertation advocates for the use of design patterns to support developers working to manage security and usability conflicts. The need for using design patterns for handling security and usability conflicts was realized in the early stages of this research, however, the potential efficacy was established during the interviews and workshops with practitioners from the industry. The artefacts formulated during this dissertation can assist in the development of a catalogue of usable security patterns, which can support developers in handling security and usability conflicts. Patterns provide benefits like a common vocabulary, shared documentation, and improved communication. Their ability to be improved over time and incorporate multiple viewpoints make them suitable for interdisciplinary fields like usable security. In addition, they can effectively assist developers in making reasonably accurate choices while dealing with conflicts.

5.2 Limitations and Open Issues

A lack of quantitative assessments of conflicts and trade-offs is one of the limitations of this work. Quantification of the degree of conflict between security and usability in a particular context could help to prioritise which security and usability requirements need to be fixed. Moreover, in cases where a trade-off is made, quantification of trade-offs could let developers understand the degree to which security is improved by losing a certain degree of usability—and vice versa—and thus make suitable trade-off decisions. The following sub-sections present some open issues that require further investigation.

5.2.1 Need for metrics and measurement of usable security

Most metrics and techniques that measure only usability are not necessarily beneficial for the usability of security systems. Thus, there is a need to develop ways of measuring the adequacy of usable security. An example could be measurement of the degree of conflict between the sub-characteristics of security and usability. Moreover, in usable security research, there has been an emphasis on determining the deviation from the user's primary task, which would require a set of metrics. A measurement methodology (Naqvi *et al.*, 2018) identifies metrics, such as the number of user complaints. However, the efficacy and completeness of such metrics needs to be explored further.

5.2.2 Towards making usable security a standard requirement

Usable security could be made a standard requirement like other quality requirements enforced by ISO standards, such as ISO 25010. Specifically, it is proposed that the security in use characteristic is added to the QinU model. Neither usability nor its sub-characteristics, as external qualities, can be added to security as sub-characteristics. Therefore, a strategic framework is required for adding the security in use characteristic to the QinU model with usability in use as a subset. Adding usability in use as a subset of security in use would involve adding characteristics such as effectiveness, efficiency, satisfaction, usability in use compliance and other sub-characteristics to security.

Other sub-characteristics could include confidentiality in use, or how efficaciously the procedures implemented in the system can preserve information/data from unauthorised disclosure. Similarly, integrity and authentication in use refer to the efficacy of implemented functions for ensuring integrity and performing authentication, respectively. When software developers and designers consider elements like effectiveness in use, efficiency in use and others in the security design, it can lead to development of systems and service that are simultaneously usable and secure.

6 Conclusion

Security and usability are often in conflict, but their mutual interdependencies can be effectively synergised by modelling their relationships and managing conflicts. This research serves as a step towards aligning security and usability during the SDLC. It advocates for the concept of usable security by design, which is based on the use of design patterns in management of the conflicts. Patterns can be used by developers who are experts in either security or usability when making decisions in similar contexts. To develop a catalogue of patterns for this purpose, the current dissertation presents artefacts developed using DSR. To standardise the documentation of patterns, a template was created and validated.

In addition to the creation of artefacts, this dissertation presents gaps in academic research and industrial practices that are relevant to usable security. Identification of these gaps helped define the rationale for this research and were thoroughly considered during development of the artefacts.

One aspect that was not explicitly highlighted previously was the relationship between security and usability with respect to quality views. This aspect, which was found during this research, has the potential to be investigated further. It also could serve as the basis for a strategic framework and proposal for an amendment to the ISO-25010 standard.

During this research, it was determined that handling the conflicts at the level of the sub-characteristics of security and usability could be more effective for management of conflicts in comparison to dealing with the conflicts at higher level. Different artefacts created during this research implement the same by illuminating the sub-characteristics in conflict.

In conclusion, it is vital to manage conflicts between security and usability in the development of systems and services. Use of design patterns can aid management of conflicts, and the artefacts presented in this dissertation could be of significant help for identification of such patterns. Academic researchers and industrial practitioners can adopt the frameworks and methodologies presented here to contribute to the development of the catalogue of usable security patterns.

References

Aldaajeh, S., Asghar, T., Khan, A. A. and Ullah, M. (2012). Communing different views on quality attributes relationships' nature. *European Journal of Scientific Research,* 68(1), 101–109.

Al-Darwiash, A. I. and Choe, P. (2019). A framework of information security integrated with human factors. In: Moallem A ed. *International Conference on Human-Computer Interaction 2019*. *Orlando, Florida, US*A. Lecture Notes in Computer Science. V, vol. 11594. Cham: Springer, pp. 217–229.

Alexander, C., Ishikawa, S., and Silverstein M. (1977). *A Pattern Language*. New York, Oxford University Press.

Bai, W., Kim, D., Namara, M., Qian, Y., Kelley, P. G. and Mazurek, M. L. (2017). B-alancing security and usability in encrypted email. *IEEE Internet Computing*, 21(3), 30–38.

Barlev, S., Basil, Z., Kohanim, S., Peleg, R., Regev, S. and Shulman-Peleg, A. (2016). Secure yet usable: Protecting servers and Linux containers. *IBM Journal of Research and Development*, 60(4), 12:1–12:10.

Baskerville, R. L., Kaul, M., Storey, V. C. Genres of Inquiry in Design-Science Research: Justification and Evaluation of Knowledge Production. *MIS Quarterly*, 39(3), 541–564.

Caputo, D. D., Pfleeger, S. L., Sasse, M. A., Ammann, P., Offutt, J. and Deng, L. (2016). Barriers to usable security? Three organizational case studies. *IEEE Security and Privacy*, 14(5), 22–32.

Cranor, L. F. and Buchler, N. (2014). Better together: Usability and security go hand in hand. *IEEE Security and Privacy*, 12(6), 89–93.

Dhillon, G., Oliveira, T., Susarapu, S. and Caldeira, M. (2016). Deciding between information security and usability: Developing value-based objectives. *Computers in Human Behavior,* 61, 656–666.

Dodier-Lazaro S., Sasse, M. A., Abu-Salma, R. and Becker, I. (2017). From paternalistic to user-centered security: Putting users first with value-sensitive design. In: *CHI 2017 Workshop on Values in Computing*, pp. 7.

Feitosa, D., Ampatzoglou, A., Avgeriou, P. and Nakagawa, E. (2015). Investigating quality trade-offs in open source critical embedded systems, In: *Proceedings of 11th International ACM SIGSOFT Conference on Quality of Software Architectures* pp. 113–122.

Feth, D. (2015). User-centric security: Optimization of the security-usability trade-off, In: *Proceedings of 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering ESEC/FSE'15, August 30 – September 4, 2015, Bergamo, Italy*, ACM Press, pp. 1034–1037.

Florêncio, D., Herley, C. and van Oorschot, P.C. (2014). An administrator's guide to Internet password research. In: *Proceedings of the USENIX LISA 2014* Seattle, WA pp. 24–29.

Garfinkel, S. and Lipford, H. R. (2014). *Usable Security: History, Themes, and Challenges*. San Rafael, California, Morgan and Clay Publishers.

Garfinkel, S. L. (2005). *Design principles and patterns for computer systems that are simultaneously secure and usable*. Department of Electrical Engineering and Computer Science Cambridge, MA: Massachusetts Institute of Technology. PhD Thesis.

Garg, H., Choudhury, T., Kumar, P. and Sabitha, S. (2017). Comparison between significance of usability and security in HCI. In: *Proceedings of 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)*. Ghaziabad, India: IEEE, pp. 1–4.

Gaw, S., Felten, E.W. and Fernandez-Kelly, P. (2006). Secrecy, flagging, and paranoia: Adoption criteria in encrypted email. In: *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 591–600.

Glaser, B. G. and Strauss, A. L. (1967). *The Discovery of Grounded Theory Strategies for Qualitative Research*. Mill Valley, CA: Sociology Press.

Glass, B. D., Jenkinson, G., Liu, Y., Sasse, M. A., Stajano, F. and Spencer, M. (2016). The usability canary in the security coal mine: A cognitive framework for evaluation and design of usable authentication solutions. In: Proceedings of 2016 European Workshop on Usable Security (EuroUSEC), Darmstadt, Germany: Internet Society.

Gorski, P., Zezschwitz, E., Iacono, L. and Smith, M. (2019), On providing systematized access to consolidated principles, guidelines and patterns for usable security research and development. *Journal of Cybersecurity*, 5(1), 1–19.

Grassi, P. A., Newton, E. M., Perlner, R. A., Regenscheid, A. R., Buff, W. E., Richer, J. P., Lefkovitz, N. B., Danker, J. M., Choong, Y.-Y., Greene K. and Theofanos M. (2017). Digital Identity Guidelines: Authentication and Lifecycle Management. US Dept of Commerce, National Institute of Standards and Technology (NIST). Special Publication. Report number: NIST SP-800–63B.

Green, M. and Smith, M. (2016). Developers are not the enemy: The need for usable security APIs. *IEEE Security and Privacy*, 14(5), 40–46.

Haley, C. B., Moffett, J. D., Laney, R. and Nuseibeh, B. (2006). A framework for security requirements engineering, In: *Proceedings of the 2006 International Workshop on Software Engineering for Secure Systems*. Shanghai China: ACM, pp. 35–42.

Haoues, M., Sellami, A., Abdallah, H. and Cheikhi, L. (2017). A guideline for software architecture selection based on ISO quality related characteristics. *International Journal of System Assurance Engineering Management*, 8(2), 886–909.

Hausawi Y. and Allen. W. (2014), An assessment framework for usable security based on decision science. In: *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Crete, Greece: Springer, pp. 33–44.

Hevner, A.R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2), 87–92.

Hevner, A. R., March, S. T. and Park, J. (2004). Design research in information systems research. *MIS Quarterly*, 28(1), 75–105.

Hong. J. (2012), State of phishing attacks. *Communications ACM*, 55(1), 74–81.

IBM. (2018). *Cost of Data Breach Study*: *Global Analysis*. Ponemon Institute LLC. Available from: https://www.ibm.com/downloads/cas/AEJYBPWA [Accessed 28th June 2018].

International Organization for Standardization (2011). ISO 25010. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*.

International Organization for Standardization (2013). ISO/IEC 27001:2013. *Information technology – Security techniques – Information security management systems – Requirements.*

International Organization for Standardization (2019). ISO 9241-210. *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*.

Kelley, P. G., Komanduri S., Mazurek M., Shay R., Vidas T., Bauer L., Christin, N., Cranor, L.F., Lopez, J. (2012). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In: *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12)*. Washington, DC: IEEE Computer Society, pp. 523–537.

Kirlappos, I. and Sasse, M. A. (2014). What usable security really means: Trusting and engaging users. In: *Human Aspects of Information Security, Privacy, and Trust*. Cham: Springer International Publishing, pp. 69–78.

Kulyk, O., Neumann, S., Budurushi, J. and Volkamer, M. (2017). Nothing comes for free: How much usability can you sacrifice for security? *IEEE Security and Privacy*, 15(3), 24–29.

Lapsey, P. (2013). *Exploding the Phone*. New York: Grove Press.

Lew, P., Olsina, L., Becker, P. and Zhang, L. (2012). An integrated strategy to systematically understand and manage quality in use for web applications. *Requirements Engineering*, 17, 299–330.

Lew, P., Olsina, L. and Zhang, L. (2010). Quality, quality in use, actual usability and user experience as key drivers for web application evaluation. In: *Proceedings of International Conference on Web Engineering ICWE 2010*. Vienna, Austria Springer, pp. 218–232.

Mahlke, S. (2007). User experience: Usability, aesthetics and emotions in human-technology interaction. In: Law E., *et al*., Arnold Vermeeren, Marc Hassenzahl, & Mark Blythe eds. *Towards a UX Manifesto: COST294-MAUSE Affiliated Workshop*. Lancaster, UK pp. 26–30.

Mehta, R., Ruiz-López, T., Chung, L. and Noguera, M. (2013). Selecting among alternatives using dependencies: An NFR approach, In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. Coimbra Portugal: ACM pp. 1292–1297.

Mohajan, H. K. (2018). 'Qualitative research methodology in social sciences and related subjects', *Journal of Economic Development, Environment and People,* 7(1), 23–48.

Mor, Y., Winters, N. and Warburton, S. (2010). *Participatory Patterns Workshops Resource Kit. Version 2.1* [Online]. Available from https://hal.archives-ouvertes.fr/hal-00593108/document [Accessed 25th January 2018].

Morville, P. (2004). *User Experience Design, Semantic Studios* [Online]. Available from http://semanticstudios.com/user_experience_design [Accessed 25th January 2017].

Mujinga, M., Eloff, M. M. and Kroeze, J. H. (2019). Towards a framework for online information security application development: A socio-technical approach. *South African Computer Journal*, 32(1), 24–50.

Naqvi, B., Clarke, N. and Porras, J. (2020). Incorporating the human facet of security in developing systems and services. *Information and Computer Security*, (in press).

Naqvi, B. and Porras, J. (2020). Usable security by design: A pattern approach. In: Moallem, A. ed. *HCI for Cybersecurity, Privacy and Trust: Second International Conference, HCI-CPT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings*. Copenhagen, Denmark: Springer, pp. 609–618.

Naqvi, B., Porras, J., Oyedeji, S. and Ullah, M. (2020). Towards identification of patterns aligning security and usability. In: Abdelnour Nocera, J., Parmaxi, A., Winckler, M., Loizides, F., Ardito, C., Bhutkar, G., Dannenmann, P., eds. *Beyond Interactions, INTERACT 2019 IFIP TC 13 Workshops, Paphos, Cyprus, September 2–6, 2019, Revised Selected Papers*. Cyprus: Springer, pp. 121–132.

Naqvi, B. and Seffah, A. (2019). Interdependencies, conflicts and trade-offs between security and usability: Why and how should we engineer them? In: Moallem A. ed. *HCI for Cybersecurity, Privacy and Trust: First International Conference, HCI-CPT 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings*. Lecture Notes in Computer Science. Vol. 11594. Cham: Springer, pp. 314–324.

Naqvi, B. and Seffah, A. (2018). A methodology for aligning usability and security in systems and services, In: *2018 3rd International Conference on Information Systems Engineering (ICISE)*. Shanghai: IEEE, pp. 61–66.

Naqvi, B., Seffah, A. and Abran, A. (2020). Framework for examination of software quality characteristics in conflict: A security and usability exemplar. *Cogent Engineering*, 7(1), 1788308.

Naqvi, B., Seffah, A. and Braz, C. (2018). Adding measures to task models for usability inspection of the cloud access control services. In: *7th IFIP WG 13.2 International Working Conference, HCSE 2018, Sophia Antipolis, France, September 3–5, 2018, Revised Selected Papers*. Lecture Notes in Computer Science. Vol. 11262. France: Springer, pp. 133–145.

Neri, H. R. and Travassos, G. H. (2018), MeasureSoft-Gram: A future vision of software product quality. In: *ESEM '18: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. Finland: ACM/IEEE, pp. 1–4.

Nwokedi, U. O., Onyimbo, B. A. and Rad, B. B. (2016). Usability and security in user interface design: A systematic literature review. *International Journal of Information Technology and Computer Science*, 8(5), 72–80.

Parveen, N., Beg, R. and Khan, M. H. (2014). Integrating security and usability at requirement specification process. *International Journal of Computer Trends & Technology*, 10(5), 236–240.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chaterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information System,* 24(3), 45–78.

Ramsdell, B. and Turner, S. (2010). *Secure/multipurpose internet mail extensions (S/MIME) version 3.2 message specification. RFC 5751 (Proposed Standard)* [Online]. Available from: http://www.ietf.org/rfc/rfc5751.txt [Accessed 25th January 2017].

Reuter A., Boudaoud K., Winckler M., Abdelmaksoud A. and Lemrazzeq W. (2020). Secure email – a usability study. In: Bernhard M. *et al*. eds. *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers*. Lecture Notes in Computer Science. Vol. 12063. Cham: Springer, pp. 34–46.

Rivera, B., Becker, P. and Olsina, L. (2012). Quality Views and Strategy Patterns for Evaluating and Improving Quality: Usability and User Experience Case Studies. *Journal of Web Engineering*, 15(5&6), 433–464.

Ruoti, S. and Seamons, K. (2019). Johnny’s journey toward usable secure email. *IEEE Security & Privacy*, 17(6), 72–76.

Sasse, M. A., Smith, M., Herley, C., Lipford, H. and Vaniea, K. (2016). Debunking security–usability trade myths. *IEEE Security and Privacy*, 14(5), 33-39.

Seffah, A. and Javahery, H. (2004). *Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces*. Chichester, England: John Wiley & Sons.

Sheng, S., Holbrook, M., Kumaraguru P., Cranor, L. F. and Downs, J. (2010). Who falls for phish?: A demographic analysis of phishing susceptibility and effectiveness of interventions. In: *CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Atlanta, GA: ACM, pp. 373–382.

Supakkul, S., Hill, T., Chung, L., Tun, T. T. and Leite, J. C. S. (2010). An NFR pattern approach to dealing with NFRs. In: *Proceedings of 18th IEEE International Requirements Engineering Conference (RE)*. Sydney, Australia: IEEE, pp. 179–188.

Suru, H. and Murano, P. (2019), Security and user interface usability of graphical authentication systems – a review. I*nternational Journal of Computer Trends & Technology*, 67(2), 17–36.

Tsai, J. Y., Egelman, S., Cranor, L. and Acquisti, A. (2011). Effect of online privacy information on purchasing behavior: An experimental study. *Information Systems Research*, 22(2), 254–268.

Ur, B., Leon, P. G., Cranor, L.F., Shay, R., Wang, Y. (2012). Smart, useful, scary, creepy: Perceptions of online behavioral advertising. In: *SOUPS '12: Proceedings of the Eighth Symposium on Usable Privacy and Security*. New York: ACM, pp. 1–15.

Wang, Y., Rawal, B., Duan, Q. and Zhang, P. (2017). Usability and security go together: A case study on database. In: *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*. Tamilnadu, India: IEEE pp. 49–54.

Whitten, A. and Tygar, J. D. (1998). *Usability of Security: A Case Study*. School of Computing Science, Carnegie Mellon University. Report number: CMU-CS-98-155.

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer-Verlag Berlin Heidelberg.

Yıldırım, M. and Mackie, I. (2019). Encouraging users to improve password security and memorability. *International Journal of Information Security*, 18, 741–759.

Zagouras, P., Kalloniatis C. and Gritzalis, S. (2017). Managing user experience: Usability and security in a new era of software supremacy. In: Tryfonas, T. ed. *Human Aspects of Information Security, Privacy and Trust: 5th International Conference, HAS 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings.* Lecture Notes in Computer Science. Vol. 10292. Location: Publisher, pp. 174–188.

Zhang, Y., Monrose, F. and Reiter, M. K. (2010). Security of modern password expiration: An algorithmic framework and empirical analysis. In: *Proceedings of the 17th ACM Conference on Computer and Communications Security*. New York: ACM, pp. 176–186.

Zhu, M. X., Luo, X. X., Chen, X. H. and Wu, D. D. (2012). A non-functional requirements tradeoff model in Trustworthy Software. *Information Sciences,* 191, 61–75.

Appendix A: Survey I Questionnaire

SURVEY STATEMENT

Please share your experience on ‘usability of security’ with us. We are gathering information on how the users of mobile devices feel about the ‘usability of security’ of their device. The survey includes 10 basic questions. The data being collected will be used for research purpose only. The results of the survey will be publishable without any explicit reference to any person that participated in the survey. Report of the survey and publications are available free of charge to all participants upon request.

□ * I here-by agree to be a part of this survey.

* Please tick the check box to show your consent.

QUESTIONNAIRE

1. Please indicate
   - □ Name _____________
   - □ Email _____________

2. Please specify your age group
   - □ 21 and under
   - □ 22-34
   - □ 35-44
   - □ 45-54
   - □ 55-64
   - □ 65 and above
3. Please specify your attained education level
   - □ High School
   - □ Graduation
   - □ Post-Graduation
   - □ Other _____________
4. Please specify the field of study
   - □ Computer Science
   - □ Other _____________
5. Encryption is the process of using an algorithm to transform information to make it unreadable for unauthorized users.
   - □ Strongly disagree

- □ Disagree
- □ Neutral
- □ Agree
- □ Strongly Agree

6. How important is the confidentiality of data on mobile device to you
   - □ Hardly Matters
   - □ Somehow Matters
   - □ Matters
   - □ Important
   - □ Very Important
7. I have encrypted my smartphone/tablet to limit unauthorized disclosure of information in case of loss/theft.
   - □ Disagree
   - □ Neutral
   - □ Agree
8. I have locked access to my mobile device using one of the authentication mechanisms available in my device.
   - □ Disagree
   - □ Neutral
   - □ Agree
9. Which authentication mechanism do you prefer to use for limiting access to your mobile device?
   - □ PIN
   - □ Password
   - □ Pattern Based
   - □ Biometrics
   - □ Other ______________
10. I find security configuration of my mobile device easy to change and manage.
    - □ Strongly disagree
    - □ Disagree
    - □ Neutral
    - □ Agree
    - □ Strongly Agree

Appendix B: Survey II Questionnaire

SURVEY STATEMENT

Members of Software Engineering Team at LUT University, Lappeenranta, Finland, are conducting this survey to have your viewpoint on usability of security features in your smartphone. The survey has 22 questions in total and expected time to complete the survey is 5-7 minutes.

The survey is distributed in three parts, (1) related to demographic details of the participants, (2) related to specifics concerning usage of smartphones, and (3) on usability of security features available in smartphone. Kindly answer all questions. The data being collected will be used for research purpose only. The results of the survey will be publishable without any explicit reference to any person that participated in the survey.

The participation in the survey is voluntary and report of the survey is available free of charge to all participants upon request.

□ I here-by agree to be a part of this survey *

* Please tick the check box to show your consent

PART I
(Demographic Details)

1. Kindly specify your gender
   - □ Male
   - □ Female
2. Select the age group you belong to
   - □ <21
   - □ 21-30
   - □ 31-40
   - □ 41-50
   - □ >50
3. Your employment sector
   - □ Public Sector
   - □ Private
   - □ Other _______________
4. Level of completed education
   - □ High school

- □ Undergraduate (e.g. bachelor’s degree)
- □ Graduate (e.g. master’s degree)
- □ Doctorate (PhD)
- □ Other ________________

5. Your area of study
   - □ Business
   - □ Computer Science
   - □ Engineering
   - □ Medical
   - □ Other_______________

PART II
(Usage of Smartphones)

6. I have been using smartphones for
   - □ <5 years
   - □ 5-10 years
   - □ >10 years
7. The operating system (OS) on my current smartphone is
   - □ Android
   - □ Apple OS
   - □ Other ____________
8. I use smartphone for following purposes (choose as many as possible)
   - □ Only work related
   - □ Staying in contact with family/friends
   - □ Social media
   - □ Internet
   - □ Other _____________
9. I keep the following categories of data on my smartphone
   - □ Work related
   - □ Personal (pictures, contacts)
   - □ Both
10. Confidentiality is a characteristic that applies to information. To protect and preserve the confidentiality of information means to ensure that it is not made available or disclosed to unauthorized entities.
    How important is the confidentiality of data on smartphone for you
    - □ Not important
    - □ Slightly Important

- □ Moderately important
- □ Important
- □ Very important

11. In event of loss/theft of my smartphone, I feel that I am protected in terms of unauthorized disclosure of data
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
12. I try changing the security configuration (settings) of my smartphone.
    - □ Never
    - □ Very rarely
    - □ Rarely
    - □ Occasionally
    - □ Frequently
    - □ Very frequently
13. I am aware of the availability of authentication mechanisms (such as PIN, passwords etc.) in my smartphone
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
14. Which authentication mechanism do you prefer to limit access to your smartphone
    - □ PIN
    - □ Password
    - □ Pattern lock
    - □ Biometric (fingerprint, face recognition, iris scan etc.)
    - □ Other _____________

    Any specific reasons for selection of a particular method over others? (Optional)

PART III
(Usability of Security)

Based on your feedback the following parameters (Appropriateness recognizability, Learnability, Operability, Understandability and Findability) will help us evaluate the usability of security of your smartphone.

