Incorporating the Human Facet of Security in Developing Systems and Services

Naqvi, Bilal; Clarke, Nathan; Porras, Jari

Published in:
Information and Computer Security

DOI:
10.1108/ICS-11-2019-0130

Publication date:
2020

Incorporating the Human Facet of Security in Developing Systems and Services

Journal:	*Information and Computer Security*
Manuscript ID	ICS-11-2019-0130.R2
Manuscript Type:	Original Article
Keywords:	Usable security, patterns, usability, Security, framework, cyber-physical systems

Incorporating the Human Facet of Security in Developing Systems and Services

Abstract

Purpose – The purpose of this paper is to present an integrative framework for handling the security and usability conflicts during the system development lifecycle. The framework has been formulated while considering key concerns raised after conducting a series of interviews with practitioners from the industry. The framework is aimed at assisting system designers and developers in making reasonably accurate choices when it comes to the trade-offs between security and usability. The outcomes of employing the framework are documented as design patterns, which are disseminated among the community of system designers and developers for use in other but similar contexts.

Design/methodology/approach – A Design Science Research (DSR) approach was used to develop the integrative framework for usable security. Interviews were conducted for identification of the key concerns; however, the framework was validated during a workshop. Moreover, to validate the patterns' template and the usable security pattern identified after instantiating the framework, a survey instrument was used.

Findings – (1) It is important to consider the usability aspect in the development of security systems, otherwise, the systems despite being secure against attacks would be susceptible to user mistakes leading to compromises. (2) It is worthwhile to handle usable security concerns right from the start of system development lifecycle. (3) Design patterns can help the developers in assessing the usability of their security options.

Practical Implications – The framework would assist the designers and developers in handling the security and usability conflicts right from the start of the system development lifecycle. The patterns documented after employing the framework would help not only the designers and developers working in the industry, but also freelancers.

Originality/value – The authors present a novel framework to handle the security and usability conflicts during the system development lifecycle. The development process of the framework was driven by the concerns raised after a series of interviews with the practitioners from industry. The framework presented in this paper was validated during a workshop in which it was exposed for review and comments by the participants from the industry. To demonstrate the use of patterns in general and the framework in particular, a case study featuring smart grids from the domain of cyber-physical systems is presented, which (to the best knowledge of the authors) features the first work relevant to usable security in the domain of cyber-physical systems.

Keywords Usable security, patterns, usability, security, framework, cyber-physical systems, design science research.

Paper type Research paper

1. Introduction

The increased reliance on technical infrastructures like cloud and cyber-physical systems (CPS), and the ever-growing scope of services they support has made their monitoring and security considerably harder tasks. Organizations deploying these systems for managing day-to-day operations are spending billions of dollars on security technologies to ensure the confidentiality and integrity of their data, and availability of their services (RSAC, 2019). However, the increasing complexity of the threat model where these systems are deployed has been so demanding in terms of providing robustness of features and complying with technology needs that it has left less time and budget to consider other vital aspects like the usability of security and enforcing policy-compliant behavior (Kirlappos and Sasse, 2014). In the case of services like smart grids, human errors leading to security breaches can have a more severe impact because of the safety-critical nature of the services.

Traditionally, security and usability have evolved independently and the recommendations from security and usability perspectives are often in conflict. Typical examples of the conflicts include, strong password guidelines, which is good from a security perspective but from usability perspective such a password adds cognitive burden on the user (to memorize the password and type it correctly), and, password masking, which is alright from a security perspective to protect against 'shoulder surfing' and other attacks, but at the cost of usability element of 'feedback' Furthermore, having evolved independently, expertise in both security and usability is hard to find in one person.

A case study finding reveals 'developer knows best' approach, which means that incorporating the human aspects in security design and development is reliant on the skills of developers who are either experts in security, or in usability (Caputo *et al*., 2016). Consequently, the emphasis of security developers has always been to make the system secure and un-exploitable, which as a byproduct may leave the system less usable (Naqvi and Seffah, 2018).

