1-1-2020

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

Bilal Naqvi
LUT University

Ahmed Seffah
Zayed University, ahmed.seffah@zu.ac.ae

Alain Abran
École de Technologie Supérieure

Cogent Engineering

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/oaen20

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

Bilal Naqvi, Ahmed Seffah & Alain Abran |

To cite this article: Bilal Naqvi, Ahmed Seffah & Alain Abran | (2020) Framework for examination of software quality characteristics in conflict: A security and usability exemplar, Cogent Engineering, 7:1, 1788308, DOI: 10.1080/23311916.2020.1788308

To link to this article: https://doi.org/10.1080/23311916.2020.1788308

Published online: 03 Jul 2020.

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

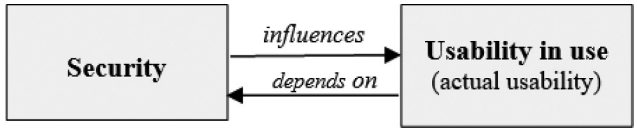

Figure 1. Relationship between usability and security in terms of quality views.

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.

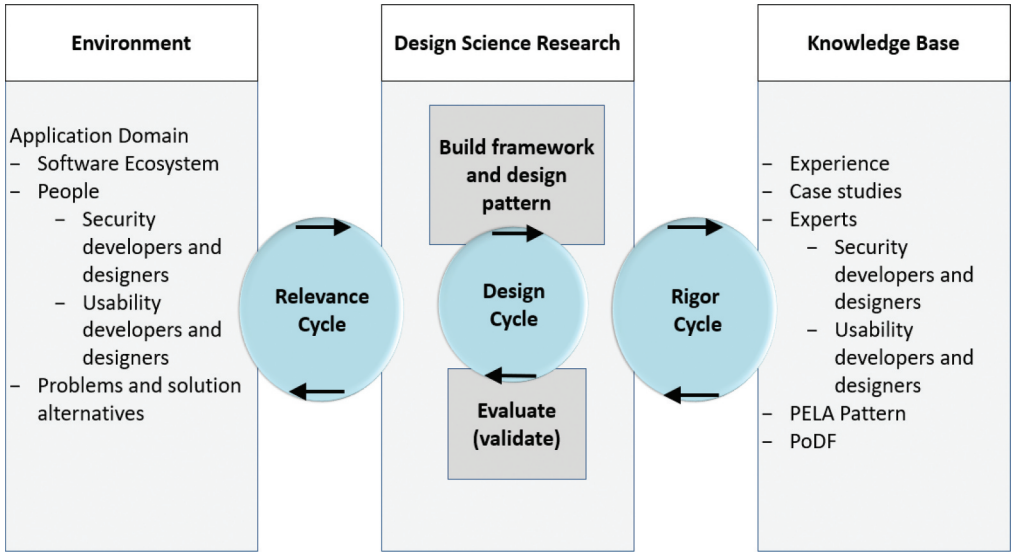

Figure 2. *Design science research process adopted and re-drawn in particular context (2007).*

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.

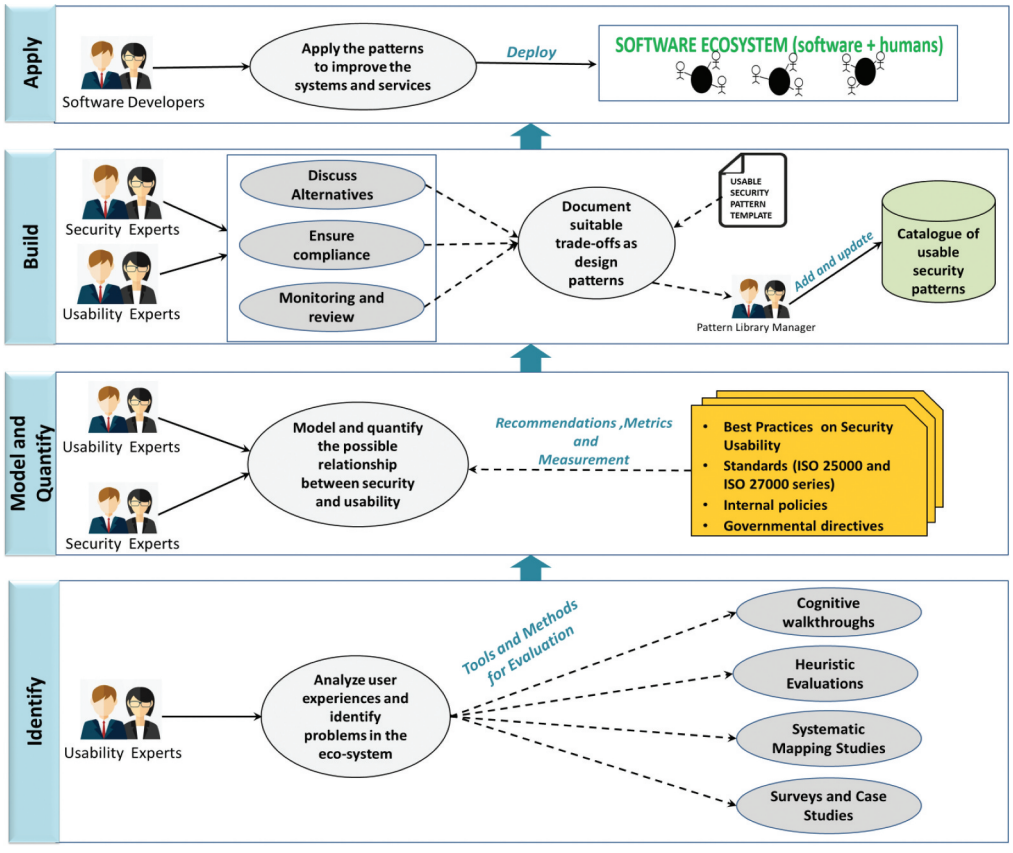

Figure 3. The pattern-oriented design framework (PoDF) for security and usability.

- 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 inter-faces*. 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

Cogent Engineering **(ISSN:) is published by Cogent OA, part of Taylor & Francis Group.**

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com