Appropriateness recognizability: "degree to which users can recognize whether a product or system is appropriate for their needs"

15. I can recognize that security features in my smartphone are appropriate to keep me safe in case loss/theft of my device

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree

Learnability: "degree to which a product or system enables the user to learn how to use it with effectiveness, efficiency in a specified context of use"

16. I can manage/change the security configuration of my smartphone effectively and efficiently

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree

Operability: "degree to which a product or system is easy to operate, control and appropriate to use."

17. I can encrypt my smartphone's memory

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree

18. I can perform all other security related tasks (e.g. changing the password, view certificates of a website, etc.) on my smartphone with ease and control

□ Strongly disagree

- □ Disagree
- □ Undecided
- □ Agree
- □ Strongly agree

Understandability: “degree to which a normal user can understand the terminologies and available information”

19. I understand the use of ‘digital certificates’ in my smartphone
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
20. I understand what are ‘trust agents’ in my smartphone
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

Findability: “degree to which the information is findable and easy to navigate”

21. I can change my smartphone’s authentication mechanism (PIN, password) without having to go through other options available in the settings in order to locate the intended procedure
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
22. I can locate other security procedures (such as encrypting the phone memory, viewing installed certificates) on my smartphone without having to go through other options available in the settings in order to locate the intended procedure
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree

□ Strongly agree

Any other comments? (Optional)

Thank you for your response

In case you need a copy of results, mention your email

Email: _______________________________

Appendix C: Survey III Questionnaire

SURVEY STATEMENT

We at LUT Software Engineering are conducting this survey to assess the information security awareness of employees as part of a project titled ‘Creating an Integrative Framework for Enhanced Usability of Security Services and Increased Business Potential’ funded by Business Finland.

The focus of the survey is on human aspects relevant to information security. The survey is to be answered anonymously, and the data collected will be used for research purposes. The results of the survey will be publishable without explicit reference to the person who participated in the survey. The survey is divided into 5 focus areas, which are clearly mentioned in the questionnaire. Overall, there are 33 questions and expected time to complete the survey is 5-7 minutes.

Kindly answer all questions.

□ I here-by agree to be a part of this survey *

* Please tick the check box to show your consent

1. Focus Area: Authentication

Practice: Reuse of passwords

1. It is acceptable to keep my work password same as my other personal accounts such as social media
   - □ Strongly disagree
   - □ Disagree
   - □ Undecided
   - □ Agree
   - □ Strongly agree
2. It is safe to use the same password for work and social media
   - □ Strongly disagree
   - □ Disagree
   - □ Undecided
   - □ Agree
   - □ Strongly agree
3. I use the same password for work and social media
   - □ Strongly disagree
   - □ Disagree

- □ Undecided
- □ Agree
- □ Strongly agree

Practice: <u>Strong passwords</u>

4. A work password should be a combination of numbers, letters, special characters, etc.
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
5. It not safe to have a work password with just letters
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
6. I use a combination of letters, numbers, and symbols in my work password
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

Practice: <u>Memorizing passwords</u>

7. It is allowed to write passwords including the work password into a file (physical files, e.g. notebook, diary; virtual files, e.g. file on a computer) in case I do not remember one among them
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
8. It is safe to write down password in a file (may it be physical or virtual)
    - □ Strongly disagree
    - □ Disagree

- □ Undecided
- □ Agree
- □ Strongly agree

9. I write down my passwords in a file
   - □ Strongly disagree
   - □ Disagree
   - □ Undecided
   - □ Agree
   - □ Strongly agree

2. Focus Area: Email Security

Practice: Opening the attachments received from within organization

10. I am allowed to open the attachments received via email from senders within organization
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
11. It is always safe to open attachments received from senders within our organization
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
12. I do not always open the attachment even if the sender is from within our organization
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

Practice: Opening the attachment received from outside organization

13. It is permissible to open the attachments received from senders outside the organization domain
    - □ Strongly disagree

□ Disagree
□ Undecided
□ Agree
□ Strongly agree

14. It is unsafe to open the attachments received from outside the organization domain

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree

15. If the email from outside organization domain looks interesting, I open the attachment to have a look

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree

3. Focus Area: <u>Internet Use</u>

Practice: <u>Downloading content from Internet</u>

16. I am allowed to download any files on my work computer if they are relevant to my job.

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree

17. It is not always safe to download files on the work computer from Internet

□ Strongly disagree
□ Disagree
□ Undecided
□ Agree
□ Strongly agree
□

18. I download any files relevant to my job on my work computer

□ Strongly disagree
□ Disagree

- □ Undecided
- □ Agree
- □ Strongly agree

Practice: <u>Clicking advertisements on the Internet</u>

19. I am allowed to click ads while using Internet from my workplace
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
20. It is not always safe to click ads from the workplace
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
21. I click ads of my interest to from my work computer
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

4. Focus Area: <u>Anti-phishing efforts</u>

Practice: <u>Understanding of Phishing</u>

22. Phishing attacks can only be successful if I do what the attacker wants me to e.g. respond to the email
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
23. It is safe to respond to emails which I suspect could be phishing
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided

- □ Agree
- □ Strongly agree

24. I respond to all emails irrespective even if they look suspicious
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

Practice: <u>Posting work related content on social media</u>

25. I am aware that posting my work-related activities and events on social media increase the chance of phishing
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
26. Nothing bad happens if I post work-related activities and events on social media
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
27. I post regarding work activities and events on social media
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

5. Focus Area: <u>Use of Mobile Devices</u>

Practice: <u>Physical Security of Mobile Devices</u>

28. I keep my work laptop at all time when I am working from public places

    - □ Strongly disagree
    - □ Disagree
    - □ Undecided

- □ Agree
- □ Strongly agree

29. When working from public places, it is safe to leave my work laptop unattended for a minute
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
30. When working from public place I leave my laptop unattended
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

Practice: **<u>Use of public wi-fi</u>**

31. I am allowed to send sensitive work files via email when connected to a public Wi-Fi
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
32. It is risky to send sensitive work files via email when connected to a public Wi-Fi
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree
33. I send sensitive work files via email when connected to a public Wi-Fi
    - □ Strongly disagree
    - □ Disagree
    - □ Undecided
    - □ Agree
    - □ Strongly agree

Publication I

Naqvi, B., Clarke, N., and Porras, J.
Incorporating the Human Facet of Security in Developing Systems and Services

Reprinted with permission from

Information and Computer Security
(in press)

Publication II

Naqvi, B., Seffah, A., and Abran, A.
Framework for Examination of Software Quality Characteristics in Conflict: A Security and Usability Exemplar

Published under a Creative Commons Attribution (CC-BY) 4.0 license in
Cogent Engineering
ISSN 2331-1916, vol. 7, no. 1, pp. 1788308

Naqvi et al., *Cogent Engineering* (2020), 7: 1788308
https://doi.org/10.1080/23311916.2020.1788308

cogent
engineering

COMPUTER SCIENCE | RESEARCH ARTICLE

Framework for examination of software quality characteristics in conflict: A security and usability exemplar

Bilal Naqvi[1,2*], Ahmed Seffah[3] and Alain Abran[4]

Received: 03 March 2020
Accepted: 23 June 2020

*Corresponding author: Bilal Naqvi, LUT Software Engineering, LENS, LUT University, Yliopistonkatu 38 Lappeenranta 53850, Finland
Email: syed.naqvi@lut.fi

Reviewing editor:
Edward Keedwell, University of Exeter, Exeter, UK

Additional information is available at the end of the article

Abstract: Standards and best practices for software quality guide on handling each quality characteristic individually, but not when two or more characteristics come into conflict such as security and usability. The objectives of this paper are twofold: (a) to argue on the importance of handling the conflicts between quality characteristics in general; (b) to formulate a framework for conflict examination of the software quality characteristics, we do so while considering the specific case of security and usability. In line with the objectives, a framework called Pattern-oriented Design Framework (PoDF) was formulated. The PoDF provides a mechanism for identification of the conflicts, modeling the conflicts to illuminate the reason for their occurrence, and eliciting the suitable trade-offs between the conflicting characteristics. The suitable trade-offs are thus documented as design patterns. The patterns can assist developers and designers in handling the conflicts in other but similar context of use. To validate and instantiate the PoDF, two studies were conducted. Usable security patterns discovered as a result of the studies are also presented in the paper.

Bilal Naqvi

ABOUT THE AUTHOR

Bilal Naqvi, Doctoral Candidate in Software Engineering at LUT University, Finland. He has received a master's degree in Information Security from NUST, Pakistan. His research focus is on studying the interdependencies between various software quality characteristics with a special focus on conflicts and trade-offs between security and usability. He is interested in investigating the security and usability conflicts from the perspective of diverse systems and services. He has authored/co-authored more than 10 articles on security and usability conflicts, and a book on integrating security and usability in the user authentication process.

PUBLIC INTEREST STATEMENT

Software quality characteristics are highly interdependent. Interdependencies between some of these characteristics lead to conflicts where recommendations from the perspective of one characteristic are negatively affecting the other dependent characteristic. This article presents a framework for the examination of interdependencies and conflicts between quality characteristics. A specific case of the conflict between security and usability has been considered while formulating the framework. The framework governs the management of the conflicts right from their identification to their resolution and documentation of the suitable trade-offs. The article proposes to document the suitable trade-offs as reusable design patterns. The patterns can thus assist developers in managing the conflicts in other but similar contexts. The lessons learned from the exemplar discussed in this article can be useful in the management of the interdependencies between other quality characteristics.

Subjects: Computing & IT Security; Software Engineering & Systems Development; User Interface; Computer Science; General

Keywords: interdependencies; quality characteristics; security; trade-offs; usability; patterns

1. Introduction

The stated and implied quality needs of various stakeholders of a software system have traditionally been characterized as distinct and almost independent quality characteristics. All these characteristics have diverse meanings for different stakeholders (based on their own viewpoints), and, in various contexts, they do not have equal importance. Experts in security, safety, reliability, and usability have developed various approaches to ensure quality from their specific perspectives without regard to the possible impact on other characteristics. Moreover, the software developers of today have to deal with challenging characteristics such as privacy, accountability, and sustainability, which intensifies the need for quality engineering. An additional challenge from a quality engineering perspective is the management of the conflicts when two or more quality characteristics are negatively affecting each other. A typical example in this regard is the conflict between security and usability, where the implementation of recommendations from the security perspective might leave the system less usable and vice versa. Consequently, security engineers perceive usability as a huge threat. Similarly, user interface (UI)/user experience (UX) designers consider a highly secure system as a big constraint for developing usable UI and providing a rich UX (Yee, 2004).

Furthermore, in practice, management of the conflicts and identification of the suitable trade-offs relies on developer's skills (Braz et al., 2007; Caputo et al., 2016; Feitosa et al., 2015). From the perspective of security and usability exemplar discussed in this paper, it is worthwhile to state that security and usability have evolved independently as two different domains, therefore, expertise in both security and usability is hard to find in one person (Garfinkel & Lipford, 2014). With management of the conflicts being reliant on developers who are either expert in security or usability, there is a need for assisting system's developers and designers in the management of these conflicts. Otherwise, we risk developing secure systems which despite being secure against various external and internal attacks might still be susceptible to user mistakes leading to a security breach. This research advocates the use of design patterns for assisting the developers in the management of the conflicts. A design pattern encapsulates information regarding the conflict and suitable trade-offs for the developers to apply these patterns in a specific context of use. From the perspective of specific exemplar discussed in this paper, the patterns can assist security engineers and developers in assessing the usability of their security options and vice versa.

However, in line with the objectives of this paper, the relationships between all quality characteristics as identified by ISO 25010 product quality model (Systems and software engineering, 2011) will be discussed. The aim is to improve the existing body of knowledge by applying the lessons learned from the exemplar of security and usability conflicts in cases where other characteristics are in conflict. Therefore, to address the aim and objectives identified earlier, the following key issues need to be explored.

(i) What quality characteristics and underlying sub-characteristics are in conflict?
(ii) How can the conflicts between quality characteristics be identified and documented before development?
(iii) Can design patterns be used to disseminate best practices and suitable trade-offs between conflicting quality characteristics?

This paper reports on these issues while presenting a framework for the conflict examination of software quality characteristics and sub-characteristics. A framework called *Pattern-oriented*

Design Framework (PoDF) is presented, which is developed based on elements of design science research (DSR) methodology. The PoDF has been formulated while considering the specific case of security and usability to govern management of the conflicts in this case. The outcome of each iteration of the PoDF is documented as reusable design patterns. The patterns can be disseminated among other developers and designers to influence their decision-making abilities when it comes to the conflicts in other but similar contexts. This is also in line with the engineering practice of not reinventing the wheel. Furthermore, it is pertinent to state that PoDF is an evolved and extended version of the framework presented in (Naqvi & Seffah, 2019). The limitations in framework (Naqvi & Seffah, 2019) such as (1) lack of means for identification of the conflicts, (2) a methodology for elicitation of suitable trade-offs, (3) identification of various roles during each stage of the framework, and (4) various questions addressed during each layer have been addressed in the PoDF design.

The remainder of this paper is organized as follows: Section 2 presents the background and related research on relationships and conflicts between software quality characteristics. Section 3 discusses the security and usability conflict. Section 4 presents the proposed framework (PoDF) for handling the conflicts and documenting the trade-offs in the form of reusable design patterns. Section 5 presents the details of the studies conducted to validate and instantiate the PoDF. Section 6 presents the discussion and outlines future perspectives and, Section 7 concludes by providing a list of actions that can foster the research and the development of a better understanding of the conflicts.

2. Background and related work

Security, usability, accessibility, trustfulness, privacy, accountability, sustainability are important quality characteristics. Some of these characteristics have been largely investigated by different communities, including usability engineering in the Human–Computer Interaction (HCI) community, and sustainability design in the green IT community, to name a few. Parallel to research in academia, the International Organization for Standardization (ISO) introduced several quality standards, such as ISO 25010 (Systems and software engineering, 2011), ISO 9241–11 (Ergonomics of human-system interaction, 2010), among others. ISO 25010 defines and identifies each quality characteristic individually without regard to the possible impact of the defined characteristics on each other. However, some of the quality characteristics are interrelated. For instance, *security* and *usability, performance efficiency* and *usability, security* and *compatibility*, among others.

To illustrate the existence of conflicts and trade-offs, an example featuring passwords is presented, which identifies a conflict between security, usability, and their associated sub-characteristics. The security dimension suggests that the passwords should be sufficiently long, frequently changed, have different case and special characters, etc. However, from the user (usability) point of view, such passwords are hard to memorize. If the suggested security guidelines are implemented, they have an adverse impact on the usability of the system, and if they are not implemented the system security might be at stake. Considering the sub-characteristics in conflict, the example features a conflict between *authentication* (a security mechanism) and *memorability* (a usability element). Another instance of a conflict between security and usability features the conflict between *confidentiality* (a security goal) and *feedback* (a usability element) while implementing password masking. Password masking is implemented in most of the authentication mechanisms to protect against shoulder surfing but at the cost of usability element of “feedback”. Consequently, in case of a mistake a legitimate user has to re-type complete password without knowing (feedback) and correcting the mistake only.

Furthermore, to illuminate the existence of conflicts between characteristics other than security and usability the following example is presented, which features a conflict between *performance efficiency* and *usability*. Developers trying to improve the performance efficiency of software systems often equate transparency with customer/user satisfaction, which in turn affects usability and user experience. From *usability* perspective, the user should be presented with a *feedback*, while such a *feedback* has an impact on *time behavior* and *resource utilization* from *performance efficiency* perspective. As an example, to keep the user updated with the status of the installations (i.e. feedback), there is an impact on the system's performance, as the system not only has to perform the main task but also has to keep the user updated with clear feedback.

Feitosa et al. investigated the trade-offs between sub-characteristics concerning the safety of a critical embedded system (Feitosa et al., 2015). Their empirical investigation shows that the trade-offs are usually in favor of critical quality characteristics. However, the work is limited to the identification of conflicts.

Zhu et al. proposed a model of fuzzy soft goal interdependency graphs (Zhu et al., 2012). The model uses qualitative and quantitative approaches to describe, analyze and evaluate the alternatives to certain quality characteristics (sometimes referred to as NFRs–Non-Functional Requirements) and the relationships among them. It facilitates making trade-off decisions among the competing NFR alternatives. The tool can help in studying or at least documenting the conflicts.

Dabbagh and Lee suggested an approach for prioritizing quality characteristics based on their relative importance to stakeholders (Dabbagh & Lee, 2013). Their approach analyzes the relationships between these characteristics to provide the developers with a prioritized list of quality characteristics. The nature of the relationships described by this approach can be investigated to see whether it leads to conflicts or not.

Other researchers have investigated prioritization and conflict resolution between quality characteristics with design patterns. For example, Mehta et al. introduced a pattern-based approach to analyze the dependencies among selection alternatives that may potentially affect the quality characteristics (Mehta et al., 2013). They classify the possible dependencies into various types, such as partial vs. total, mandatory vs. optional. They argue that their approach could help in making better selections among alternatives. Supakkul et al. presented four kinds of NFR patterns for capturing and reusing knowledge of NFRs. These patterns enable visualizing NFRs and manage synergy and conflict among them (Supakkul et al., 2010)).

Henningsson and Wohlin highlight that the overall quality is a complex combination of many characteristics (Henningsson & Wohlin, 2002) . These characteristics have different meanings and importance for different people and in different projects. The authors state, "the actual nature of relations between the characteristics are mostly poorly understood". Organizations and developers must cope with these relations in their daily software development. Neri and Travassos identified that there is empirical evidence on multidimensional linkage between software quality characteristics and that the one-dimensional perspective limits their use in continuous software engineering environments (Neri & Travassos, 2018).

Zulzalil et al. used an experience-based approach and an online survey to gather the findings regarding relationships between quality characteristics for web-based applications (WBA). The authors identified three types of relationships between quality characteristics: positive, negative and independent (Zulzalil et al., 2008). Haoues et al. during their research on establishing guidelines for the selection of appropriate software architecture also identified that relationships and dependencies exist between quality characteristics. The authors categorized the relationship in four categories: (1) positive "+" e.g., *security* and *reliability*, (2) negative "-" e.g., *security* and *performance efficiency*, (3) positive-negative "±" e.g., *usability* and *performance efficiency*, which is "+" in case of *appropriateness recognizability* (usability sub-characteristic) and *time behavior* (performance efficiency sub-characteristic), and "-" in case of *user error protection* (usability sub-characteristic) and *resource utilization* (performance efficiency sub-characteristic), and, (4) independent '0′ e.g., *performance efficiency* and *functional suitability* (Haoues et al., 2017). Furthermore, Aldaajeh et al. identified that the relationship between quality characteristics is one of the critical aspects for formulating suitable trade-offs and to achieve quality. However, the authors extend their argument to claim that, "unfortunately, quality attributes relationships' nature is poorly explored" (Aldaajeh et al., 2012).

Based on the analysis of literature (Zulzalil et al., 2008; Aldaajeh et al., 2012; Haoues et al., 2017) and ISO 25010 standard (Systems and software engineering, 2011), the relationships between software quality characteristics are presented in Table I. The characteristics listed in Table I have been considered in the same way as identified and defined by the *product quality model* of the ISO 25010 standard. In Table 1, the following types of relationships between quality characteristics are identified.

- *Positive "+"*—Relationship Definition: Supportive relationship. If characteristic X is enhanced, then Y will also be enhanced.
- *Negative "-"*—Relationship Definition: Conflicting relationship. If characteristic X is enhanced, then Y will be degraded.
- *Positive-Negative "±"*—Relationship Definition: "+" in case of some sub-characteristics of X and Y, and "-" in case of other sub-characteristics.
- *Independent '0'*—Relationship Definition: Independent relationship. Characteristic X and Y have altogether no impact on each other.

Furthermore, from the perspective of the key issues explored in this paper, it is worthwhile to highlight the following aspects:

(i) There are relationships between major quality characteristics.

(ii) There is a need to measure the degree of interdependency (qualitative/quantitative) between the identified characteristics related to each other.

(iii) There are inconsistencies between views of industry and academia concerning the relationship between certain quality characteristics, for example, from industry's perspective reliability has a "+" impact on usability, whereas, from academia's perspective the relationship between these characteristics is "±" (Zulzalil et al., 2008).

(iv) The existence of "±" relationships between some quality characteristics identifies the need to examine the relationship between the characteristics at a low-level, i.e., at the level of sub-characteristics. (Aldaajeh et al., 2012).

The PoDF proposed in this paper has been tailored while considering these aforementioned aspects. The purpose of PoDF is to govern the management of the conflicts. It does so by providing various means for identification of the conflicts, modeling the conflicts at the level of respective sub-characteristics, eliciting a suitable trade-off between conflicting characteristics while documenting the suitable trade-offs as patterns for use by other system developers and designers in similar contexts.

3. The security and usability conflict

Before presenting the framework, it is worthwhile to discuss the details of security and usability conflict.

3.1. Rationale

For almost two decades, security and usability have been identified as conflicting quality characteristics, which means there is a need to find an effective balance between them (Whitten & Tygar, 1998). ISO 25010:2011 model lists security and usability among the eight characteristics of the product quality model. ISO 25010 model defines security as "degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization". It is pertinent to state that the views about security are consistent among different standards and communities, with confidentiality, integrity, availability, etc., as its main goals; however, the same is not true for usability. There are two perceptions about usability in ISO 25010:2011 as identified by its product quality and quality in use models. The definition of usability that we consider in this paper is "degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (Systems and software engineering, 2011). The domain of research considering interdependencies, conflicts and trade-offs between usability and security is known as *usable security*.

It has been reported that the weakest link in security today is the human factor (Garfinkel & Lipford, 2014). Among the root causes of data breaches, the report published by IBM regarding the "Cost of Data Breach 2018" identifies that 27% of data breaches are caused due to human factors (IBM, 2018). While organizations employ a litany of security controls to limit the risk of becoming the victim of a security incident or breach, human error and human experiences are still factoring that cannot always be controlled. Furthermore, the report NISTIR 8080 by the National Institute of Standards and Technology (NIST) identifies that "the human element is a critical yet often overlooked component during technology integration [...], it is critical to understand users' primary goals, the characteristics of the users (both physical and cognitive characteristics), and the context in which they are operating" (Choong et al., 2016).