Despite the challenges posed by security as a quality, it is important to note that it is the human user in different roles, who interacts with the security functionality to protect the system; therefore, it is imperative to consider the human aspects while designing the security systems and services. With incorporating the human aspects, it is expected to consider in security design the elements of usability (as identified and defined by ISO 25010 standards) such as effectiveness, efficiency, *effective in use*, and the elements of user experience (UX), which include memorability, findability, satisfaction, credibility, (Morville, 2004). The domain considering human aspects related to security and the integration of usability in the security design is commonly referred to as usable security. Moreover, the state of the art concerning the integration of human aspects in security design identifies gaps (Naqvi and Seffah, 2019), including, failure of security specialists to address usability as perceived and defined by the human computer interaction (HCI) community, and industry's behavior being more driven by bug fixing rather than trying to examine and consider the context in which the bugs occurs.

Caputo *et al*., (2016) 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". Security developers are often not trained in handling usability concerns and vice versa yet management of the conflicts is reliant on skill of the developers. This identifies the need for assisting the developers in management of the conflicts. To do so, the following issues need to be explored.

1. How can the conflicts and suitable trade-offs (addressing those conflicts) be identified and documented during the system development lifecycle?

2. Can design patterns be used to disseminate the suitable trade-offs thereby assisting the system designers and developers in managing the conflicts?

While considering these issues, this paper presents an integrative framework for usable security (IFUS), integrative in a way that the framework is based on combining the principles of security and elements of usability in the development of systems and services. The IFUS has been co-created with the industry while considering the concerns raised after a series of interviews with security and usability practitioners, and limitations in the methodology Naqvi and Seffah, (2018). Moreover, the IFUS was validated by conducting workshop involving the practitioners who had participated in the interviews. During the workshop IFUS was exposed for review and comments by the experts, and it was adopted. The IFUS governs management of the security and usability conflicts within the scope of system development lifecycle (SDLC) and allows security and usability concerns to be incorporated collectively right from the start of the SDLC. In IFUS, besides identification the management of conflicts include elicitation of suitable trade-offs, which are documented as patterns to support re-use and assist the community of designers and developers in handling the conflicts in a particular context of use. Patterns in our perception are artifacts documented in natural language and addressing a repetitive problem (Naqvi and Seffah, 2019). Patterns are practical and describe instances of "good" design principles. Patterns solving a particular usable security problem can be disseminated among the community of developers and designers to assist them in making decisions regarding the usability of security, thus delivering a balanced solution addressing the conflict and maximizing both factors. To standardize the documentation of usable security patterns, a template to document the patterns is also presented. Furthermore, in the line with the design science research method a case study was conducted to instantiate (demonstrate) the IFUS and identify a design pattern. The pattern was later validated by involving a group of developers.

The rest of the paper is organized as follows. Section 2 discusses the background and related work. Section 3 presents the Integrative Framework for Usable Security (IFUS). Section 4 presents a case study to instantiate the IFUS. Section 5 presents the results of survey conducted to validate the patterns' template and the pattern identified after instantiating IFUS. Section 6 presents the discussion, and Section 7 concludes the paper.

2. Background and Related Work

2.1 Background

Before presenting the framework, it is worthwhile to discuss two questions.

1. *Why* we should consider handling usable security concerns in the SDLC
2. *Where* in the SDLC should we consider handling usable security concerns

Each of these questions is discussed in subsequent sub-sections.

2.1.1 Why we should consider handling usable security concerns?

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). The human element is one of the most critical, yet unaddressed element in computer security research (NISTIR 8080, 2016). The report NISTIR-8080 (2016) 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".

Whitten and Tygar as early as in 1998 identified the need for developers (of security functionality) to think from the user's perspective. The authors stated that designers of security systems should not assume that the users would read manuals for configuration; instead, the security should be easy to use (Whitten and Tygar, 1998), however, despite this recognition more than 20 years ago, the integration of usability aspects in security design still pose a challenge.