Moreover, different communities and interest groups have been studying the relationships between security and usability, including usable security community, the traditional computer security community, human–computer interaction (HCI) community, and the software engineering community. The study of security and usability dependencies by different communities and interest groups from their respective viewpoints has led to inconsistent perceptions. Consequently, recent research on usable security identifies an inconsistency between views on conflicts between security and usability. Traditionally, the existence of conflicts between the two has been accepted, but parallel to that some research also claims that the conflicts and trade-offs between security and usability are mere myths (Caputo et al., 2016; Cranor & Buchler, 2014; Sasse et al., 2016). The authors (Arteaga et al., 2009) while discussing the relationship between security and usability argue on the importance of integrating usability and security into a single design method, the authors state, "despite recognition, there is no or little attempt to integrate those two factors in a single design method. Some guidelines, recommendations, and best practices exist, but their effective integration remains the designer's responsibility".

Despite the recognition that the human element is critical to achieve effective security, much of the research work in usable security over the past decade suffers from a tactical approach (Garfinkel & Lipford, 2014), for example, CAPTCHAs pose readability problems, new CAPTCHAs were developed which required the user to select from a certain set of pictures; the fundamental question which remains unaddressed from a usable security perspective is do we need CAPTCHAs? Is it the responsibility of the users to protect the system against denial of service attacks that too proving themselves as humans? Moreover, the tactical approach addresses specific problems only and has limited use (Garfinkel & Lipford, 2014). The tactical solutions have a cosmetic effect and leave the need to have generalized solutions addressing this conflict. The question is whether these tactical efforts from a particular perspective are enough to address the conflict, or do we need a generalized approach and set strategic goals within the scope of the software development life cycle (SDLC) to study and solve the conflicts?

However, one positive aspect regarding usable security is that there is a growing emphasis on shifting the thinking from "the user is the problem and technology is the solution" to "the user must be part of the technology-based solution".

3.2. Interdependency between security and usability according to different quality views

The discussion on the interdependency between security and usability and the need for an acceptable trade-off between the two requires a broad explanation of the quality views concept.

ISO 25010:2011 model (Systems and software engineering, 2011) identifies two models for categorizing the quality characteristics, (1) the product quality model, and (2) the quality in use model. The product quality model has eight characteristics and focuses on conforming to the stated product requirements (Rivera et al., 2016), whereas the quality in use model has five characteristics and focuses on meeting the users' expectations while using the product. ISO defines quality in use as "the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use" (Systems and software engineering, 2011).

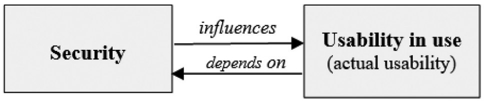
Furthermore, the ISO 25010:2011 model identifies three quality views, which are the *internal quality* view, the *external quality* (EQ) view, and the *quality in use* (QinU) view (Lew et al., 2010). The *internal quality* view is specified by the product quality model and can be evaluated using static attributes (such as requirement specifications, architecture, piece of code). The EQ view is specified by the product quality model and can be measured and evaluated by dynamic attributes (for example, running the code in a simulated environment). However, the QinU view is specified by the quality in use model and can be measured and evaluated by the degree to which the product meets the user's needs and expectations during actual use in its operating environment. The ISO model also identifies the relationships, namely "*influences*" and "*depends on*", between these views, i.e. between EQ and QinU and vice versa (Lew et al., 2012).

As stated earlier that security has been defined in a consistent way and with the same meanings by different communities and in different standards. However, this is not the case with usability, which makes usability a very confusing quality characteristic, and one which is most often measured using a subjective measurement scale. Despite listing usability as one of the eight characteristics in the product quality model, the ISO 25010:2011 standard defines usability as "as a subset of quality in use consisting of effectiveness, efficiency, and satisfaction, for consistency with its established meaning" in the QinU model. Therefore, to distinguish between the two perceptions about usability, usability in use is usually referred to as actual usability (Lew et al., 2010). In the context of usable security and the existence of a relationship between internal/EQ and QinU views, there exists an influences/depends on the relationship between security and actual usability (see Figure 1).

Figure 1 clarifies further the existence of dependency and the nature of the relationship between usability and security. The way security procedures are implemented as internal/EQ of the system determines and influences the usability level the end-user would be able to achieve. In the case of complex security systems, there is less usability, in fact, less actual usability. Therefore, when referring to the conflict between usability and security, it is mostly the interdependency between security and usability in use (actual usability), which has never been explicitly identified in existing studies (Zulzalil et al., 2008; Aldaajeh et al., 2012; Haoues et al., 2017). It is pertinent to mention that existing research related to usability and security does not clearly distinguish between different quality views, which adds to the contributions of this work.

The instances of conflict between sub-characteristics of security and usability in use (actual usability/actual user experience) (Rivera et al., 2016) are presented in Table 2, where "x" in the Table represents the existence of a conflict between the sub-characteristics. The sub-characteristics of security and usability mentioned in Table 2 have been considered the same way as identified and defined by ISO 25010 model. Garfinkel and Lipford discuss various themes and challenges in the

Figure 1. Relationship between usability and security in terms of quality views.



domain of usable security research while also identifying the conflicts between security and usability in general, without mentioning the affected sub-characteristics (Garfinkel & Lipford, 2014). We performed an analysis of various instances of the conflicts reported in the literature to identify the relevant sub-characteristics in conflict.

As an example, there is a conflict between *authenticity* and *satisfaction*. Satisfaction considers "the user's response to interaction with the product or system and includes attitudes towards the use of the product"; however, complex authenticity mechanisms like strong passwords, false rejection rates (FRR) in case of biometrics significantly affect user's attitude towards the product, ultimately affecting satisfaction. Similarly, a study (Imperva, 2010) presents the results of an analysis of 32 million passwords for a web service, among which 1% were mere "123,456", and around 20% of the passwords were the user's name, slang or a common dictionary word. Moreover, the conflicts been *authenticity* and *efficiency* arise in case of graphical passwords schemes, where authenticating to the graphical passwords can take longer than the text passwords (Garfinkel & Lipford, 2014).

4. Framework for examination of the quality characteristics in conflicts

4.1. Approach for the development of the framework

The approach used for the development of the PoDF is design science research (DSR). Design science research is a research methodology involving the design and investigation of the artifacts in a particular context (Wieringa, 2014). The design science research methodology guides the design of *artifacts* (patterns) and *processes* (framework-PoDF). Moreover, the design science research methodology supports the iterative model of development, which means the building of new and evolved processes and artifacts after the communication phase of the last completed iteration (Peffers et al., 2007). The essential aspect to consider during new iteration is the feedback recorded during the last iterations' communication phase; the feedback should be reflected in the evolved processes and artifacts. As stated earlier, the PoDF is an evolved version and extension of the framework presented in (Naqvi & Seffah, 2019). The key drivers considered while designing an evolved version were the feedback received during the presentation of the framework at the conference.

4.1.1. Method of development

The design science process used for the development of PoDF (process) and the identification of patterns (artifacts) is presented in Figure 2. The development process for the PoDF involved three cycles in line with the principles of design science research (Hevener, 2007) .

(i) *The relevance cycle*: The motivation behind this cycle is to improve the environment (software ecosystem) through the introduction of new artifacts (patterns) and processes (PoDF) for the construction of these artifacts. The artifacts are developed to facilitate the developers while handling the conflicts. As presented in Figure 2, the problem considered during the relevance cycle is the conflict between security and usability, and the evaluation criterion is to use the PoDF for a real-world usable security problem and discover a usable security pattern. The cycle iterates as much as it is required.

(ii) *The rigor cycle*: This cycle includes selection, application, and evaluation of knowledge bases to build and evaluate artifacts. Knowledge bases include theories, experiences, experts, and existing artifacts and processes. In this context, the knowledge base includes personal experiences, existing case studies, existing frameworks, and interviews of experts.

(iii) *The design cycle*: This is iterative and involves build and evaluate loop for artifact design both as product and as a process. The cycle iterates until the item is validated and new knowledge could be added to the knowledge base.

Table 1. Relationships between software product quality characteristics of ISO 25010 standard (Zulzalil et al., 2008) (Aldaajeh et al., 2012) (Haoues et al., 2017)

Product Quality Characteristics (ISO 25010-Product Quality Model)

	Functional suitability	**Performance efficiency**	**Compatibility**	**Usability**	**Reliability**	**Security**	**Maintainability**	**Portability**
Functional suitability		-	0	+	+	-	+	0
Performance efficiency	0		-	-	0	-	-	-
Compatibility	0	0		0	0	-	±	+
Usability	+	±	0		+	0	±	0
Reliability	+	0	0	+		0	+	0
Security	0	-	-	-	+		0	0
Maintainability	+	-	+	0	±	±		+
Portability	0	-	+	0	0	0	+	

Table 2. Conflicts between security and usability in use

Security sub-characteristics	*Usability in use* sub-characteristics		
	Effectiveness	Efficiency	Satisfaction
Confidentiality	x	x	
Integrity	x		x
Non-repudiation		x	x
Accountability	x		x
Authenticity		x	x

4.1.2. Artifacts

The artifacts, in this case, are the patterns. Patterns have shown their effectiveness to document the best practices addressing a common design problem. The term "pattern" is used here as introduced by Alexander in the 1980s, "each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice". Patterns provide real solutions and not abstract principles by explicitly mentioning the context and the problem and summarizing the rationale for their effectiveness. Since the pattern provides a generic "core" solution, its use can vary from one implementation to another.

Design patterns have been used to support a smooth integration and cross-pollination of communities (Seffah & Javahery, 2004). Patterns are recommended for improving communication among team members from different disciplines. They foster the development of a common language or vocabulary when explaining design; therefore, they can be helpful in the case of multidisciplinary fields like usable security, and in general, when two different quality characteristics are in conflict. The solution in the pattern will address one usable security problem in a particular context. It is therefore unrealistic to expect one pattern to solve more than one design problem. Moreover, the design patterns can prove to be effective in handling inconsistency of views between academia and industry by providing shared documentation in the form of patterns. The patterns' ability to evolve with time provides a common ground for incorporating several views, i.e., from industry and academia.

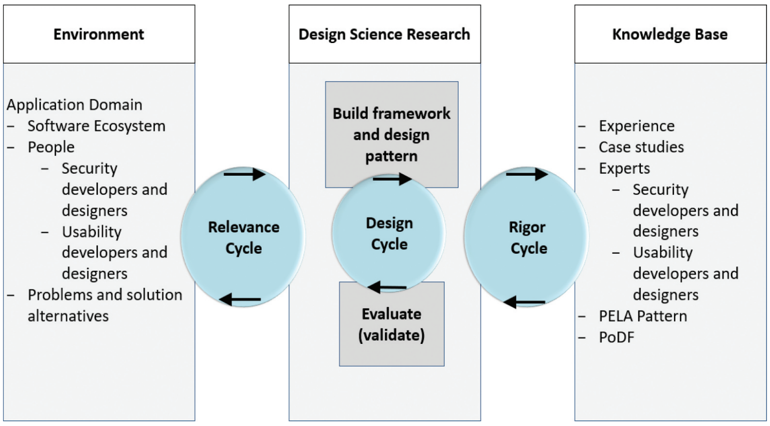

Environment
Application Domain
– Software Ecosystem
– People
– Security developers and designers
– Usability developers and designers
– Problems and solution alternatives
Relevance Cycle
Design Science Research
Build framework and design pattern
Design Cycle
Evaluate (validate)
Rigor Cycle
Knowledge Base
– Experience
– Case studies
– Experts
– Security developers and designers
– Usability developers and designers
– PELA Pattern
– PoDF

Figure 2. *Design science research process adopted and re-drawn in particular context* (2007)**.**

4.2. The pattern-oriented design framework (PoDF)

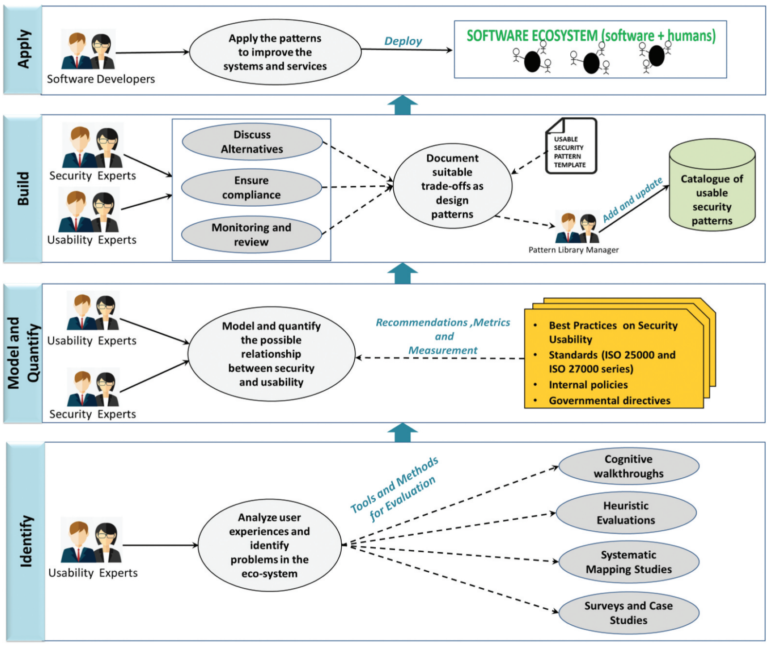
Figure 3 presents the pattern-oriented design framework PoDF proposed to handle the conflicts between security and usability. The PoDF follows a bottom-up layered architectural style. The first layer (*identify*) deals with the identification of the conflicts. The experts can utilize different tools and methods such as cognitive walkthroughs, surveys for identification of the conflicts. Once the conflicts are identified, security and usability experts model and quantify the possible relationship between security, usability, and their sub-characteristics. Recommendations from standards on security and usability, internal policies and governmental directives in specific contexts play a key role in the modeling of conflicts at the *model and quantify* layer. In the *build* layer, the security and usability experts brainstorm to discuss various solutions for eliciting the right trade-offs, once the suitable solution is identified it is documented as a pattern. To support reuse, the pattern is added to the catalog for use by other developers and designers in other but similar context of use. At the *apply* layer, the software developers apply these patterns to deliver simultaneously usable and secure systems.

The key questions considered during the four layers of PoDF include:

(i) **When do the conflicts occur? *(Identify)***

Analysis of diverse user experiences and tasks of the stakeholders and end-users that involve security technologies, modeling of the interaction between stakeholders and users to accomplish those tasks, while identifying the possible usability problems. The usability experts can utilize the following methods for identification of the security and usability conflicts.

Figure 3. The pattern-oriented design framework (PoDF) for security and usability.


Apply
Software Developers
Apply the patterns to improve the systems and services
Deploy
SOFTWARE ECOSYSTEM (software + humans)
Build
Security Experts
Usability Experts
Discuss Alternatives
Ensure compliance
Monitoring and review
Document suitable trade-offs as design patterns
USABLE SECURITY PATTERN TEMPLATE
Pattern Library Manager
Add and update
Catalogue of usable security patterns
Model and Quantify
Usability Experts
Security Experts
Model and quantify the possible relationship between security and usability
Recommendations ,Metrics and Measurement
Best Practices on Security Usability
Standards (ISO 25000 and ISO 27000 series)
Internal policies
Governmental directives
Identify
Usability Experts
Analyze user experiences and identify problems in the eco-system
Tools and Methods for Evaluation
Cognitive walkthroughs
Heuristic Evaluations
Systematic Mapping Studies
Surveys and Case Studies

- Cognitive Walkthrough, the usability experts inspect the user interfaces of security systems and services by going through a set of tasks and evaluating its understandability, ease of use and learning from the perspective of the targeted population.
- Heuristic Evaluations, which are conducted by experts to identify usable security problems due to violations of usability heuristics and security policies.
- Systematic mapping studies, the involved set of experts can conduct a systematic mapping study to identify conflicts based on published research and studies.
- Surveys and Case studies, involving end-users' feedback and qualitative assessment of the problems faced by users while using a security system or service.

(ii) **Why did the conflicts occur? *(Model and Quantify)***

Identification of the sub-characteristics and recommendations from security/usability perspective to illuminate why did the conflict occur. This stage involves two activities, (1) identification of the relevant sub-characteristics in conflict, and (2) assigning a severity scale to the conflicts based on elements of the quantitative methodology presented in Naqvi et al. (2018). In line with the first activity, a matrix with sub-characteristics of security (rows) and sub-characteristics of usability (columns) is drawn; the intersection in the matrix (cell) represents a potential conflict (see Figure 4).

It is pertinent to mention that the sub-characteristics of security and usability are added for exemplary purposes, more rows and columns can be added as per requirement. However, in line with the second activity during this stage, a three-staged methodology (Naqvi et al., 2018) would involve activities such as: (1) identification of goals from security and usability perspectives, (2) connecting the security goals with usability criteria, and, (3) formulating the usable security inspection method. After this stage, all security and usability conflicts are rated by three severity levels.

- *Major*: refers to catastrophic problems that should be given a high fixing priority level, they must be fixed before releasing the software.
- *Intermediate*: it is important to fix this type of problem as soon as possible.
- *Minor*: refers to problems with a low fixing priority level, which means that these problems should be fixed only if there is extra time available.

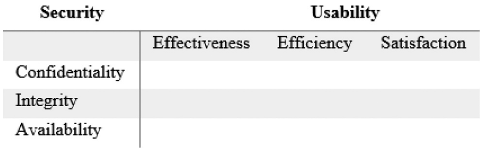
The identified conflicts are modeled keeping in view the best practices and standards on usability and security. Governmental directives might also come into play in specific contexts.

(ii) **How can a suitable trade-off be developed? *(Build)***

Building suitable trade-offs providing a balanced solution from the perspective of characteristics in conflict, and their documentation in the format of patterns. Elements of *risk-based approach* such as discussion and evaluation of alternate solutions, ensuring compliance with standards and best practices are applicable at this stage.

The security and usability experts discuss and evaluate different possible solutions to fix the problem under consideration while complying with the standards and best practices. The expertise

Figure 4. Matrix for describing a potential conflict.



Security	Usability		
	Effectiveness	Efficiency	Satisfaction
Confidentiality			
Integrity			
Availability			

of the professionals also comes into play during this stage; however, the finalized solution documented as a pattern is under monitoring and review by the experts and the developers. This is very much consistent with patterns' ability to evolve with time. The patterns can be used by participating organizations to enhance the usability of existing security technologies or the development of new ones. The documented patterns are added to the catalog of the patterns. Each pattern is documented on a standard template presented in (Naqvi & Seffah, 2019).

(iv) **Where can the identified solutions be deployed? *(Apply)***

Identification of the usable security problems of similar context and applying the recommendations by the pattern to solve the problem. As stated earlier, patterns provide real solutions and not abstract principles by explicitly mentioning the context and the problem and summarizing the rationale for their effectiveness; therefore, multiple implementations can be derived from a single pattern. Implementation aspects are purely dependent on the developers; however, the patterns provide a suggestion on how to avoid a conflict in a particular context.

Once the patterns are developed, they are disseminated among the community of developers and designers to influence their decision-making abilities in other but similar contexts. The software developers use these patterns to develop newer versions of their systems in other but similar contexts.

5. Validation

To validate and instantiate the PoDF, we conducted two studies involving practitioners from the industry and members of the Software Engineering Laboratory at LUT University. The objective of the studies was to test the PoDF and discover a pattern. The details of the studies are presented in subsequent sections.

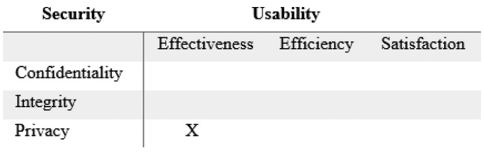
5.1. Study I (cognitive walkthrough)

- *Identify*: In this case study, we utilized cognitive walkthrough to identify a conflict between security and usability in case of smartphones. The specific case considered during the case study was when the smartphone user checks for weather updates or maps. For this purpose, all smartphones in use today require the enabling of the location awareness feature. Location awareness remains enabled even after the weather is updated, or the user reaches the destination in the case of maps. Thus, the user's problem is that in most cases the location awareness feature, once enabled, remains enabled for a long time even when it is not required (e.g., at home, in the gym, while sleeping, cooking, etc.). With the usability feature of preventing the user from enabling/disabling the location feature every time, the user's privacy/security is at stake.

Moreover, Minch (2004) discuss 13 privacy issues that arise from location awareness capability. From a security perspective, if the location awareness feature is enabled, then the adversary can read the location information in one of many ways. In addition, this location information is transmitted through apps, which can be eavesdropped or, in case of poorly protected servers of the service providers for the apps, can be gathered from there. While presenting an experimental study on location-based privacy breaches in Google apps, Liu et al., (2017) state, "these location-based services apps facilitate users in many application scenarios, but they raise concerns on the breach of privacy related to location access. Smartphone users can hardly perceive location access, especially when it happens in the background. In comparison to location access in the foreground, location access in the background could result in more serious privacy breach because it can continuously know a user's locations". The authors also point out that location recorded in the background can incur more serious implications from a privacy perspective, as it can collect more locations of the user.

- *Model and Quantify*: Based on the case description above, the security and usability experts detailed the goals required from their own perspectives and the sub-characteristics of security and usability in conflicts were identified (see Figure 5).

Figure 5. Matrix for describing a potential conflict between effectiveness in use and privacy.



Security	Usability		
	Effectiveness	Efficiency	Satisfaction
Confidentiality			
Integrity			
Privacy	X		

In the given context, the matrix presents a conflict between user privacy and effectiveness in use in the considered context. Due to stake of user privacy and security involved, the experts using the methodology proposed by Naqvi et al., (2018) assigned a '*intermediate*' severity level to the problem, which means that the problem is imperative to fix as soon as possible.

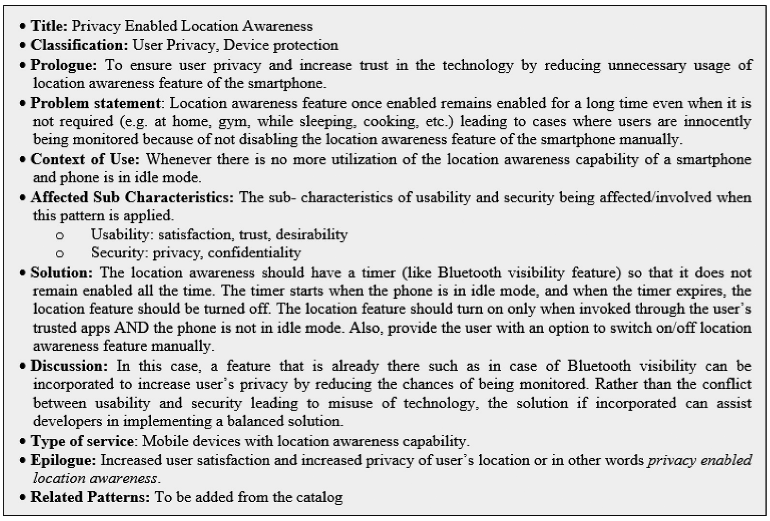
- *Build*: During this phase different options were considered to identify the suitable trade-offs and documented as pattern. As a result of the discussion, the Privacy Enabled Location Awareness (PELA) Pattern was documented (Figure 6).

The PELA pattern addresses the trade-off to user's privacy caused by a usability feature. If developers implement this pattern, it will result in the preservation of the privacy and security of the device as well as enhanced user trust and satisfaction. What seems evident in the case just discussed, is that security developers are working on ways to secure the dissemination of location information, and UI/UX designers are proposing location awareness to remain enabled so that it does not bother the user every time to enable and disable the feature. Therefore, some implementations may seem to be attractive but are compromising user's privacy in several ways.

5.2. Study II (Survey)

A survey was conducted to record user feedback on security of their mobile devices with the aim of identifying potential conflicts between security and usability in day-to-day use. *Ethical concerns* were considered during the survey and the users' consent was obtained before they answered the questions. Personal information that was recorded during the survey has been kept confidential and will not

Figure 6. Privacy enabled location awareness (PELA) pattern.



- **Title:** Privacy Enabled Location Awareness
- **Classification:** User Privacy, Device protection
- **Prologue:** To ensure user privacy and increase trust in the technology by reducing unnecessary usage of location awareness feature of the smartphone.
- **Problem statement**: Location awareness feature once enabled remains enabled for a long time even when it is not required (e.g. at home, gym, while sleeping, cooking, etc.) leading to cases where users are innocently being monitored because of not disabling the location awareness feature of the smartphone manually.
- **Context of Use:** Whenever there is no more utilization of the location awareness capability of a smartphone and phone is in idle mode.
- **Affected Sub Characteristics:** The sub- characteristics of usability and security being affected/involved when this pattern is applied.
  - Usability: satisfaction, trust, desirability
  - Security: privacy, confidentiality
- **Solution:** The location awareness should have a timer (like Bluetooth visibility feature) so that it does not remain enabled all the time. The timer starts when the phone is in idle mode, and when the timer expires, the location feature should be turned off. The location feature should turn on only when invoked through the user's trusted apps AND the phone is not in idle mode. Also, provide the user with an option to switch on/off location awareness feature manually.
- **Discussion:** In this case, a feature that is already there such as in case of Bluetooth visibility can be incorporated to increase user's privacy by reducing the chances of being monitored. Rather than the conflict between usability and security leading to misuse of technology, the solution if incorporated can assist developers in implementing a balanced solution.
- **Type of service**: Mobile devices with location awareness capability.
- **Epilogue:** Increased user satisfaction and increased privacy of user's location or in other words *privacy enabled location awareness*.
- **Related Patterns:** To be added from the catalog

be disclosed at any stage. The participation in the survey was voluntary and respondents were not paid. The online link containing the survey was disseminated using email, IMs, and social media. The inclusion criteria for the participants were limited to the users of smartphones/tablets, irrespective of make, model, and operating system of their devices. The survey consisted of 10 questions, and 75 respondents completed the survey. The number of questions in the survey was kept limited since the focus to identify the conflicts for the purpose of this study. The survey questionnaire is presented in Figure 7.

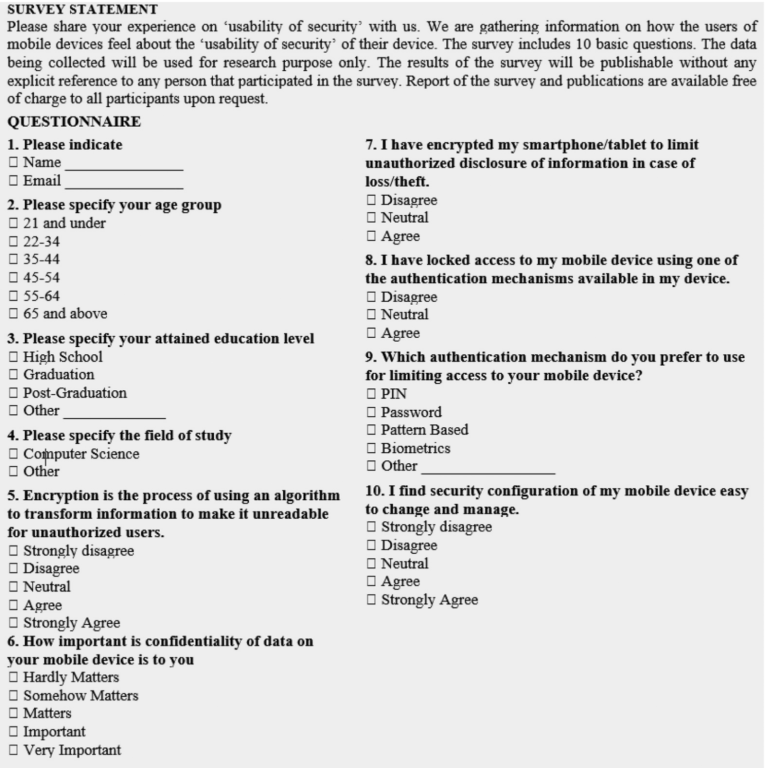
The details of the survey questionnaire are not discussed, since the focus is to illustrate the approach, not the survey. However, the key findings after analysis of the survey results include:

F1: The majority of respondents had an idea about data encryption with around 70% of them rating the confidentiality of their data between "important" and "very important".

F2: Besides understanding encryption and the importance of data confidentiality, only 32.7% of respondents with knowledge of encryption had encrypted their device.

F3: 94% the respondents locked access to their mobile device; pattern-based lock was the most common authentication method, followed by biometric authentication, passwords and PIN, respectively.

Figure 7. Validation study survey questionnaire.



SURVEY STATEMENT

Please share your experience on 'usability of security' with us. We are gathering information on how the users of mobile devices feel about the 'usability of security' of their device. The survey includes 10 basic questions. The data being collected will be used for research purpose only. The results of the survey will be publishable without any explicit reference to any person that participated in the survey. Report of the survey and publications are available free of charge to all participants upon request.

QUESTIONNAIRE

1. Please indicate
□ Name _______________
□ Email _______________

2. Please specify your age group
□ 21 and under
□ 22-34
□ 35-44
□ 45-54
□ 55-64
□ 65 and above

3. Please specify your attained education level
□ High School
□ Graduation
□ Post-Graduation
□ Other _____________

4. Please specify the field of study
□ Computer Science
□ Other

5. Encryption is the process of using an algorithm to transform information to make it unreadable for unauthorized users.
□ Strongly disagree
□ Disagree
□ Neutral
□ Agree
□ Strongly Agree

6. How important is confidentiality of data on your mobile device is to you
□ Hardly Matters
□ Somehow Matters
□ Matters
□ Important
□ Very Important

7. I have encrypted my smartphone/tablet to limit unauthorized disclosure of information in case of loss/theft.
□ Disagree
□ Neutral
□ Agree

8. I have locked access to my mobile device using one of the authentication mechanisms available in my device.
□ Disagree
□ Neutral
□ Agree

9. Which authentication mechanism do you prefer to use for limiting access to your mobile device?
□ PIN
□ Password
□ Pattern Based
□ Biometrics
□ Other _________________

10. I find security configuration of my mobile device easy to change and manage.
□ Strongly disagree
□ Disagree
□ Neutral
□ Agree
□ Strongly Agree

F4: 18.7% of the respondents did not find it easy to change the security configuration of their mobile device.

F5: Only 20% of the respondents used passwords for authentication, which is consistent with previous studies on desirability of passwords for human users (Garfinkel & Lipford, 2014).

- *Identify*: According to the survey results and finding F3, around 94% of the respondents locked access to their device by any means. The question from usable security perspective is how can a user of the smartphones and tablets authenticate to their device while cooking in the kitchen or working with work gloves on especially when the prevalent methods of authentication include fingerprint recognition, pattern-based locks, passwords on touch screens, etc. Therefore, it is a trade-off between security (authentication) and usability (ease of use, effectiveness, satisfaction, desirability).

Moreover, the existing work also identifies a similar problem, the user wishing to check a scheduled entry on her/his smartphone might find that entering the password takes longer than the task itself, which was to check the scheduled entry (Botha et al., 2009). It is all right from a security perspective, but from the usability point of view, this causes inconvenience. However, if authentication to the mobile devices is not enabled, there will be no concerns from usability perspective, but from a security perspective, this could lead to a breach of data and privacy in the case of loss or theft.

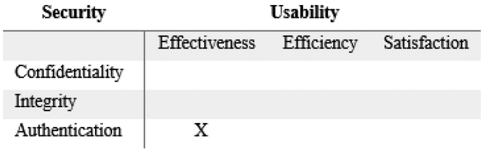
- *Model and Quantify*: As discussed earlier, the matrix describing the sub-characteristics in a conflict is presented in Figure 8.

The matrix represents a conflict between authentication and effectiveness in use. Taking into consideration the survey findings, the users are using authentication (less effectively though from usability perspective), there seem to be less security risks involved and the recommendations from usability do not pose a serious compliance issue, the experts using the methodology (Naqvi et al., 2018) assigned a '*minor*' severity level to the problem.

- *Build*: During this phase different options were considered to identify the suitable trade-off to be documented as pattern. As a result of the discussion, the Adaptable Authentication Pattern was documented (Figure 9).

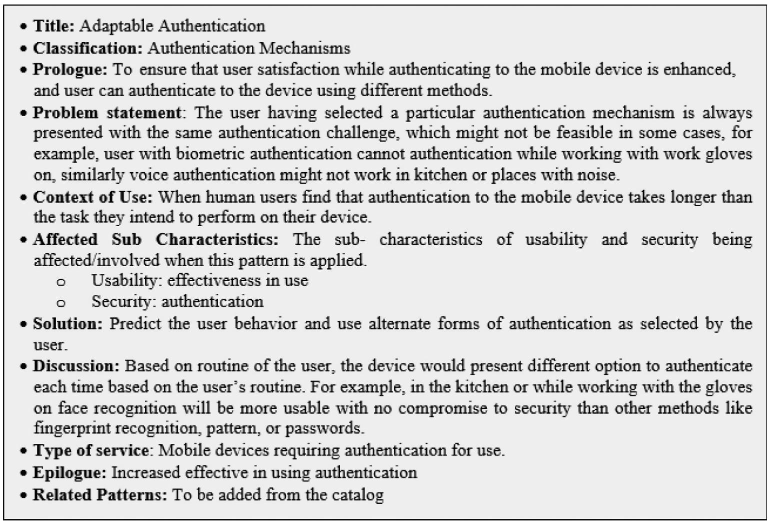
The Adaptable Authentication pattern suggests a method to achieve a balance between security and usability, where a user is able to authenticate to the mobile using multiple authentication methods while alternating between them, and using one and only one method at a time to grant access. For implementation purposes, an artificial intelligence tool that predicts the user's behavior and varies the form of authentication can work. Alternatively, another option would be an application that can ask the user about their routines and that presents the user with different authentication methods based on their routine. For example, at work, face recognition or voice recognition may be more feasible than passwords and fingerprint recognition. Similarly, during cooking, face recognition will be more usable in terms of elements of usability (with no compromise to security) than other methods like fingerprint recognition, pattern, or passwords. From the

Figure 8. Matrix for describing a potential conflict between effectiveness in use and authentication.



Security	Usability		
	Effectiveness	Efficiency	Satisfaction
Confidentiality			
Integrity			
Authentication	X		

Figure 9. Adaptable authentication pattern.



- **Title:** Adaptable Authentication
- **Classification:** Authentication Mechanisms
- **Prologue:** To ensure that user satisfaction while authenticating to the mobile device is enhanced, and user can authenticate to the device using different methods.
- **Problem statement**: The user having selected a particular authentication mechanism is always presented with the same authentication challenge, which might not be feasible in some cases, for example, user with biometric authentication cannot authentication while working with work gloves on, similarly voice authentication might not work in kitchen or places with noise.
- **Context of Use:** When human users find that authentication to the mobile device takes longer than the task they intend to perform on their device.
- **Affected Sub Characteristics:** The sub- characteristics of usability and security being affected/involved when this pattern is applied.
  - Usability: effectiveness in use
  - Security: authentication
- **Solution:** Predict the user behavior and use alternate forms of authentication as selected by the user.
- **Discussion:** Based on routine of the user, the device would present different option to authenticate each time based on the user's routine. For example, in the kitchen or while working with the gloves on face recognition will be more usable with no compromise to security than other methods like fingerprint recognition, pattern, or passwords.
- **Type of service**: Mobile devices requiring authentication for use.
- **Epilogue:** Increased effective in using authentication
- **Related Patterns:** To be added from the catalog

discussion above, it is evident that a solution in the form of a pattern is generic and two possible implementations could be derived from it, showing the concept of re-use.

6. Discussion and future work

The PoDF provides a mechanism not just for identification of conflicts but modelling the relationship between them and eliciting the suitable trade-offs, whereas the related work (Aldaajeh et al., 2012; Dabbagh & Lee, 2013; Feitosa et al., 2015; Sasse et al., 2016; Zhu et al., 2012) is limited to identification of the relationships between quality characteristics and prioritizing them. The main difference between PoDF and its previous version (Naqvi & Seffah, 2019) is that the PoDF presents, (1) various ways for identification of the conflicts, (2) includes a mechanism for assigning severity levels to the conflicts to assist modeling and quantification of the conflicts (Naqvi et al., 2018), and, (3) identifies the method for eliciting suitable trade-offs as design patterns based on elements of the risk-based approach. The question addressed during each layer of the framework has explicitly been presented based on the feedback received during the last iteration. Moreover, the PoDF has been tailored to be generic and adaptable for other quality characteristics as well.

Ferreira et al.,(2009) while listing 20 usable security patterns also presented the results after analysis of commonly used software browsers like Internet Explorer, Mozilla Firefox and email clients like Microsoft Outlook. It was also revealed that the identified patterns had a 61.67% application in the analyzed software implementations. The authors state "patterns make sense and can be useful guide for software developers". It is pertinent to state that the patterns presented in this paper are different from the ones presented by Ferreira et al.,(2009). The work byFerreira et al., was limited to listing the patterns and justifying their usage, however, this paper provides a mechanism for the identification of patterns, and a pattern template to standardize the documentation of patterns.

It is pertinent to state that PoDF has been designed specifically for security and usability conflicts, however, it can be generalized to derive patterns addressing the conflicts between other characteristics as well. The key activities and the questions addressed at each layer would remain the same; however, what would be different are the tools and methods for identification of conflicts, the guidelines and best practices, etc. For example, in case of *performance efficiency* and *maintainability*, instances of the conflicts can be identified using methods such as, heuristic evaluations, which are conducted by experts to identify problems due to violations of maintainability heuristics and performance expectations. Other methods include *contextual inquiry*, that consists of observing services and systems in use within the context of participants' daily activities and asking for explanations as interesting events arise, *semi-structured interviews*, online or on-site with the users of systems and services, among others.

Furthermore, from the perspective of specific exemplar discussed in the paper, the research on interdependencies (relationships between characteristics), conflicts (context and problems) and trade-offs (solutions and consequences) between quality characteristics may continue in the following directions. Firstly, the identification of more interdependencies and patterns requires an analysis of the academic literature and case studies from the industry. The second direction is to investigate the relationship between security and usability from the perspective of different quality views as discussed in Section 3.

6.1. Identification of more interdependencies and patterns

Further analysis of the literature and case studies from industry is required. The goal is to identify more interdependencies and discover more patterns. Documented patterns can be made accessible via web pages, as some collections of patterns are already available via web, e.g., HCI (welie.com), Gang of Four Patterns, etc. Other options for the dissemination of design patterns include pocketbooks for developers and designers, whitepapers, etc.; however, a preferred approach for disseminating patterns can be a web-based approach. A web-based interface to access the usable security patterns should present the following:

- A set of characteristics used to describe patterns; the differences between two patterns should be evident so that one pattern can be chosen over another in an informed manner.
- An explicit set of interrelations between patterns to categorize and inter-link usable security patterns.
- A digital database including data about the access and frequency of usage of specific patterns, which can be used to determine patterns' usefulness in terms of its application by the users of this database (as patterns are only patterns if they are re-used in a similar context). This may reveal the need for reformulation or dismissal of a pattern.

Usable security pattern writers are usually security and usability experts with a background in security and/or psychology, with a focus on usability and the human aspects of security. One problem that may arise in this regard is that usability experts prefer to use narrative formats to convey solutions to common user problems with supporting theories and concepts of interaction design and human factors. On the other hand, security developers need concise and pragmatic guidance through their design and coding activities. Rather than focusing on what should be presented in terms of information contents within usable security patterns, a fundamental challenge is how it should be packaged and appropriately offered to security developers to help them understand and apply the patterns correctly, and to record the feedback on the effectiveness and applicability as well as their usability, because usable security patterns should be usable too. One approach that needs to be mentioned is the Pattern Almanac (Rising, 2000). It is an attempt to make accessible (via a unifying user interface) a very large collection of all existing patterns and pattern languages. Several databases accessible via the Internet have also been proposed. However, these attempts fail in increasing the "ease to use and learning patterns" while making the pattern user experience a pleasant and enjoyable activity.

6.2. Formulating a proposal for catering usable security considering various quality views

One avenue for future research is to enhance the ISO 25010:2011. To our knowledge, there is no similar work related to *quality views* in the field of security and its associated characteristics. Concerning usability, a framework *internal/external quality, quality in use, actual usability and user experience (2Q2 U)* was proposed (Lew et al., 2010).

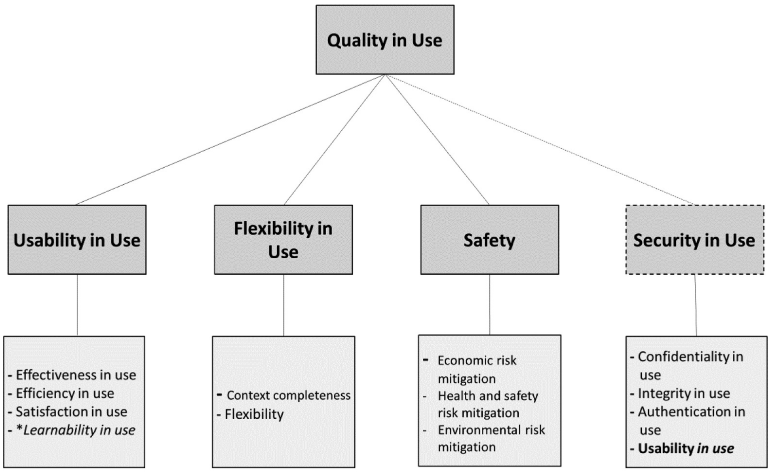
For the security and usability conflicts, it is imperative to investigate further the concept of quality views and the relationships between the views. The proposal is to add the *security in use* characteristic to the *quality in use* model. Neither usability nor its sub-characteristics as EQ and at the level of product quality model can be added to security as one of its sub-characteristics. Therefore, we plan to develop a strategic framework for adding *security in use* to the QinU model with *usability in use* as a subset of *security in use* (Figure 10). Since the usable security problem is relevant when the security features are being used, the appropriate place for designing an acceptable trade-off between them is thus also at the usage level. Adding *usability in use* as a subset of *security in use* would mean adding to security effectiveness, efficiency, satisfaction, usability in use compliance besides other sub-characteristics.

The general definitions of the terms in Figure 10 are available in (Systems and software engineering, 2011). Furthermore, the augmentation proposal is presented with a dotted line. It shows sub-characteristics like confidentiality in use, which would mean how efficaciously the procedures implemented in the system to preserve information/data from unauthorized disclosure. Similarly, integrity and authentication in use refer to the efficacy of implemented functions to ensure the integrity and implementing authentication, respectively. It also shows usability in use as a subset, which would mean adding the characteristics of usability to security. If software developers and designers can add elements like effectiveness in use, efficiency in use, etc., to security, then it can result in the implementation of simultaneously usable and secure solutions.

7. Conclusion

There is evidence as well as a collective agreement that software quality characteristics are highly dependent and often in conflict with each other. In line with the objectives of this research identified in the beginning, we justified the existence of interdependencies between quality characteristics and discussed the importance of handling such conflicts. A framework for the identification of conflicts and suitable trade-offs between conflicting characteristics was also presented while considering the specific case of security and usability. The lessons learned from the case of security and usability can be applied in the case of other quality characteristics in conflict. We also conducted two instantiation studies to validate the proposed approach.

Figure 10. Proposal for augmenting the QinU model.



Furthermore, investigation of the interdependencies, conflicts, and trade-offs is a timely required research problem, which requires the following actions:

(i) Building common ground and creating a unifying vocabulary across communities. One important force that complicates the situation is that the same concept is currently defined and perceived differently in the communities of researchers and practitioners, for example, different perceptions and definitions of usability across different communities. The same issue may arise in case of security and usability conflicts where the opinion is divided between the existence and non-existence of trade-offs.

(ii) Conducting internal and cross-corporation data collection to identify the current interdependencies, and how the trade-offs are being managed. The industry's best practices can prove to be valuable while designing the best design practices for the trade-offs.

(iii) Using patterns to document the identified conflicts and the best solutions for solving those conflicts using patterns, for example, usable security patterns. To our knowledge, very few patterns are available on the Internet. Gamification techniques with the complicity of crowdsourcing can assist in enabling the practitioners to join the efforts in building common ground in the form of a usable security pattern language.

(iv) Working on augmentation of ISO standards and related quality models such as ISO 25,000 and 27,000 for evaluating the interdependencies and conflicts for example, *security in use*.

Researching the interdependencies between quality characteristics needs the involvement of the entire software engineering community, including practitioners and standardization bodies. It requires bridging the gaps between the current research efforts made in different communities. There is a need for the software engineering community to create a cross-disciplinary research medium for discussing the definitions, perceptions, and understanding of conflicts between quality characteristics.

Funding
The authors received no direct funding for this research.

Author details
Bilal Naqvi[1,2]
E-mail: syed.naqvi@lut.fi
ORCID ID: http://orcid.org/0000-0001-5271-5604
Ahmed Seffah[3]
E-mail: ahmed.seffah@zu.ac.ae
Alain Abran[4]
E-mail: Alain.Abran@etsmtl.ca
ORCID ID: http://orcid.org/0000-0003-2670-9061
[1] Software Engineering, LENS, LUT University, Lappeenranta, Finland.
[2] Department of Software Engineering, Mirpur University of Science and Technology, MUST, Pakistan.
[3] Zayed University, Abu Dhabi, UAE.
[4] Department of Software and IT Engineering, École de Technologie Supérieure, Montreal, Canada.

Citation information
Cite this article as: Framework for examination of software quality characteristics in conflict: A security and usability exemplar, Bilal Naqvi, Ahmed Seffah & Alain Abran, *Cogent Engineering* (2020), 7: 1788308.

References

Aldaajeh, S., Asghar, T., Khan, A. A., & Ullah, M. (2012). Communing different views on quality attributes relationships' nature. *European Journal of Scientific Research*, *68*(1), 101–109. https://www.researchgate.net/publication/228449235_Communing_Different_Views_on_Quality_Attributes_Relationships'_Nature

Arteaga, J. M., Gonzalez, R. M., Martin, M. V., Vanderdonckt, J., & Rodriguez, F. A. (2009). A methodology for designing information security feedback based on user interface patterns. *Advances in Engineering Software*, *40*(12), 1231–1241. https://doi.org/10.1016/j.advengsoft.2009.01.024

Botha, R. A., Furnell, S., & Clarke, N. L. (2009). From desktop to mobile: Examining the security experience. *Computers & Security*, *28*(3–4), 130–137. https://doi.org/10.1016/j.cose.2008.11.001

Braz, C., Seffah, A., & M'Raihi, D. (2007). Designing a trade-off between usability and security: A metrics based model. In *Proceeding of IFIP Conference on Human-Computer Interaction* (pp. 114–126), Rio de Janeiro, Brazil.

Caputo, D. D., Pfleeger, S. L., Sasse, M. A., Ammann, P., Offutt, J., & Deng, L. (2016). Barriers to usable security? Three organizational case studies. *IEEE Security Privacy*, *14*(5), 22–32. https://doi.org/10.1109/MSP.2016.95

Choong, Y. Y., Greene, K., & Franklin, J. (2016). *Usability and security considerations for public safety mobile authentication* (NIST IR 8080). [Online]. https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8080.pdf

Cranor, L. F., & Buchler, N. (2014). Better together: Usability and security go hand in hand. *IEEE Security and Privacy*, *12*(6), 89–93. https://doi.org/10.1109/MSP.2014.109

Dabbagh, M., & Lee, S. P. (2013). A consistent approach for prioritizing system quality attributes. In *Proceeding of 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (pp. 317–322), Honolulu, HI, USA.

Ergonomics of human-system interaction. (2010). International Organization for Standardization (ISO Standard 9241).

Feitosa, D., Ampatzoglou, A., Avgeriou, P., & Nakagawa, E. (2015). Investigating quality trade-offs in open source critical embedded systems. "In *Proceeding of 11th International ACM SIGSOFT Conference on Quality of Software Architectures* (pp. 113–122), Montréal QC Canada.

Ferreira, A., Rusu, C., & Roncagliolo, S. (2009). Usability and security patterns. In *Proceeding of 2nd International Conference on Advances in Computer-Human Interaction* (pp. 301–305), Cancun, Mexico.

Garfinkel, S., & Lipford, H. R. (2014). *Usable Security, history, themes and challenges*. Morgan and Claypool Publishers.

Haoues, M., Sellami, A., Abdallah, H. B., & Cheikhi, L. (2017). A guideline for software architecture selection based on ISO quality related characteristics. *International Journal of System Assurance Engineering Management*, *8*(2), 886–909. https://doi.org/10.1007/s13198-016-0546-8

Henningsson, K., & Wohlin, C. (2002). Understanding the relations between software quality attributes – a survey approach. In *Proceeding of 12th International Conference for Software Quality* (pp.1–12). Canada.

Hevener, A. (2007). A three cycle view of design science research. Scandinavian Journal of Information Systems, 191, 87–92. https://aisel.aisnet.org/sjis/vol19/iss2/4

IBM. (2018). *Cost of data breach study: Global analysis*. Ponemon Institute LLC.

Imperva. (2010). *Consumer password worst practices*. Application Defense Center. [Online]. www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf

Lew, P., Olsina, L., Becker, P., & Zhang, L. (2012). An integrated strategy to systematically understand and manage quality in use for web applications. *Requirements Eng*, *17*(4), 299–330. https://doi.org/10.1007/s00766-011-0128-x

Lew, P., Olsina, L., & Zhang, L. (2010). Quality, quality in use, actual usability and user experience as key drivers for web application evaluation," In *Proceeding of International Conference on Web Engineering* (pp. 218–232), Vienna, Austria.

Liu, D., Gao, X., & Wang, H. (2017). Location privacy breach: Apps are watching you in background. In *Proceeding of IEEE 37th International Conference on Distributed Computing Systems* (pp. 2423–2429), Atlanta, GA, USA.

Mehta, R., Ruiz-López, T., Chung, L., & Noguera, M. (2013). Selecting among alternatives using dependencies: An NFR approach. In *Proceeding of 28th Annual ACM Symposium on Applied Computing* (pp. 1292–1297), Coimbra, Portugal.

Minch, R. P. (2004). Privacy issues in location aware mobile devices. In *Proceeding of 37th Annual Hawaii International Conference on System Sciences* (pp. 1–10), Big Island, Hawaii.

Naqvi, B., & Seffah, A. (2019). Interdependencies, conflicts and tradeoffs between security and usability: Why and how should we engineer them?. In *2019 1st International Conference HCI-CPT held as part of the 21st HCI International Conference* (pp. 314–324), Orlando, FL, USA, HCII.

Naqvi, B., Seffah, A., & Braz, C. 2018. Adding measures to task models for usability inspection of the cloud access control services In *Proceeding of 7th International Conference on Human Centered Software Engineering (HCSE)* (pp.133–145), Nice, France.

Neri, H. R., & Travassos, G. H. 2018. MeasureSoft-Gram: A future vision of software product quality. In *Proceeding of ACM International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1–4), Oulu, Finland.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chaterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information System*, *24*(3), 45–78. https://doi.org/10.2753/MIS0742-1222240302

Rising, L. (2000). *The pattern alamanac 2000*. Addison Wesley Publishing Company.

Rivera, B., Becker, P., & Olsina, L. (2016). Quality views and strategy patterns for evaluating and improving quality: Usability and user experience case studies. *Journal of Web Engineering*, *15*(5&6), 433–464. https://dl.acm.org/doi/abs/10.5555/3177218.3177222

Sasse, M. A., Smith, M., Herley, C., Lipford, H., & Vaniea, K. (2016). Debunking security–usability tradeoff myths. *IEEE Security and Privacy*, *14*(5), 33–39. https://doi.org/10.1109/MSP.2016.110

Seffah, A., & Javahery, H. (2004). *Multiple user inter-faces: Cross-platform applications and context-aware interfaces*. John Wiley & Sons Ltd.

Supakkul, S., Hill, T., Chung, L., Tun, T. T., & Leite, J. C. S. (2010). An NFR pattern approach to dealing with NFRs. In *Proceeding of IEEE International Requirements Engineering Conference (RE)* (pp. 179–188), Sydney, New South Wales, Australia.

Systems and software engineering – systems and software quality requirements and evaluation (SQuaRE) – system and software quality models. (2011). International Organization for Standardization (ISO Standard 25010).

Whitten, A., & Tygar, J. D. (1998). *Usability of security: A case study," School of Computing Science, Carnegie Mellon University* (Technical Report CMU-CS-98-155). http://reports-archive.adm.cs.cmu.edu/anon/1998/CMU-CS-98-155.pdf

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Yee, K. P. (2004). Aligning security and usability. *IEEE Security & Privacy*, *2*(5), 48–55. https://doi.org/10.1109/MSP.2004.64

Zhu, M. X., Luo, X. X., Chen, X. H., & Wu, D. D. (2012). A non-functional requirements tradeoff model in Trustworthy Software. *Information Sciences*, *191*, 61–75. https://doi.org/10.1016/j.ins.2011.07.046

Zulzalil, H., Ghani, A. A. A., Selamat, M. H., & Mahmod, R. (2008). A case study to identify quality attributes relationships for web based applications. *International Journal of Computer Science and Network Security*, *8*(11), 215–220. doi:10.1.1.474.4656&rep=rep1&type=pdf

Publication III

Naqvi, B., Porras, J., Oyedeji, S., and Ullah, M.
Towards Identification of Patterns Aligning Security and Usability

In: *INTERACT 2019 IFIP TC 13 Workshops*

Reprinted with permission from
Lecture Notes in Computer Science
LNCS 11930, pp. 121–132, 2020

Towards Identification of Patterns Aligning Security and Usability

Bilal Naqvi(✉), Jari Porras, Shola Oyedeji, and Mehar Ullah

Software Engineering, LENS, LUT University, 53850 Lappeenranta, Finland
syed.naqvi@student.lut.fi

Abstract. Academic research and existing implementations of various systems and services identify instances of conflict between security and usability. Engineering the right trade-offs between security and usability is often not an easy task. Engineering of such trade-offs is mainly reliant on developers' skills, who are either experts in security or usability. This research aims to assist the developers in engineering the right trade-offs by proposing the use of patterns. Patterns provide benefits like means of common vocabulary, shared documentation, reuse, among others. The use of patterns can assist security and usability developers by influencing their decision-making abilities when dealing with conflicts in other but similar context of use. For the identification of such patterns, the paper presents a three-stage methodology. To instantiate the methodology, a case study was conducted whose results are also presented in the paper.

Keywords: Security · Usability · Usable security · Patterns

1 Introduction

Security and usability are considered as conflicting goals [1]. The trade-offs between the two are discussed at different forums not limited to cyber-security and Human Computer Interaction (HCI). Typical examples of the security and usability conflict include (1) complex password guidelines having an impact on memorability, (2) implementation of password masking to protect against 'shoulder surfing attacks' but at the cost of feedback (usability element), among others.

Traditionally security and usability have evolved independently and as different domains, therefore, expertise in both security and usability is hard to find in one person [2]. Despite this, the developers are ones who face most of the criticism when the security solutions are unusable, or when usability features pose a threat to systems' security. The domain considering the integration of principles of security and aspects of usability is known as *usable security*.

The early efforts in the field of usable security date back to 1998 when different properties of usability problems relevant to the development of security systems were identified [3]. Despite this recognition, state of the art concerning usable security still has some catching up to do. Practices and trends followed in the large organizations reveal a

J. Abdelnour Nocera et al. (Eds.): INTERACT 2019, LNCS 11930, pp. 121–132, 2020.
https://doi.org/10.1007/978-3-030-46540-7_12

lack of motivation in considering usable security as a quality dimension [4]. One possible reason for this state is the cost associated with usable security [19]. The implementation of security due to the constantly evolving threat environment and usability due to rapid technological advancements has been so demanding that it leaves less time and costs to manage the trade-offs between the two. Among the other reasons for the current state of the art, it is imperative to discuss the following.

- *Different perceptions concerning security and usability*: The community has a different opinion concerning the existence of trade-offs between security and usability. Most of the research argues the existence of trade-offs between security and usability [5, 6]. However, in parallel with the research establishing the existence of the trade-offs, there is some research classifying security and usability trade-offs as mere myths [7, 8]. When the opinion on the existence of the problem is divided, then it is difficult to effectively contribute towards solving it.
- *Varying types of users:* In the community of users of the same device or application, opinions and requirements concerning security and privacy differ. Therefore, it is difficult to cater to the requirements of such a diverse category of users, which further complicates the task of finding common ground between security and usability and delivering a usable secure system.
- *Studying the conflicts by different communities in silos*: Various communities and interest groups have been studying usable security in silos, independently from each other. Some of these include, *(1)* SOUPS (Symposium on Usable Security and Privacy), small community studying trends, avenues and advancements in usable security. Much of the content is tactical, rather than being strategic, *(2)* The cybersecurity community dealing with the wider scope of security services; usability is a minor concern for this community, *(3)* The software engineering community where security and usability are considered as quality characteristics. Some of the standards provide contradictory perceptions and models for the same software quality characteristics, e.g. definition of usability in ISO 9126 and ISO 9241-11, *(4)* The HCI community, where the researchers try to explain from a cognitive perspective how users make poor security decisions leading to system compromises. There is no medium for collaboration that enables views from different communities and perspectives to be incorporated.
- *Ineffective joint working groups*: Because of independent activities, there is a lack of joint efforts concerning usable security. However, there exist multiple working groups specifically on usable security, but combining their findings to come up with a strategic vision for usable security, remains a challenge.
- *Lack of strategic approach*: Much of the work related to usable security suffers from a cosmetic approach that is the solutions are limited to specific problems, rather than contributing towards the management of the conflicts in general [2]. For example, there was a perception that CAPTCHA (*Completely Automated Public Turing Test to Tell Computers and Humans Apart*) poses *readability* problems for the users, therefore, new CAPTCHAS were developed that allow the user to select relevant images in response to the challenge. The question that remains valid for the community to address is, '*do we need CAPTCHAS?*'. The prime purpose of CAPTCHA is to protect against denial of service (DoS) attacks, which is the responsibility of the service provider, and then why the user should bear the burden to deal with the CAPTCHA

especially when they cause deviation from the users' primary task. Likewise, the majority of the work on usable security has been on the operational and tactical level and therefore, has a cosmetic effect on the usable security problem. However, what is required in this regard are the long term and strategic solutions, for example, a requirement-engineering framework for aligning security and usability during the phases of the system development lifecycle (SDLC).

Moreover, one aspect on which there is a consensus among different groups working on usable security is to focus on learning and assisting the developers in handling the security and usability conflicts. This forms the primary research question addressed in this paper, which is '*how to assist security and usability developers in handling the conflicts and identifying suitable trade-offs while enabling learning in a specific context of use*?'. This research advocates the concept of '*usable security by design*', which is aimed at assisting the developers in handling the conflicts and identifying suitable trade-offs by using design patterns. Each design pattern solves a recurring design problem in a particular context of use. Using the patterns' approach can be advantageous not only for the developers but for the organizations as well. Software development organizations can also contribute to the catalog of patterns, based on previous experiences from the projects. Furthermore, using the patterns while ensuring effective management of the trade-offs does not affect the timely completion and costs associated with the project.

There are some existing usable security design patterns, but there is a need to collect those patterns, add them to a catalog and disseminate the catalog among the developers and designers. Furthermore, it is imperative to identify more patterns to be added to the catalog. For identifying more usable security patterns, the proposal for a three-stage methodology is presented in this paper. The remainder of the paper is organized as follows. Section 2 presents the background and literature review. Section 3 presents the proposed methodology for the identification of usable security patterns from existing implementations. Section 4 presents a case study to instantiate the proposed methodology. Section 5 presents the discussion and avenues for future investigation identified after the workshop, and Sect. 6 concludes the paper.

2 Background and Literature Review

In line with the research question addressed in this paper, the literature review was conducted considering the following objectives.

1. To rationalize the use of patterns as a way of assisting developers in handling interdisciplinary conflicts e.g. security and usability conflicts.
2. To identify existing usable security patterns (if any) and methodologies for identification for such patterns.

The authors [9] state, "insufficient communication with users produces a lack of user-centered design in security mechanisms". Both usability and security professionals recognize the importance of incorporating their concerns throughout the design cycle and acknowledge the need for an iterative rather than a linear design process. The use of patterns allows the concerns from both security and usability viewpoints to be incorporated right from the start of system development lifecycle. Patterns' ability to be improved over time and incorporate multiple viewpoints make them suitable for interdisciplinary fields like usable security [1]. Handling the security and usability concerns earlier in the development lifecycle helps in saving significant costs and delays associated with re-work.

An architect Christopher Alexander in the book 'A Pattern Language' originally introduced the concept of patterns [10]. Deriving inspiration from this, the same concept was implemented in computer science particularly in software engineering to assist the designers of the system, while providing guidelines and high-level principles. A similar concept was introduced in HCI to assist the development of user interface design (e.g. [11, 12]).

Each pattern expresses a relation between three things, *context*, *problem,* and *solution*. Patterns provide real solutions, not abstract principles, by explicitly mentioning the context and problem and summarizing the rationale for their effectiveness. Since the patterns provide a generic "core" solution, its use can vary from one implementation to another.

Furthermore, the patterns have three dimensions: descriptive, normative, and communicative [17]. From the perspective of usable security, the communicative dimensions of the patterns enable different communities to discuss design issues and solutions. Patterns also prove effective in the domains, which lack an existing body of knowledge; in such cases, the patterns assist in identifying effective practices as they emerge and capture them as objects for discussion, scrutiny, and modification [17].

In line with the second objective of the literature review, it was identified that the authors [13] while listing 20 usable security patterns also presented the results after analysis of commonly used software browsers like Internet Explorer, Mozilla Firefox and email clients like Microsoft Outlook. It was revealed that the identified patterns had a 61.67% application in the analyzed software implementations. The authors state "patterns make sense and can be useful guide for software developers". However, the work was limited to listing the patterns and justifying their usage.

The authors [14] presented a list of patterns to align security and usability. They classified the patterns into two categories: data sanitization patterns and secure messaging patterns. Different patterns listed include, 'explicit user audit', 'complete delete', 'create keys when needed', among others.

The authors [15] proposed a set of user interface design patterns for designing information security feedback based on elements of user interface design. Furthermore, the authors created prototypes incorporating the user interface patterns in the security feedback to conduct a laboratory study. The results of the study showed that incorporating the elements of usability interface design patterns could help in making security feedbacks more meaningful and effective.

The authors [1] presented a methodology for deriving usable security patterns during the requirements engineering stage of system development. The methodology relies on handling the conflicts during the early stages of system development and documenting the suitable trade-offs in the form of design patterns for reuse. What distinguishes the methodology presented in this paper from the work [1] is that the methodology discussed in this paper focuses on identifying and documenting instances of good implementations by experienced developers in the form of design patterns. This is more of a bottom-up approach involving the identification of the patterns from existing implementations. However, the work [1] focuses on the creation of new patterns based on system requirements where possible trade-offs are identified and managed. The managed trade-offs are documented as patterns for implementation in the specific project and reuse by other developers.

3 Methodology for the Identification of Usable Security Patterns

In this section, the proposed three-stage methodology for the identification of usable security patterns is presented. As stated earlier, the methodology is based on identifying new patterns from existing implementations, which are setting good practices in the industry (*see* Fig. 1). This methodology provides uniform means to identify new patterns, and an opportunity for various stakeholders to contribute towards identification of the patterns and building the usable security patterns catalog. Particularly, from the industrial perspective, it can enable documenting new patterns from the implementations by experienced developers, thereby facilitating the learning and training of new developers.

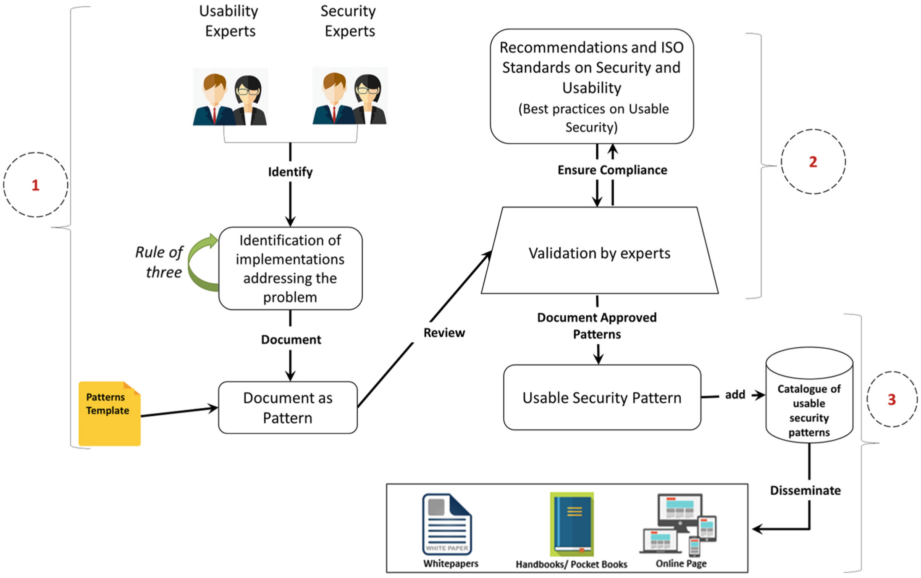


Fig. 1. The proposed methodology for identification of usable security patterns

- **Stage-1**: The first stage involves the selection of a common usable security problem. For the selection of a usable security problem, the experts can utilize one of the instruments such as surveys involving end-users, cognitive walkthroughs, heuristic evaluations, to mention a few. The next step is to identify existing implementations addressing the problem. Since the implementations can have different ways of approaching the problem, therefore, to document the best implementation as a pattern it is imperative to fulfill the '*Rule of Three*'. The rule of three requires at least three instances of similar implementations before a pattern could be identified and documented [17]. Once three instances of similar implementations for a particular problem are identified, the pattern is documented on a standard template. The details of usable security patterns' template are presented elsewhere [16]. Furthermore, selection of the best implementation is mainly based on the expertise of the professionals who are identifying it, however, to formalize the process it can also include evaluating the implementation with respect to a pre-defined set of heuristics. Defining such a set of heuristics for the evaluation would be considered as a part of the future work.

- **Stage 2**: The second stage involves a review of the newly documented pattern by one or more experts in the field. This stage involves activities like selection of expert(s) and gathering the reviews. Based on reviews the pattern is either accepted, which means it is ready to be finalized (*Stage 3*), or require modification, which means it goes back for modification to the experts who identified it during Stage 1, and in other cases, it may be rejected, which means it is discarded. The review by experts besides validation of the pattern has two advantages, (1) ensuring compliance with the underlying standards and best practices concerning security and usability, and (2) ensuring that the solution proposed in the pattern manages the trade-off effectively. The expert(s) review concerning each pattern is recorded on a checklist (see Table 1).

Table 1. Usable security pattern review checklist

<table>
<tr><td colspan="12">Usable security pattern review checklist</td></tr>
<tr><td colspan="12">Description: For the pattern under consideration fill in the columns below. Accessing ISO standards on security and usability is highly recommended to ensure compliance</td></tr>
<tr><td>Name of the pattern</td><td colspan="2">Relevant to usable security</td><td colspan="3">Effectively manages the trade-off</td><td colspan="3">Compliance with the standards an best practices</td><td>Decision</td><td>Additional recommendations</td></tr>
<tr><td>*/*Unique name of the pattern */*</td><td><b>Y</b></td><td><b>N</b></td><td><b>Y</b></td><td><b>N</b></td><td><b>Y/N</b></td><td><b>Y</b></td><td><b>N</b></td><td><b>Y/N</b></td><td>□ Accept
□ Modify
□ Reject</td><td>Include recommendations for improvement of pattern, proposal for modification, compliance to the standard, reasons for rejection, etc.</td></tr>
</table>

- **Stage 3**: This stage comprises the following activities subject to the decision by the expert(s):
  - *Accept:* The accepted patterns are added to the catalog. The patterns in the catalog can be disseminated among the community of developers and designers. The ways

of disseminating the patterns include online pages, pocketbook for developers, and whitepapers.

- *Modify:* The documented pattern is referred back to the security and usability experts who identified it. The proposal for modification is considered and after necessary amendments, the pattern is subjected to review for the second time.
- *Reject:* The rejected patterns are discarded; however, the recommendations are considered for compliance in the other identified patterns with similar as well as the other context of use.

4 Instantiating the Methodology: A Case Study

To instantiate the methodology and identify a usable security pattern from existing implementations, a case study was conducted. The participants in the case study were the members of the software engineering laboratory at LUT University. Participation in the case study was voluntary. The objective behind the case study was to identify instances of good implementations by experienced developers, which set best practices in the field concerning the problem described below.

Case Description:
Mobile devices, particularly smartphones and tablets have become an inseparable companion for human users, as they have a wide range of features not just limited to communication. With such increased usage, we have seen an increase in cases of loss/theft of mobile devices, which ultimately leads to data breaches.

Consider a scenario when someone's smartphone is lost. Even if the lost smartphone it was locked, the victim would still be worried about ways in which an adversary could bypass the authentication mechanism and get access to the device. Access to the device could mean a breach of privacy and identity (if payment options were linked to the lost device). The authors [18] report a user study revealing that 50% of the respondents did not feel protected in case of loss/theft of their smartphone. Based on the scenario, the following problem statement was formulated.

Problem Statement:
In case of loss/theft of the users' device, the data on the device increases the impact of loss in the form of breach of privacy. The user needs to have trust and protection feelings to be able to use the device for personal/work purposes.

Stages of Case Study:

- **Stage 1**: This first stage involved the selection of the usable security problem. The results of a survey [18] led to the selection of the problem. While identifying the implementations addressing this problem, a solution 'remote data deletion' was identified. The next step involved the application of the 'rule of three'. Once three similar implementations addressing the problem were identified, the pattern (presented in Fig. 2) was documented on the standardized template. The solution offered by the pattern for the problem stated above is to "Offer the user with remote deletion functionality hosted by the mobile vendor or mobile service provider via a usable secure interface".

A secure service available online will work in this regard. It should offer the remote deletion by invoking the restore factory settings procedure, which would erase all the information from the device in case of loss/theft. This procedure not only ensures the security of data but also incorporates the human aspect of security, achieving human satisfaction and trust (elements of the global usability).

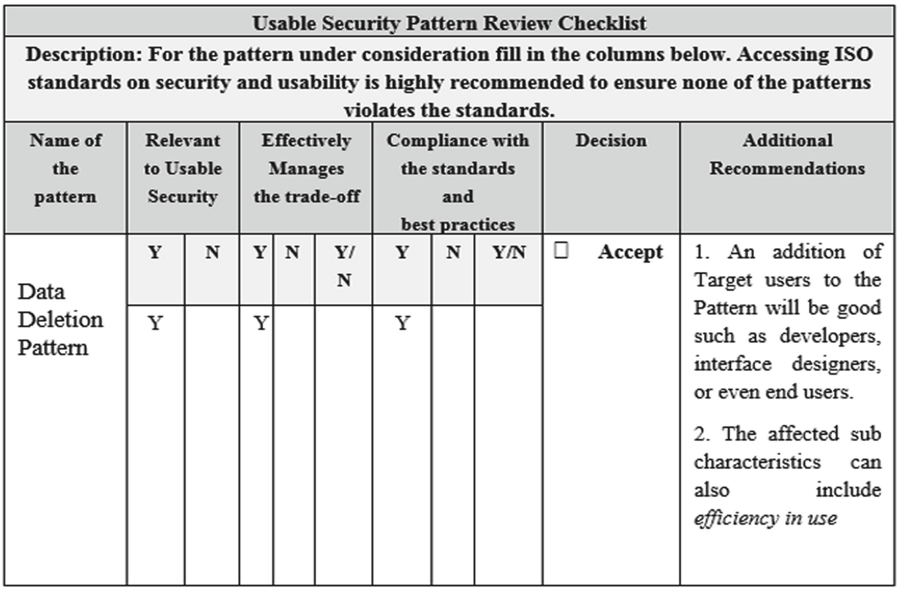
- **Title:** Data Deletion Pattern
- **Classification:** Data Protection, Device protection
- **Prologue:** To reduce the impact of loss in case of loss/theft of a device carrying sensitive personal/business information.
- **Problem statement**: In case of loss/theft of the users' device, the data on the device increases the impact of loss in the form of breach of privacy. The user needs to have trust and protection feelings to be able to use the device for personal/work purposes.
- **Context of Use:** Whenever there is loss/theft of device carrying user's data, which can lead to a breach of data.
- **Affected Sub Characteristics:** The sub- characteristics of usability and security being affected/involved when this pattern is applied.
    - Usability: satisfaction, trust, *efficiency in use*
    - Security: privacy, confidentiality, integrity
- **Solution:** Offer the User with remote deletion functionality hosted by the mobile vendor or mobile service provider via a usable secure interface.
- **Discussion:** Even if the lost smartphone was locked, the human user can still be bothered by breach of their privacy and the device's security. However, when the data has been removed from the device, the impact of loss can be minimized to an exclusively monetary loss.
- **Type of service**: Mobile devices or similar used in the same context.
- ***Target Users****: developers, designers*
- **Epilogue:** Improved data protection and reduced impact of loss.
- **Related Patterns:** Can be added later from the catalog.

Fig. 2. Data deletion pattern

Implementations of this pattern are available in the form of a "remote data deletion" functionality made available by smartphone manufacturers like Samsung and Apple for their users. Now the question arises who will use this pattern when this feature is already implemented? One scenario for the application of this pattern is in the case of other mobile devices including PDAs for inventory records, GPS, etc.

It is imperative to state that as per classification, the pattern (presented in Fig. 2) is a data/device protection usable security pattern. Other classifications of usable security patterns include usable authentication, usable security interface, among others. For example, a usable security interface pattern is presented in [16].

- **Stage 2**: This stage involved the validation of the pattern by the experts. It is pertinent to state that the pattern presented in Fig. 2 is a validated version of the pattern after reviewing by the experts. The items in *italic* were added based on experts' recommendations. The pattern review checklist from one of the experts is presented in Fig. 3.



Usable Security Pattern Review Checklist

Description: For the pattern under consideration fill in the columns below. Accessing ISO standards on security and usability is highly recommended to ensure none of the patterns violates the standards.

Name of the pattern	Relevant to Usable Security		Effectively Manages the trade-off			Compliance with the standards and best practices			Decision	Additional Recommendations
Data Deletion Pattern	Y	N	Y	N	Y/N	Y	N	Y/N	□ Accept	1. An addition of Target users to the Pattern will be good such as developers, interface designers, or even end users. 2. The affected sub characteristics can also include *efficiency in use*
	Y		Y			Y				

Fig. 3. Data deletion pattern review checklist

- **Stage 3:** Involved addition of this pattern to the catalog we are maintaining for dissemination and reuse by other developers.

5 Discussion

The presentation of the methodology during the workshop generated a discussion from which we identified the following avenues for future consideration.

- *Evaluation of instruments for identification of the usable security problem*: There is a need to evaluate different instruments that can help the security and usability experts in identifying the usable security problems with efficacy. For example, some of these instruments include:

  - *Surveys*, involving end-users' feedback and qualitative assessment of the problems faced by users while using a security system or service.
  - *Heuristic Evaluations*, which are conducted by experts to identify usable security problems due to violations of usability heuristics and security policies.
  - *Cognitive Walkthrough*, the security and usability experts inspect the user interfaces of security systems and services by going through a set of tasks and evaluating its understandability, ease of use and learning from the perspective of the targeted population.
  - *Contextual Inquiry,* that consists of observing services and systems in use within the context of participants' daily activities and asking for explanations as interesting events arise (security problems, usability problems, comments from users)
  - *Semi-structured interviews,* online or on-site with the users of security systems and services. The interviews would be focused on specific usability problems arising from security implementations.
  - *Use of tools*, within a lab, the users can be recruited to use security systems and services in a controlled environment. The human system interactions can be recorded using specialized tools like *Morae* or Observer *XT*.

- *Adding quantitative aspects to the methodology*: One dimension that needs further investigation is the addition of a quantitative method in the selection of the best implementations while documenting patterns. As stated in Sect. 3, the methodology considers only the qualitative aspects (expertise of professionals) in the selection of best implementations, therefore, considering the quantitative aspects will support the security and usability experts in selecting the best implementations for identifying and documenting new patterns. A quantitative methodology would also require a set of metrics to assist the identification of best implementations, for example, NUC (number of user complaints) is one such metric that can help in determining the best implementations from the user perspective. The lesser is the NUC, the better is the implementation from the user point of view. However, there is a need to identify a set of these metrics and incorporate their values by assigning weights to come up with a final valuation of the implementations quantitively. This valuation can be used by experts in the selection of the best implementations addressing the usable security problem under consideration.
- *Assessing the across system properties perspective*: Bouzekri *et al.*, presented their work on "Characterizing Sets of Systems: Across-Systems Properties and their Representation" during IFIP WG 13.2 & WG 13.5 Workshop at INTERACT 2019, an interesting aspect to consider from the perspective of our work is the effect of within systems and across system properties on the identified patterns. Considering the across system properties perspective, an important question to address is, do we need different patterns addressing the same usable security problem but requiring different solutions due to the nature of the context in which these systems are deployed?
- *Formalizing the process of selection of experts for review*: To have a set of experts for validation of the identified patterns, the work presented by Larusdottir and Kyas during the workshop identifies a mechanism that can be incorporated for selecting the right set of people for performing a validation job. The authors presented their work

related to the selection of an agile team for a developing development task. However, learning from their approach can be useful in formalizing the process of selection of experts.

6 Conclusion

Inter-dependencies and trade-offs between security and usability need to be approached strategically. The three-stage methodology presented in this paper is an attempt in this regard. Efforts need to be put in to develop a framework within the scope of the system development life cycle (SDLC) for eliciting the conflicts between security and usability while identifying suitable trade-offs between the two. The use of patterns can also be influential in documenting the outcomes of employing such frameworks. Patterns can assist also assist in improved communication between various segments working on the project more precisely the security and usability teams.

Additionally, the use of patterns does not only assist the developers within the organizational setting but also free-lancers in assessing the usability of their security options and vice versa. Furthermore, one pattern only solves one problem in a particular context of usage; therefore, an entire catalog of usable security patterns is required just like the user interface patterns catalog. The development of such a catalog is a time-consuming process and requires community-level efforts, therefore, we intend to present our proposal of using patterns and the methodology for identifying patterns to participants of the Human-Centered Software Engineering and HCI community for their feedback and participation in the development of the usable security patterns catalog.

Acknowledgment. The first author wishes to thank Professor Ahmed Seffah for his feedback during the initial phases of this research.

References

1. Naqvi, B., Seffah, A.: A methodology for aligning usability and security in systems and services. In: 2018 3rd International Conference on Information Systems Engineering, pp. 61–66 (2018)
2. Garfinkel, S., Lipford, H.R.: Usable Security History, Themes, and Challenges. Morgan and Claypool, New York (2014)
3. Whitten, A., Tygar, J.D.: Usability of security: a case study. School of Computing Science, Carnegie Mellon University. Rep. Technical Report CMU-CS-98-155 (1998)
4. Caputo, D.D., et al.: Barriers to usable security? Three organizational case studies. IEEE Secur. Priv. **14**(5), 22–32 (2016)
5. Garg, H., Choudhury, T., Kumar, P., Sabitha, S.: Comparison between significance of usability and security in HCI. In: 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT), pp. 1–4 (2017)
6. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: how much usability can you sacrifice for security? IEEE Secur. Priv. **15**(3), 24–29 (2017)
7. Sasse, M.A., Smith, M., Herley, C., Lipford, H., Vaniea, K.: Debunking security-usability tradeoff myths. IEEE Secur. Priv. **14**(5), 33–39 (2016)

8. Cranor, L.F., Buchler, N.: Better together: usability and security go hand in hand. IEEE Secur. Priv. **12**(6), 89–93 (2014)
9. Cranor, L., Garfinkel, S.: Security and Usability. O'Reilly Media, Inc., Sebastopol (2005)
10. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. Oxford University Press, Oxford (1977)
11. Tidwell, J.: Designing Interfaces. O'Reilly Media, Inc., Sebastopol (2005)
12. Welie, M.V.: Patterns in interaction design (2008). https://www.welie.com/patterns/
13. Ferreira, A., Rusu, C., Roncagliolo, S.: Usability and security patterns. In: 2nd International Conference on Advances in Computer-Human Interaction, pp. 301–305 (2009)
14. Garfinkel, S., Miller, R.C.: Patterns for aligning security and usability. In: Symposium on Usable Privacy and Security (SOUPS) (2005). https://cups.cs.cmu.edu/soups/2005/2005posters/13-garfinkel.pdf
15. Munoz-Arega, J., et al.: A methodology for designing information security feedback based on user interact patterns. Adv. Eng. Softw. **40**(2009), 1231–1241 (2009)
16. Naqvi, B., Seffah, A.: Interdependencies, conflicts and trade-offs between security and usability: why and how should we engineer them? In: Moallem, A. (ed.) HCII 2019. LNCS, vol. 11594, pp. 314–324. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22351-9_21
17. Mor, Y., Winters, N., Warburton, S.: Participatory patterns workshops resource kit. Version 2.1 (2010). https://hal.archives-ouvertes.fr/hal-00593108/document
18. Sophos: Security threat report (2010). https://www.sophos.com/sophos/docs/eng/papers/sophos-security-threat-report-jan-2010-wpna.pdf
19. Kirlappos, I., Sasse, M.A.: What usable security really means: trusting and engaging users. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2014. LNCS, vol. 8533, pp. 69–78. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07620-1_7

Publication IV

Naqvi, B., Porras, J.
Usable Security by Design: A Pattern Approach

In: *2nd International Conference on HCI for Cybersecurity, Privacy and Trust, Held as part of 22nd International Conference on Human Computer Interaction (HCII), 2020*

Reprinted with permission from
Lecture Notes in Computer Science
LNCS 12210, pp. 609-618, 2020

Usable Security by Design: A Pattern Approach

Bilal Naqvi[1,2](✉) and Jari Porras[1]

[1] LUT Software, LENS, LUT University, 53850 Lappeenranta, Finland
syed.naqvi@student.lut.fi
[2] Software Engineering, Mirpur University of Science and Technology, MUST, Mirpur, Pakistan

Abstract. Security and usability are often in conflict. There is a recognition that security cannot be achieved in real sense unless it incorporates the human factor (usability elements). Despite this recognition, the state of the art identifies many challenges and reasons for conflicts between security and usability. This paper discusses some of these challenges while proposing the use of design patterns to handle those challenges. While justifying the use of patterns as one of the effective ways of handling the problem (conflicts), the paper presents a proposal for participatory usable security design patterns workshop. The workshop provides a forum for discussing a variety of issues concerning the usability and security conflicts while documenting the instances of conflicts and suitable tradeoffs as design patterns for use by other designers and developers. A catalog of usable security design patterns can assist the system designers and developers by positively influencing their decision-making abilities when it comes to conflicts.

Keywords: Patterns · Security · Usability · Usable security

1 Introduction

Security and usability are essential quality characteristics in today's software systems. To address the quality demands, security and usability are considered in specialized teams where the focus of each team is specific, the security team focuses on making the system security as robust as possible against internal and external attacks, however, usability is a minor concern for them. Whereas the usability team focuses on improving usability issues arising with the use of the system while providing a positive user experience (UX). With this specific focus, the need for usable security is realized when the instances of conflicts between security and usability are identified. A classic example in this regard is the password for authentication. The security dimension suggests that the passwords should be sufficiently long, frequently changed, have different cases and special characters, etc. However, from the user's (usability) point of view, such passwords are hard to memorize. If the suggested security guidelines are implemented, they have an adverse impact on the usability of the system, and if not implemented the system security is at stake.

Recently, there has been a realization that security cannot be implemented effectively unless we pay attention to the usability aspects [1]. US National Institute of Standards and Technology (NIST) report NIST Special Publication 800-63B states "evaluating the

A. Moallem (Ed.): HCII 2020, LNCS 12210, pp. 609–618, 2020.
https://doi.org/10.1007/978-3-030-50309-3_41

usability of authentication is critical, as poor usability often results in coping mechanisms and unintended workaround that can ultimately degrade the effectiveness of security controls" [4]. Initially, usable security was considered as limited to the usability of security interfaces, however, with time aspects like, (1) correspondence between systems' internal procedures and user's thoughts, (2) incorporating user values into security design [2, 3], were identified as important aspect to be considered in development of simultaneously usable and secure systems. With correspondence between the system's internal procedures and human thoughts, it is meant that there should be compliance between user perceptions and the way security procedures are performed on the system. Such compliance could be achieved in two ways, (1) training the users, and, (2) designing the security systems while considering the human aspects, thereby decreasing the chances of human errors as the system works the same way as the user thinks it does.

Similarly, incorporating user values into security design can also contribute towards implementing security effectively. In the development of security systems, the goals are set by experts who are unaware that users might have different priorities and values concerning security [3]. Certain user value-based objectives associated with security include objectives such as minimize system interruptions, maximize information retrieval, maximize ease of use, enhance system-related communication, etc. [2]. Therefore, the elements of value-sensitive design (VSD) can improve users' engagement with security.

Despite the realization of aligning security and usability in the development of systems and services, the state of the art concerning usable security identifies many challenges. While considering all the challenges identified via literature review and conducting exploratory studies in the industry, this paper advocates the concept of '*usable security by design*'. The usable security by design concept is aimed at aligning security and usability right from the start of the system development lifecycle [5]. The concept is centered on the development of a catalog of usable security design patterns to assist the system designers and developers in dealing with the conflicts, thus delivering simultaneously secure and usable solutions. The fundamental question addressed in this paper is '*how do we develop a catalog of usable security patterns?'*. The paper presents a proposal for a participatory usable security design patterns workshop [6]. To conduct such a workshop, various templates to be used during the workshop are also presented.

The remainder of the paper is structured as follows. Section 2 presents the background. Section 3 presents the proposal for a participatory usable security design patterns workshop. Section 4 presents the related work and Sect. 5 concludes the paper.

2 Background

2.1 Challenges in the State of Art

The authors [8] state that "usable security assumes that when security functions are more usable, people are more likely to use them, leading to an improvement in overall security. Existing software design and engineering processes provide little guidance for leveraging this in the development of applications". Based on an analysis of existing literature and exploratory studies in the industry, the following are some of the challenges in aligning security and usability during the system development lifecycle.

- *Security and usability handled independently*: Security and usability are considered by different teams, where the focus of each team is specific i.e. the team working on security is focused on making the system secure; whereas the team focusing on usability and UX is focused on improving the human interaction with the system. There does not exist a mechanism where concerns from both teams can be integrated towards achieving the goal of simultaneously usable secure systems, therefore it is a tradeoff between security and usability.
- *Reliant on Skill of Developers*: Handling usable security in an organizational setting is reliant on the skill of developers [8]. Developers are either experts in security or usability. Despite this, there does not exist a mechanism (in practice) to assist developers in handling the issues where security and usability are in conflict.
- *Lack of emphasis during the early phases of development*: Security requirements are usually improperly specified, due to lack of emphasis on security during the early stages of development; the same holds for usable security [9]. The authors [10] argue that system security is usually considered in the production environment by employing protections like firewalls, IDS/IPS, AV servers, etc., which identifies the state of consideration on security during the system development phases, let alone its usability.
- *Existence of suitable technique for assessing adequacy*: Concerning the adequacy of security, techniques like vulnerability scan and penetration testing can be employed to check the robustness of security features, however, there is no such technique for evaluating the adequacy of usable security [16].
- *Constraint to a Constraint*: The requirement engineering community defines security as a constraint to the system's functional requirements [11]. The question is, if security is a constraint to the system's requirements, then usability of security could be a constraint to a constraint, which is one of the reasons that usable security requirements are neither specified nor addressed adequately.

The challenges discussed above often serve as contributing factors to the complexity of usable security problem. Furthermore, the standards concerning software quality in general and usability, security in particular, do not provide any guidance when these characteristics are in conflict. While considering all these aspects, this paper advocates the use of design patterns for handling security and usability conflicts.

2.2 Why Patterns?

A pattern expresses a relationship between three things, *context*, *problem,* and *solution.* Furthermore, the patterns have three dimensions: descriptive, normative, and communicative [6]. In its *descriptive* dimension, a pattern is an analytic form to describe problems, context and solutions. However, in the *normative* dimension, a pattern is a meta-design tool to identify key issues and propose a method for addressing them. It is a *communicative* tool to allow different communities to discuss and address issues [6].

Moreover, for multidisciplinary fields such as usable security, it is important to consider the concerns from both perspectives. Patterns can incorporate multiple concerns due to their descriptive nature and enable different communities to discuss design issues and solutions due to their communicative ability. Patterns' ability to evolve with time

Table 1. Challenges in the state of art with a description of how pattern addresses it

Challenges	Description	Involved patterns' dimension
Usability and security handled independently	Patterns allow concerns from both usability and security to incorporated before documenting a final solution	*Communicative*
Reliant on skill of developers	Information provided by the pattern including problem addressed, solution, context facilitates the developers in making reasonably accurate decisions in other similar contexts	*Descriptive*
Lack of emphasis during the early phases of development	Patterns can be incorporated right from the beginning of development life cycle and can be used by designers and developers as a meta-design tool for identification of key problems and solution for resolving them	*Descriptive/communicative*
Existence of suitable technique for accessing adequacy	Patterns ability to be improved with time helps in establishing adequacy of the solution presented by the pattern. Even when the patterns are disseminated, they are monitored and reviewed and proposal for amendments can be incorporated at any stage	*Descriptive/normative*
Constraint to a constraint	Security and usability are considered together thereby decreasing the chances of being considered as constraint to constraint or as after thoughts	*Normative*

also makes them suitable for problems like usable security. A pattern has different states, a *proto pattern* is a pattern which is newly documented after the first iteration, and it captures the basic elements of problem, context, and solution. However, after undergoing various refinement stages it is in *alpha-state,* ready to be released for use and testing by designers and developers.

Furthermore, patterns provide benefits like means of common vocabulary, shared documentation, improved communication among the different stakeholders during product development [5]. Patterns provide real solutions by explicitly mentioning the context and problem and summarizing the rationale for their effectiveness. Since the patterns provide a generic "core" solution, its use can vary from one implementation to another [7]. All the challenges in the state of art identified earlier along with how the pattern approach helps in addressing them are presented in Table 1. The Table 1 also presents the dimension of the pattern involved in addressing a particular challenge.

As stated earlier, security and usability have evolved independently as different domains, therefore, expertise in both security and usability is hard to find in one person. Todays' industrial practices reflect that handling the security and usability conflicts is reliant on the skill of the developers [8]. The use of patterns provides a way of assisting developers at work by influencing their decision-making abilities when it comes to the conflicts between security and usability. Moreover, the patterns can be incorporated right from the start of the systems' development lifecycle, which helps in saving significant costs and effort associated with rework in contrast to the cases where security and usability are afterthoughts.

3 Proposal for Participatory Usable Security Design Patterns Workshop

Having discussed the problem and motivation for using design patterns, the question is how we can identify such patterns to be able to build a catalog of patterns for dissemination among common developers. One mechanism for creating such a catalog is a participatory usable security design patterns workshop. The activities during the workshop are to be performed in groups (3–5 participants each). Participants of the workshop are security and usability developers and designers. The key activities during the workshop are presented in Fig. 1, which include:

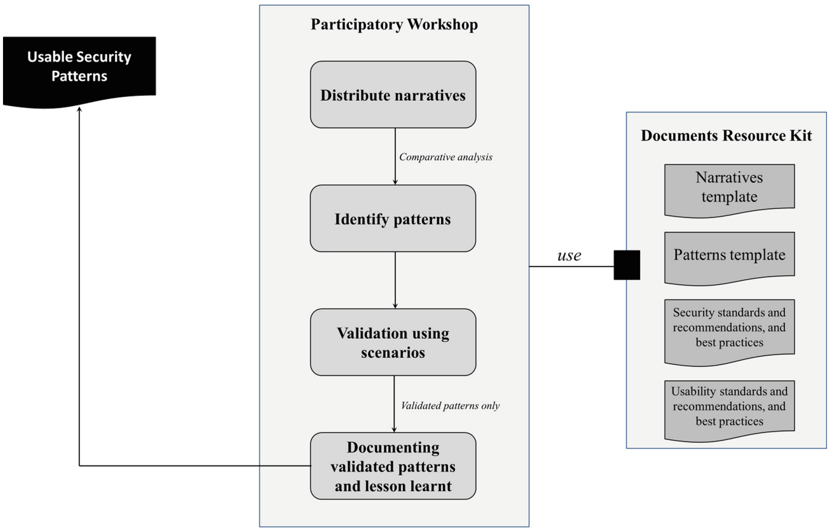
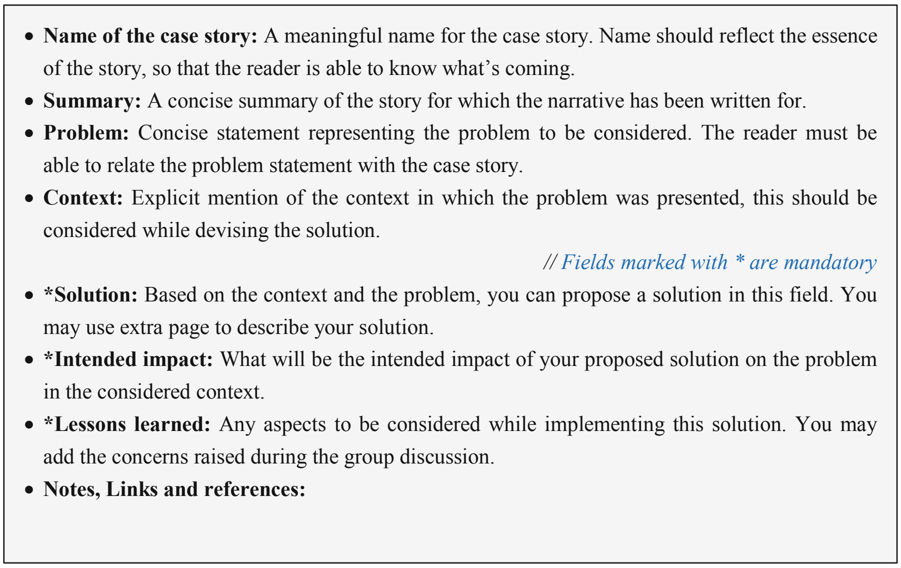

Usable Security Patterns
Participatory Workshop
Distribute narratives
Comparative analysis
Identify patterns
Validation using scenarios
Validated patterns only
Documenting validated patterns and lesson learnt
use
Documents Resource Kit
Narratives template
Patterns template
Security standards and recommendations, and best practices
Usability standards and recommendations, and best practices

Fig. 1. Proposal for participatory usable security design patterns workshop

1. **Distribute narratives**: The narratives describing a usable security problem are distributed among the participants in groups. The narrative elicits a case story describing a usable security problem. The groups are tasked to design a solution of their own for the problem under consideration. The narrative template used during the first activity of the workshop is presented in Fig. 2.



- **Name of the case story:** A meaningful name for the case story. Name should reflect the essence of the story, so that the reader is able to know what's coming.
- **Summary:** A concise summary of the story for which the narrative has been written for.
- **Problem:** Concise statement representing the problem to be considered. The reader must be able to relate the problem statement with the case story.
- **Context:** Explicit mention of the context in which the problem was presented, this should be considered while devising the solution.

*// Fields marked with * are mandatory*

- ***Solution:** Based on the context and the problem, you can propose a solution in this field. You may use extra page to describe your solution.
- ***Intended impact:** What will be the intended impact of your proposed solution on the problem in the considered context.
- ***Lessons learned:** Any aspects to be considered while implementing this solution. You may add the concerns raised during the group discussion.
- **Notes, Links and references:**

Fig. 2. The narrative template to be used during the workshop

2. **Identify patterns**: The solutions from each group are subjected to comparative analysis in an attempt to identify instances of good design. The 'Rule of Three' also comes into play here. The rule of three requires at least three instances of similar implementations before a pattern could be identified and documented [6]. Once three instances of similar implementations for a problem are identified, the pattern is documented on a standard template.
3. **Validation using scenarios**: The participants are provided with a list of design patterns (already identified) and a problem scenario. The problem scenario being used during this stage involves a set of problems, and the task involves the selection of the patterns (from the list) that are applicable in the context being considered. The participants are tasked to document the description of a solution derived by applying a pattern in the considered context. If the right pattern is applied in the right context, it is validated; otherwise, it is subjected to a modification to ensure the use of the right patterns in the right scenarios.
4. **Documenting validated patterns and lessons learned**: In the end, the lessons learned and recommendations for future use of patterns are documented. The outcome of the activity is a catalog of validated usable security design patterns, which will be disseminated among the community of designers and developers to positively influence their decision-making abilities when it comes to conflicts.

An example of how a usable security pattern looks like is presented in Fig. 3. It is imperative to state that the pattern is documented on a standard template.

- **Title:** Toggle Password Visibility
- **Classification:** Authentication
- **Prologue:** To ensure secure authentication and users' privacy while preserving the usability element of feedback.
- **Problem statement**: Password for authentication is masked by default to protect against attacks such as shoulder surfing. This is done to preserve breach of privacy and authentication, but at the cost of 'feedback'. If the user makes an error while typing the long password s/he has to retype the entire password without just knowing and correcting the error.
- **Context of Use:** Whenever the password is masked to protect against shoulder surfing and other similar attacks.
- **Affected Sub Characteristics:** The sub characteristics of usability and security being affected/involved when this pattern is applied.
    - Usability: satisfaction, effectiveness in use, desirability
    - Security: privacy, confidentiality, authentication
- **Solution:** Provide the user with option to toggle password visibility by providing an icon or button. The button/icon should unmask the users' password. The password should remain unmasked until the button/icon is being clicked. The button/icon should be accessible with the mouse pointer.
- **Discussion:** This solution enhances the usability element of feedback while preserving users' privacy and security of the authentication process. The button/icon can be presented at the far end of password field or below it. This would help users in correcting the mistyped character in the password rather than retyping the entire password.
- **Type of service**: Desktop/ Web application requiring authentication with passwords.
- **Epilogue:** Increased user satisfaction, desirability of the service while providing the effectiveness in use.
- **Related Patterns:** To be added from the catalog

Fig. 3. Toggle password visibility pattern

The pattern presented in Fig. 3. addresses the conflict between authentication (security mechanism) and feedback (usability element) in cases where the user is confident that the password is not readable by the adversary. There are instances of this pattern on the authentication screens by major service providers, however, it is documented and intended for other designers and developers for consideration in newer versions of the system they develop. Moreover, other usable security patterns are available elsewhere [5, 7, 13, 15].

4 Related Work

The authors [7] presented a four staged framework for identification of conflicts and elicitation of suitable tradeoffs as patterns. In the first stage, the usable security problems are identified, which are modeled and quantified during the second stage. Standards and best practices on security and usability are accessed while developing suitable tradeoffs (solutions) to be documented as patterns. The documented patterns are applied to the software ecosystem.

Furthermore, the authors [15] presented a methodology for deriving usable security patterns during the requirements engineering stage of system development. The methodology is aimed at handling the conflict from the requirement engineering stage of system development. It does so by enumerating all security-related features. For all the enumerated features the security concerns are listed, and usability concerns arising from security features are identified. Once the concerns from both security and usability perspectives are known, the tradeoffs are explicitly elicited and then documented as patterns.

The authors [12], while listing 20 usable security patterns presented the results after analyzing applications such as Internet Explorer, Mozilla Firefox, and Microsoft Outlook. The authors state "patterns make sense and can be useful guide for software developers". However, the work was limited to listing the patterns and justifying their usage.

The authors [13] presented a list of patterns to align security and usability. They classified the patterns into two categories: data sanitization patterns and secure messaging patterns. Different patterns listed include, 'explicit user audit', 'complete delete', 'create keys when needed', among others.

The authors [14] proposed a set of user interface design patterns for designing information security feedback based on elements of user interface design. In addition, the authors created prototypes incorporating the user interface patterns in the security feedback to conduct a laboratory study. The results of the study showed that incorporating the elements of usability interface design patterns could help in making security feedbacks more meaningful and effective.

What distinguishes this work from others just discussed is that it provides a mechanism to involve a wider group of developers and designers during a workshop and identifying patterns based on their expertise. Though the work [7, 15] provides an avenue for identifying patterns, their scope and intended environment of application is during a project or in a team. However, the current proposal has been designed to hold good for participants from multiple projects and teams. We believe that the workshop proposal discussed in this paper can help attract a wider audience and identify usable security patterns.

5 Conclusion

There is a recognition that security and usability need to be handled together and integrated during the entire system development life cycle, rather than being considered as afterthoughts. With reference to the literature, we identified various challenges in the state of the art concerning usable security and proposed the use of design patterns as a way to handle the usable security challenge. While justifying the use of patterns in handling the usable security problem, we presented the proposal for a workshop for identifying and developing a catalog of usable security patterns. The catalog of patterns can help common security and usability designers and developers by influencing their decision-making abilities when it comes to security and usability conflicts in other but similar contexts.

Developing such catalogs requires a community-level effort and arranging various participatory workshops. We hope to gather the attention of the HCI community during the conference towards establishing a joint effort framework for arranging such workshops and collecting more usable security design patterns.

Moreover, the research advocates the shift in approach from 'user is the weakest link in security chain' to achieving, (1) correspondence between systems' internal procedures and human thoughts, and, (2) incorporating user values into security design. As an instance of the patterns' approach, a usable security pattern was also presented in the paper.

References

1. Garfinkel, S., Lipford, H.R.: Usable Security History, Themes and Challenges. Morgan and Claypool, San Rafael (2014)
2. Dhillon, G., Oliveira, T., Susarapu, S., Caldeira, M.: Deciding between information security and usability: developing value-based objectives. Comput. Hum. Behav. **61**, 656–666 (2016 2016
3. Dodier-Lazaro, S., Sasse, M.A., Abu-Salma, R., Becker, I.: From paternalistic to user-centered security: putting users first with value-sensitive design. In: CHI 2017 Workshop on Values in Computing, p. 7 (2017)
4. Grassi, P.A., Newton, E.M., Perlner, R.A., et al.: Digital identity guidelines: authentication and lifecycle management. Special Publication (NIST SP)-800-63B (2017). https://doi.org/10.6028/NIST.SP.800-63b
5. Naqvi, B., Porras, J., Oyedeji, S., Ullah, M.: Aligning security, usability, user experience: a pattern approach. In: IFIP Joint WG 13.2 & WG 13.5 International Workshop on Handling Security, Usability, User Experience and Reliability in User-Centered Development Processes held during International Conference on Human Computer Interaction (INTERACT) (2019)
6. Mor, Y., Winters, N., Warburton, S.: Participatory Patterns Workshops Resource Kit. Version 2.1 (2010). https://hal.archives-ouvertes.fr/hal-00593108/document
7. Naqvi, B., Seffah, A.: Interdependencies, conflicts and trade-offs between security and usability: why and how should we engineer them? In: 1st International Conference, HCI-CPT 2019 Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, pp. 314–324 (2019)
8. Caputo, D.D., Pfleeger, S.L., Sasse, M.A., Ammann, P., Offutt, J., Deng, L.: Barriers to usable security? Three organizational case studies. IEEE Secur. Priv. **14**(5), 22–32 (2016)
9. Riaz, M., Williams, L.: Security requirements patterns: understanding the science behind the art of pattern writing. In: Requirements Patterns (RePa 2012), pp. 29–34. IEEE (2012)
10. Parveen, N., Beg, R., Khan, M.H.: Integrating security and usability at requirement specification process. Int. J. Comput. Trends Technol. **10**(5), 236–240 (2014)
11. Xuan, X., Wang, Y., Li, S.: Privacy requirements patterns for mobile operating systems. In: IEEE 4th International Workshop on Requirements Patterns, pp. 39–42. IEEE (2014)
12. Ferreira, A., Rusu, C., Roncagliolo, S.: Usability and security patterns. In: Second International Conference on Advances in Computer-Human Interaction, pp. 301–305 (2009)
13. Cranor, L., Garfinkel, S.: Patterns for aligning security and usability. In: Symposium on Usable Privacy and Security (SOUPS), Poster (2005)
14. Munoz-Arega, J., et al.: A methodology for designing information security feedback based on user interact patterns. Adv. Eng. Softw. **40**(2009), 1231–1241 (2009)

15. Naqvi, B., Seffah, A.: A methodology for aligning usability and security in systems and services. In: 2018 Third International Conference on Information Systems Engineering, pp. 61–66 (2018)
16. Wang, Y., Rawal, B., Duan, Q., Zhang, P.: Usability and security go together: a case study on database. In: 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), pp. 49–54 (2017)

Publication V

Naqvi, B., Porras, J.
Interdependencies, Conflicts and Trade-offs between Security and Usability: Why and how should we Engineer Them?

In: *1st International Conference on HCI for Cybersecurity, Privacy and Trust, Held as part of 21st International Conference on Human Computer Interaction (HCII), 2019*

Reprinted with permission from
Lecture Notes in Computer Science
LNCS 11594, pp. 314-324, 2019

Interdependencies, Conflicts and Trade-Offs Between Security and Usability: Why and How Should We Engineer Them?

Bilal Naqvi[1,2](✉) and Ahmed Seffah[3]

[1] Software Engineering, LENS, LUT University, Lappeenranta, Finland
syed.naqvi@student.lut.fi
[2] Mirpur University of Science and Technology, MUST, Mirpur, Pakistan
[3] Green UX Design, Thinking Associates, Paris, France

Abstract. Security and usability are considered as conflicting goals. Despite the recognition that security and usability conflicts pose a serious challenge to achieve effective security, the review of the state of art identifies many gaps in today's practices including, (1) failure of security specialists to address usability, as perceived and defined by the human computer interaction (HCI) community, (2) industry's behavior is being more driven by bug fixing, rather than trying to examine and consider the context and the human experiences in which the bugs occurs, and (3) the lack of HCI skills required for conducting effective user studies. Furthermore, analysis of the existing literature identifies different perceptions concerning the relationship between security and usability. Some researchers have identified existence of trade-offs when it comes to the security and usability conflicts, however, others refer to the trade-offs as mere myths. A four staged process oriented framework to address the security and usability conflict is presented in this paper. The framework governs aspects from identification of the conflicts to elicitation of suitable trade-offs. To support re-use, the outcomes of employing the framework are documented in form of design patterns. A template to standardize documentation of the patterns is also presented along with one example of the usable security patterns.

Keywords: Usability · Security · Usable security · Conflicts · Trade-offs · Framework · Patterns · Usable security patterns

1 Introduction

ISO 25010 model lists security and usability among the eight characteristics of its product quality model [1]. Despite providing guidance on handling each quality characteristic individually, ISO 25010 does not provide guidance when two or more dependent characteristics come into conflict. An example of such a conflict is the conflict between security and usability. As an instance of security and usability conflict consider passwords; despite their role in implementing authentication (a security mechanism), passwords have a human dimension. The password security guidelines suggest passwords to be sufficiently long, frequently changed, have different cases and special characters, etc., however, from user's perspective such passwords are hard to

A. Moallem (Ed.): HCII 2019, LNCS 11594, pp. 314–324, 2019.
https://doi.org/10.1007/978-3-030-22351-9_21

memorize especially when re-use of the passwords is strongly discouraged and an average user has to manage around 22 online password [2].

Password masking is another instance of the security and usability conflict. To protect against shoulder surfing and other similar attacks, almost all authentication implementations mask the password when the user types it. However, for a legitimate user it impacts usability element of 'feedback' as in case of a mistake the user has to re-type long complex password, rather than knowing and correcting the mistake. Therefore, it can be gathered that password masking approach holds good from security perspective, but it has an impact from the usability point of view.

Human factors are perhaps considered as greatest barrier to effective computer security [3]. Most security mechanisms are too difficult and confusing for the average computer user to manage correctly. Furthermore, a common belief is that security and usability are two the opposed quality factors that are related to different components of a system (functionality and user interface respectively). This means that, security of the system and usability of the services can be engineered by two separate teams, mainly by software engineering and user interface (UI)/user experience (UX) teams. However, there are several cases in which security and usability are enhanced by modelling their mutual relationships. Typical examples include online payment and e-banking services, supervision of critical industrial infrastructures, crisis management. This research aims to bridge the gaps between security specialists and UI/UX experts. The following are the key gaps:

One gap explains the failure of security specialists to address usability, as perceived and defined by the HCI community. Security and usability have historically evolved independently or have been considered as two opposite factors. Another historical explanation is that researchers were more driven by technology rather than user problems and perceptions of security. For example, the development of identity management technologies was so demanding in terms of security that it left little time and costs to cater usability and the human factors in general.

A second gap that may be advocated is the industry's behaviors is more driven by bug fixing, rather than trying to examine and consider the context and the user experiences in which the bugs occurs. Therefore, most industry efforts have been on automating the process of reporting and handling bugs, rather than looking for human experiences and how they can promote more secure operations overall.

Another gap that demonstrates the lack of alignment between security and usability is the design and innovation approach leading to new security technologies. Most often, the innovation is initiated by a company developing an "in-house technology" addressing a specific problem which occurs in a specific project. Other groups in the same company or others companies may develop their own versions of these solutions. This makes it difficult to ensure the usability of these in-house solutions and several versions of them, while changing the original context of their applicability. Fire-walls, junk mail filters, spyware, and antivirus are good examples.

Finally, the lack of HCI skills required for conducting effective user studies are a serious obstacle. Moreover, user studies are difficult to conduct because regulations governing use of human subjects' in experiments related to safety and security of the systems and services have to be considered.

Despite these gaps and non-alignment between security and usability, the conflict between these two is a recognized problem; the primary question addressed in this paper is why and how to engineer the conflicts and trade-offs between security and usability. One approach that we consider appropriate for engineering the conflicts and appropriate trade-offs involves the use of design patterns. Patterns can be used to document instances of the conflict and balanced solution to address the conflict (right trade-off). Patterns can be disseminated among the community of security and usability developers to influence their decision making when it comes to the conflict between the two characteristics.

The remainder of this paper is organized as follows. Section 2 presents the literature review, which was conducted considering two main objectives. Section 3 discusses the primary question addressed in the paper i.e. why and how to engineer the conflicts and trade-offs between security and usability, both 'why' and 'how' to engineer conflicts and trade-offs are discussed in subsequent sub-sections. A template to standardize documentation of the patterns is also presented along with one example of the usable security patterns. Section 4 concludes the paper.

2 Literature Review

Despite the recognition of security and usability conflict as a challenge, not much has been accomplished for two reasons, (1) security and usability are considered as after thoughts, and (2) security and usability are not considered strategically, and not integrated into to the strategic plans for system development [4].

The literature review was conducted in two stages with objectives as follows.

1. To identify one of the core reasons for non-alignment between security and usability.
2. To identify solutions for addressing security and usability conflicts.

The result of the first stage of literature review revealed inconsistent perceptions about relationship between security and usability as one of the reasons for non-alignment between the two characteristics. However, the findings relevant to both objectives are presented in subsequent sub-sections.

2.1 Inconsistent Perceptions About Relationship Between Security and Usability

Various communities and interest groups have been studying the security and usability conflicts independently from each other, these include: (1) traditional computer security community dealing with the wider scope of quality of services in computer and communication technologies; usability is a minor concern addressed at a cosmetic level in this community, (2) the software engineering community where security and usability have been defined as two among the eight major quality characteristics, and usability is a characteristic of user interfaces and security is a characteristic of the functionality, (3) the HCI community, to name a few. As a result, the available literature on relationships between security and usability can be classified in two categories.

- There are trade-offs when it comes to conflicts between security and usability.
- Trade-offs between security and usability are mere myths.

Most of the research till date argues on existence of the trade-offs between security and usability. The authors [5] conducted a case study on iOS and Android to find an answer for “what is more important: usability or security”. The authors identified that importance of security and usability is purely situation based, and that the trade-offs are sometimes in favor of security and vice versa. The authors [6] presented an empirical evidence in favor of existence of the trade-offs between security and usability. The empirical study featured three different schemes for code voting systems. The authors state, “nevertheless, the security gains come at the cost of usability losses”. The authors [7] presented an empirical investigation concerning existence of trade-offs between security and usability. The results of within-subjects study to understand and value security and usability trade-offs in end-to-end email encryption were presented. The results of the study identify that the participants in their choice for the preferred system to use deliberately made the trade-offs between security and usability.

In parallel with the research establishing existence of the trade-offs, there is some research classifying security and usability trade-offs as mere myths. A special issue ‘the security-usability trade-off myth’ features one such discussion between researchers and practitioners in usable security [8]. The participants were of the view that decreasing usability can lead to less security and understanding the context in which solutions are deployed is important. The participants discussed the example of two-factor authentication involving one-time passwords (OTP) and its consequences if the length of OTP is increased from 6 to 8 characters. Overall, the participants were of the view that, “security experts simply invoke the myth of tradeoff between usability and security, and use this as cover to avoid the exercise of saying precisely what security benefit in precisely what scenarios this usability burden is going to deliver.” The authors [9] stated that security and usability are not inherently in conflict. The authors suggested that the researchers have to go beyond than just adopting human-centered design principles and consider involving the user in the decision making process.

2.2 Solutions to Address the Security and Usability Conflicts

In line with the second objective of the literature review, we present the solutions that have been proposed to address the security and usability conflict. The author [10] presented a set of guidelines to cater the security and usability conflict. The work is mainly focused on avoiding the conflict by depriving the user from making system security related decisions. The author presented guidelines like, providing a check-list to developers of security systems, hiding security related tasks from users, reducing the user memory load etc. The author also suggested that user should be involved in making security decisions on the system only when the situation is clear to the user; otherwise, the system should take the security decisions itself.

The authors [11] while studying the trade-offs between security and usability presented a set of guidelines to cater the conflict. The authors considered various aspects of usability such as effectiveness, satisfaction, efficiency, learnability and presented different guidelines focusing on each of the mentioned elements in conjunction with security.

The authors [12] suggested to implement security features as a separate service on cloud naming it CaaS “Confidentiality as a Service”, which would perform the confidentiality function on behalf of the users even if the credentials are lost. The main theme discussed in their work is to create a level of abstraction, and let the service perform security tasks on user’s behalf.

The authors [13] presented an ontological framework for catering the security and usability conflict. The framework is based on identification of usability/security requirements, identifying meaning and system context. After that the conflicts are identified on basis on system requirements, which are characterized on basis of their impact and listed. The nature of the identified conflict is then determined, and based on that the conflict resolution strategy is made in accordance with the system requirements.

The authors [14] presented an ‘Assessment Framework for Usable Security’ (AFUS), which works by filtering and merging the security and usability requirements, and then applying utility functions for risk analysis. The decision trees are generated to calculate the weight and utility of each attribute of security and usability. The weights determine relative importance of attributes to be considered for requirement specification of software. The authors claim that requirements specified after AFUS have a balance between usability, security and usable security.

3 Interdependencies, Conflicts and Tradeoffs Between Security and Usability: Why and How Should We Engineer Them?

3.1 Why Is It Important to Handle Security and Usability Conflicts?

Security cannot be achieved in real sense unless it incorporates the human element [22]. To establish why it is important to handle security and usability conflicts, we refer to some existing empirical evidences and technical reports. National Institute of Standards and Technology (NIST) report NISTIR 8080 states that “the human element is a critical yet often overlooked component during technology integration […], it is critical to understand users’ primary goals, the characteristics of the users (both physical and cognitive attributes), and the context in which they are operating” [15].

IBM global analysis report on ‘cost of data breach’ mentions that a data breach caused by human error takes around 162 days to identify and 59 days to contain. Among the root causes of data breaches, the report identifies that 25% of data breaches are caused due to human factors [16]. Considering these stats in conjunction with NIST report, identifies one possible reason for such high number of breaches due to human errors i.e. due to overlooking human factors while designing the security systems. We extend this argument to postulate that security features are unnecessarily complex thereby increasing the chances of error.

As early as in 1998, Whitten and Tygar suggested the need for developers (of security functionality) to think from user’s perspective. They further stated that designers of security systems should not assume that the users will read manuals for configuration, instead, the security should be easy to use [17].

The study [18] revealed results of analysis of 32 million passwords for a service, among which 1% were merely "123456" and around 20% of the passwords were the user's name, slang or a common dictionary word. These stats basically describe the user's will, as stated by authors [19], "unless you stand over them with a loaded gun, users will disable, evade, or avoid any security system that proves to be too burdensome or bothersome".

Usable security poses a distinct challenge that needs to be addressed, while working on security of the system. With reference to stats discussed earlier, it is relevant to state that developing a system without incorporating human aspects even being secure against external threats, would be susceptible to: (1) user mistakes ultimately leading to system compromise, (2) increased user disengagement and frustration, (3) users working around anything necessary to do their job [20].

It is important to mention that security and usability conflict is not limited to usability of the interface, and should not also be considered as limited to studies featuring passwords and other authentication mechanisms; however, there are other instances of this conflict beyond just authentication and user interfaces. One such example features conflicts arising with deployment of complex encryption ciphers, which impact 'understandability' of human users while implementing 'confidentiality' (a security mechanism). Furthermore, the authors [20] state, "researchers have identified an increasing number of security mechanisms that are so unusable that the intended users either circumvent them or give up on a service rather than suffer the security". Therefore, it is imperative to consider all aspects of the conflict between security and usability, otherwise we risk building complex secure systems that are susceptible to user mistakes ultimately leading to security compromises.

3.2 How to Engineer the Security and Usability Conflicts?

Figure 1 portrays the proposed four-staged process oriented framework. The framework provides sequence of activities to be followed in order to address the conflict. The framework helps in identifying the conflicts between security and usability while documenting balanced solutions (right trade-offs) in the format of patterns. The four major activities that form basis of this framework are as under.

1. *Analysis* of the diverse human experiences and tasks of the stakeholders and end-users that involve security technologies, modeling of the interaction between stakeholders and user's interaction to accomplish those tasks, and quantifying the possible usability problems.
2. *Modelling* of the relationship between security and usability using as input the descriptions of human experiences, tasks and usability problems identified in the previous step.
3. *Development* of the solutions and their documentation in the format of patterns. The solutions can be used by participating organizations to enhance usability of existing security technologies or the development of new ones.

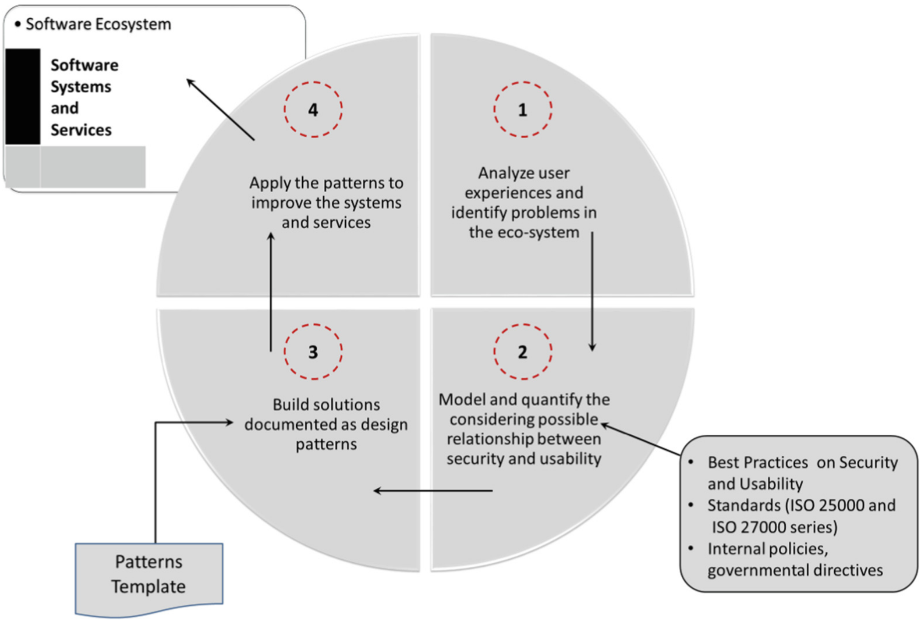

• Software Ecosystem
Software Systems and Services
4
Apply the patterns to improve the systems and services
1
Analyze user experiences and identify problems in the eco-system
3
Build solutions documented as design patterns
2
Model and quantify the considering possible relationship between security and usability
• Best Practices on Security and Usability
• Standards (ISO 25000 and ISO 27000 series)
• Internal policies, governmental directives
Patterns Template

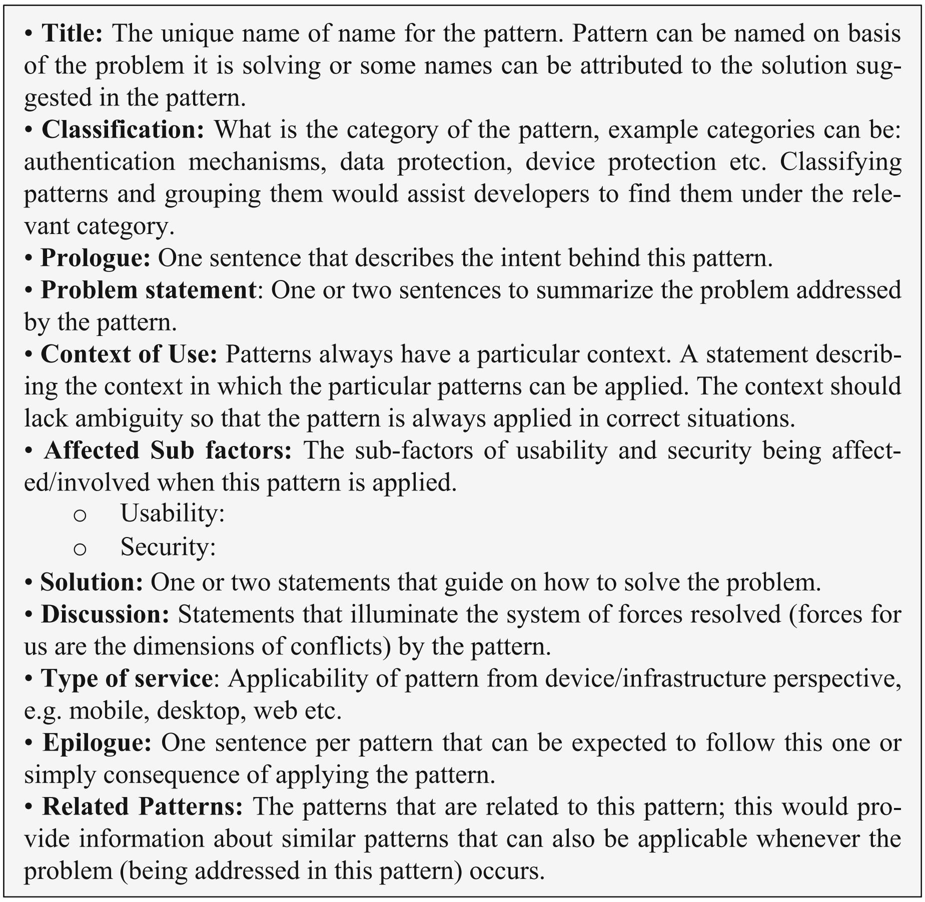
Fig. 1. The proposed process-oriented framework for engineering conflicts between security and usability

4. *Application* of the documented patterns in the software eco-system. Pattern can serve as an effective tool for developers in order to deal with usability concerns in security services. Particularly, the patterns can serve less experienced developers and free-lancers in influencing their decision making abilities when it comes to the security and usability conflicts.

We have been developing and refining this framework using a series of experiments in lab and industry following a design science research (DSR) approach. The main advantage of DSR is the "build-and-evaluate" loop, which allows suggestions from community to be incorporated in the evolved versions of the framework.

As evident from the Fig. 1, the first step involves identification of the conflicts. For this purpose, user studies, cognitive walkthroughs, heuristic evaluations are conducted. Once the conflicts are identified the relationship between security and usability are modelled and quantified. Best practices and standards on security and usability also come into play when modelling the relationship between the two. When do's and don'ts from the perspective of security and usability are known after accessing the underlying best practices, standards and directives, the security and usability professionals brainstorm together to build a balanced solution (the right trade-off) between the two conflicting characteristics. The right trade-offs along with other necessary information are then documented as design patterns. A standardized template for documenting the usable security patterns is presented in Fig. 2.

Once the pattern is documented it can be applied to solve the recurring problem in the software eco-system in similar context of use. The pattern is expected to facilitate

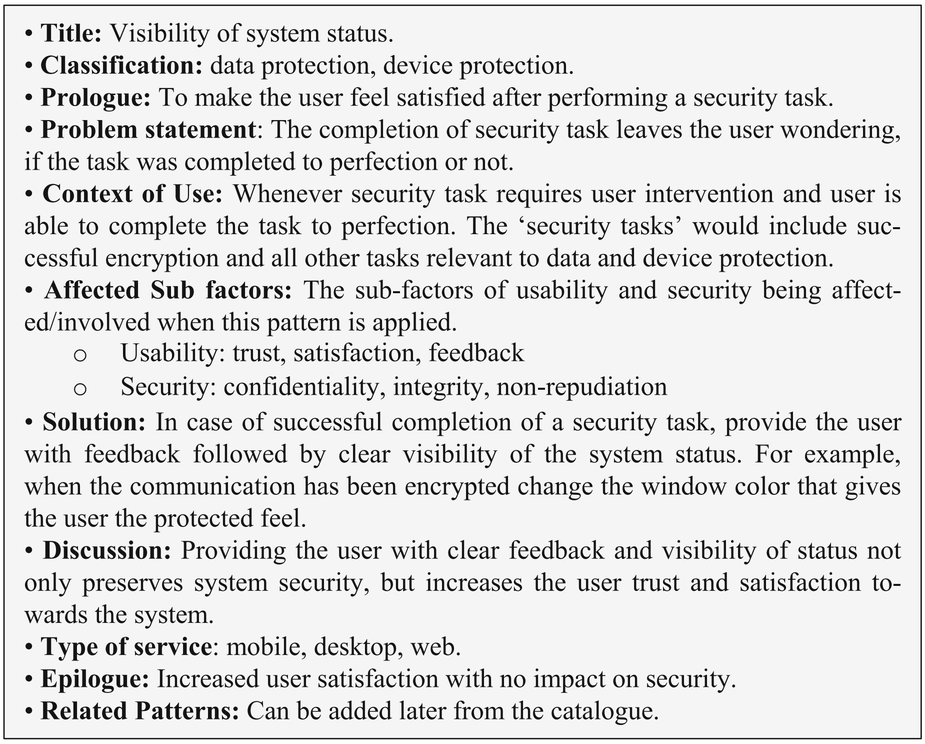


- **Title:** The unique name of name for the pattern. Pattern can be named on basis of the problem it is solving or some names can be attributed to the solution suggested in the pattern.
- **Classification:** What is the category of the pattern, example categories can be: authentication mechanisms, data protection, device protection etc. Classifying patterns and grouping them would assist developers to find them under the relevant category.
- **Prologue:** One sentence that describes the intent behind this pattern.
- **Problem statement**: One or two sentences to summarize the problem addressed by the pattern.
- **Context of Use:** Patterns always have a particular context. A statement describing the context in which the particular patterns can be applied. The context should lack ambiguity so that the pattern is always applied in correct situations.
- **Affected Sub factors:** The sub-factors of usability and security being affected/involved when this pattern is applied.
    - ○ Usability:
    - ○ Security:
- **Solution:** One or two statements that guide on how to solve the problem.
- **Discussion:** Statements that illuminate the system of forces resolved (forces for us are the dimensions of conflicts) by the pattern.
- **Type of service**: Applicability of pattern from device/infrastructure perspective, e.g. mobile, desktop, web etc.
- **Epilogue:** One sentence per pattern that can be expected to follow this one or simply consequence of applying the pattern.
- **Related Patterns:** The patterns that are related to this pattern; this would provide information about similar patterns that can also be applicable whenever the problem (being addressed in this pattern) occurs.

Fig. 2. Usable security patterns template

developers and designers in making reasonably accurate choices when it comes to the conflicts between security and usability. Both usability and security professionals recognize the importance of incorporating their concerns throughout the design cycle and acknowledge the need for an iterative, rather than a linear design process. Design patterns have shown their effectiveness in supporting a smooth integration and cross-pollination of communities [21]. Patterns also assist in an improved communication among team members from different disciplines by developing a common language or vocabulary when explaining design. For elaborating how a usable security pattern would look like, an example pattern is presented in Fig. 3.

It is pertinent to state that one pattern solves one problem only, therefore, an entire catalogue of patterns is required to support the development of simultaneously usable and secure software systems. The documented patterns can be disseminated among the community of developers and designers using online pages, conducting developer workshops and symposiums, research publications, etc.



- **Title:** Visibility of system status.
- **Classification:** data protection, device protection.
- **Prologue:** To make the user feel satisfied after performing a security task.
- **Problem statement**: The completion of security task leaves the user wondering, if the task was completed to perfection or not.
- **Context of Use:** Whenever security task requires user intervention and user is able to complete the task to perfection. The 'security tasks' would include successful encryption and all other tasks relevant to data and device protection.
- **Affected Sub factors:** The sub-factors of usability and security being affected/involved when this pattern is applied.
  - Usability: trust, satisfaction, feedback
  - Security: confidentiality, integrity, non-repudiation
- **Solution:** In case of successful completion of a security task, provide the user with feedback followed by clear visibility of the system status. For example, when the communication has been encrypted change the window color that gives the user the protected feel.
- **Discussion:** Providing the user with clear feedback and visibility of status not only preserves system security, but increases the user trust and satisfaction towards the system.
- **Type of service**: mobile, desktop, web.
- **Epilogue:** Increased user satisfaction with no impact on security.
- **Related Patterns:** Can be added later from the catalogue.

Fig. 3. Visibility of system status pattern

4 Conclusion

Security cannot be achieved in real sense unless it is usable by the users. This research advocates an evolving approach from 'user is the problem' to 'user must be a part of technology based solution'. This paper research is an attempt towards aligning security and usability, and for that a process oriented framework is presented. The framework governs the process from identification of the conflicts to documentation of the right trade-offs in form of re-usable design patterns.

Design patterns can also prove to be effective in handling inconsistency of views between different communities, and between academia and industry, by providing shared documentation in form of patterns. The patterns' ability to be improved over the time provides a common ground to incorporate several views i.e. from industry and academia. With the use of patterns, it is imperative to ensure that they are applied in relevant context of use. We have also developed a usable security patterns template to standardize the documentation of the patterns. For standardized documentation, a pattern template encapsulating information like, title, classification, prologue, problem statement, context of use, solution, discussion, is presented in this paper. To instantiate the use of patterns, a novel pattern 'visibility of system status' is also presented. It is worthwhile to state that one pattern addresses only one instance of the conflict, therefore, it is unrealistic to expect a pattern to solve a systems' problem; however, a catalogue of patterns addressing different instances of the conflict between security and usability would be required in this regard.

References

1. ISO 25010, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (2011)
2. Password Guidance: Simplifying Your Approach. The National Cyber Security Centre. https://www.ncsc.gov.uk/guidance/password-guidance-simplifying-your-approach
3. Naqvi, B., Seffah, A.: A methodology for aligning usability and security in systems and services. In: International Conference on Information System Engineering (ICISE), pp. 61–66. IEEE (2018)
4. Dhillon, G., Oliveira, T., Susarapu, S., Caldeira, M.: Deciding between information security and usability: developing value based objectives. Comput. Hum. Behav. **61**, 656–666 (2016)
5. Garg, H., Choudhury, T., Kumar, P., Sabitha, S.: Comparison between significance of usability and security in HCI. In: 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT), pp. 1–4 (2017)
6. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: how much usability can you sacrifice for security? IEEE Secur. Priv. **15**, 24–29 (2017)
7. Bai, W., Kim, D., Namara, M., Qian, Y., Kelley, P.G., Mazurek, M.L.: Balancing security and usability in encrypted email. IEEE Internet Comput. **21**, 30–38 (2017)
8. Sasse, M.A., Smith, M., Herley, C., Lipford, H., Vaniea, K.: Debunking security–usability tradeo myths. IEEE Secur. Priv. **14**(5), 33–39 (2016)
9. Cranor, L.F., Buchler, N.: Better together: usability and security go hand in hand. IEEE Secur. Priv. **12**, 89–93 (2014)
10. Hof, H.-J.: User-centric IT security-how to design usable security mechanisms. In: 5th International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC), pp. 7–12 (2012)
11. Sahar, F.: Tradeoffs between usability and security. Int. J. Eng. Tech. **5**, 434–437 (2013)
12. Fahl, S.: Confidentiality as a service—usable security for the cloud. In: 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 153–162 (2012)
13. Mairiza D., Zowghi, D.: An ontological framework to manage the relative conflicts between security and usability requirements. In: 3rd International Workshop on Managing Requirements Knowledge (MARK), pp. 1–6 (2010)
14. Hausawi, Y.M., Allen, W.H.: An assessment framework for usable-security based on decision science. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2014. LNCS, vol. 8533, pp. 33–44. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07620-1_4
15. National Institute of Standards and Technology. NISTIR 8080 Usability and Security Considerations for Public Safety Mobile Authentication (2016)
16. IBM: Cost of Data Breach Study: Global Analysis by Ponemon Institute LLC, Sponsored by IBM (2016)
17. Whitten, A., Tygar, J.D.: Usability of security: A case study. School of Computing Science, Carnegie Mellon University. Rep. Technical Report CMU-CS-98-155 (1998)
18. Imperva: Application Defense Center: Consumer Password Worst Practices. http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf

19. Ben-Asher, N., Kirschnick, N., Sieger, H., Meyer, J., Ben-Oved, A., Möller, S.: On the need for different security methods on mobile phones. In: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, pp. 465–473 (2011)
20. Glass, B.D., Jenkinson, G., Liu, Y., Sasse, M.A., Stajano, F., Spencer, M.: The usability canary in the security coal mine: a cognitive framework for evaluation and design of usable authentication solutions. In: Internet Society. Wiley (2003). https://www.wiley.com/en-ad/Multiple+User+Interfaces:+Cross+Platform+Applications+and+Context+Aware+Interfaces-p-9780470854440
21. Seffah, A., Javahery, H.: Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces. Wiley (2014)
22. Garfinkel, S., Lipford, H.R.: Usable security, history, themes and challenges. Morgan and Claypool Publishers, San Juan (2014)

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

895. BERGMAN, JUKKA-PEKKA. Managerial cognitive structures, strategy frames, collective strategy frame and their implications for the firms. 2020. Diss.

896. POLUEKTOV, ANTON. Application of software-defined radio for power-line-communication-based monitoring. 2020. Diss.

897. JÄRVISALO, HEIKKI. Applicability of GaN high electron mobility transistors in a high-speed drive system. 2020. Diss.

898. KOPONEN, JOONAS. Energy efficient hydrogen production by water electrolysis. 2020. Diss.

899. MAMELKINA, MARIA. Treatment of mining waters by electrocoagulation. 2020. Diss.

900. AMBAT, INDU. Application of diverse feedstocks for biodiesel production using catalytic technology. 2020. Diss.

901. LAAPIO-RAPI, EMILIA. Sairaanhoitajien rajatun lääkkeenmääräämistoiminnan tuottavuuden, tehokkuuden ja kustannusvaikuttavuuden arviointi perusterveydenhuollon avohoidon palveluprosessissa. 2020. Diss.

902. DI, CHONG. Modeling and analysis of a high-speed solid-rotor induction machine. 2020. Diss.

903. AROLA, KIMMO. Enhanced micropollutant removal and nutrient recovery in municipal wastewater treatment. 2020. Diss.

904. RAHIMPOUR GOLROUDBARY, SAEED. Sustainable recycling of critical materials. 2020. Diss.

905. BURGOS CASTILLO, RUTELY CONCEPCION. Fenton chemistry beyond remediating wastewater and producing cleaner water. 2020. Diss.

906. JOHN, MIIA. Separation efficiencies of freeze crystallization in wastewater purification. 2020. Diss.

907. VUOJOLAINEN, JOUNI. Identification of magnetically levitated machines. 2020. Diss.

908. KC, RAGHU. The role of efficient forest biomass logistics on optimisation of environmental sustainability of bioenergy. 2020. Diss.

909. NEISI, NEDA. Dynamic and thermal modeling of touch-down bearings considering bearing non-idealities. 2020. Diss.

910. YAN, FANGPING. The deposition and light absorption property of carbonaceous matter in the Himalayas and Tibetan Plateau. 2020. Diss.

911. NJOCK BAYOCK, FRANCOIS MITERAND. Thermal analysis of dissimilar weld joints of high-strength and ultra-high-strength steels. 2020. Diss.

912. KINNUNEN, SINI-KAISU. Modelling the value of fleet data in the ecosystems of asset management. 2020. Diss.

913. MUSIKKA, TATU. Usability and limitations of behavioural component models in IGBT short-circuit modelling. 2020. Diss.

914. SHNAI, IULIIA. The technology of flipped classroom: assessments, resources and systematic design. 2020. Diss.

915. SAFAEI, ZAHRA. Application of differential ion mobility spectrometry for detection of water pollutants. 2020. Diss.

916. FILIMONOV, ROMAN. Computational fluid dynamics as a tool for process engineering. 2020. Diss.

917. VIRTANEN, TIINA. Real-time monitoring of membrane fouling caused by phenolic compounds. 2020. Diss.

918. AZZUNI, ABDELRAHMAN. Energy security evaluation for the present and the future on a global level. 2020. Diss.

919. NOKELAINEN, JOHANNES. Interplay of local moments and itinerant electrons. 2020. Diss.

920. HONKANEN, JARI. Control design issues in grid-connected single-phase converters, with the focus on power factor correction. 2020. Diss.

921. KEMPPINEN, JUHA. The development and implementation of the clinical decision support system for integrated mental and addiction care. 2020. Diss.

922. KORHONEN, SATU. The journeys of becoming ang being an international entrepreneur: A narrative inquiry of the "I" in international entrepreneurship. 2020. Diss.

923. SIRKIÄ, JUKKA. Leveraging digitalization opportunities to improve the business model. 2020. Diss.

924. SHEMYAKIN, VLADIMIR. Parameter estimation of large-scale chaotic systems. 2020. Diss.

925. AALTONEN, PÄIVI. Exploring novelty in the internationalization process - understanding disruptive events. 2020. Diss.

926. VADANA, IUSTIN. Internationalization of born-digital companies. 2020. Diss.

927. FARFAN OROZCO, FRANCISCO JAVIER. In-depth analysis of the global power infrastructure - Opportunities for sustainable evolution of the power sector. 2020. Diss.

928. KRAINOV, IGOR. Properties of exchange interactions in magnetic semiconductors. 2020. Diss.

929. KARPPANEN, JANNE. Assessing the applicability of low voltage direct current in electricity distribution - Key factors and design aspects. 2020. Diss.

930. NIEMINEN, HARRI. Power-to-methanol via membrane contactor-based CO2 capture and low-temperature chemical synthesis. 2020. Diss.

931. CALDERA, UPEKSHA. The role of renewable energy based seawater reverse osmosis (SWRO) in meeting the global water challenges in the decades to come. 2020. Diss.

932. KIVISTÖ, TIMO. Processes and tools to promote community benefits in public procurement. 2020. Diss.

LUT
University

ISBN 978-952-335-585-9
ISBN 978-952-335-586-6 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491
Lappeenranta 2020