It is relevant to consider the usability aspects in the security design as a key factor of security hygiene (Kirlappos and Sasse, 2014). Otherwise, the developed security systems despite being secure against external threats could be susceptible to:

- User mistakes ultimately leading to system compromise.
- Increased user disengagement and frustration.
- Users working around anything necessary to do their job (Glass *et al*., 2016) e.g. in case of complex authentication systems users would employ unwanted techniques like pretexting, reusing credentials, wherever possible.

It is relevant to note that initially human aspects or usability of security was considered as limited to usability of the security interfaces, however, with time it was realized that it is intended to incorporate in security the elements of usability (as identified and defined standards such as ISO 25010) and the elements of user experience (UX) (Morville, 2004). **Zagouras et al., (2017) presented concepts and definitions regarding incorporating the user experience into usable security. The authors assert that user experience 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. As a step further, the work from S. Mahlke (2007) presents a framework for integration of non-instrumental qualities, symbolic aspects and emotional user reactions to traditional approaches of interaction. This work could be helpful in incorporating the elements of user experience to usable security. In addition, the author demonstrated with the help of three case studies the importance of system properties, user characteristics and context parameters on user experience.**

To cope with this challenge, the researchers studying usable security started investigating several avenues including, usability issues arising with user authentication, usability issues arising with email security and public key infrastructure (PKI), anti-phishing efforts, web-privacy and fair information practice; however, there are other as well (Garfinkel and Lipford, 2014). With a broader scope of security services ranging from authentication to email security and web-based services such as behavioral advertising, there is need of a generalized solution within the scope of system development lifecycle to address the usable security challenges arising in various contexts.

Moreover, usable security challenges are generally in the form of conflicts. It is worthwhile to note that there are conflicts in each of the areas identified earlier, for example:

1. Study of text passwords features a conflict between *authentication* (a security mechanism) and *memorability* (a usability element) (Naqvi and Seffah 2018).
2. Various graphical password schemes in which authenticating the user takes longer than the text passwords feature a conflict between *authentication* (a security mechanism) and *efficiency* (a usability element) (Garfinkel and Lipford, 2014).
3. Email security and PKI based systems feature conflict between *confidentiality* (a security mechanism) and *understandability* (a usability element) (Garfinkel and Lipford, 2014).

However, more conflicts arise in industry during the development of state-of-the-art systems and services, therefore, it is vital that the solution to cater the usable security challenge includes:

1. A mechanism for identification and documentation of conflicts as they arise.
2. A mechanism for identification and elicitation of suitable trade-offs.
3. Encapsulating the information about the *conflict*, *context* in which it occurred, and *suitable trade-offs;* and disseminating this information among the community of developers and designers to assist them in managing the conflicts.

2.1.2 Where in SDLC should we consider handling usable security concerns?

Yee (2004) suggested that security and usability issues should be handled together and early during the design process. Praveen *et al*., (2014), while proposing a model for incorporating security and usability during the requirements phase state that, "security and usability must go together hence both should begin at requirement level". Similarly, Flechais *et al*., (2007), also suggest incorporating usable security concerns from the requirements phase of SDLC. Therefore, for newly developed systems security and usability concerns should be considered early in the system development lifecycle specifically during the requirements and design phase. However, for a system already in the production environment, it is vital to fix all reported usable security problems in the upcoming releases of the system.

From economic perspective, it is worthwhile to handle usable security concerns right from the start of the SDLC to avoid conflict situations thereby circumventing costs and efforts associated with re-work induced due to change in system design at later stages in the development lifecycle.

Having discussed why and where to handle usable security concerns, it is worthwhile to discuss briefly 'how' to do that. Firstly, it is important to identify different approaches to handle usable security concerns, should it be handled as a requirement, as a goal, or, as a constraint. The proposal presented in this paper advocates handling usable security concerns as a requirement. The need for usable security could be enforced by the developing organization or by the customer and the product owner, which should be reflected in the elicited requirements provided to the designers and developers. However, considering usable security as a constraint might lead to several inconsistencies in overall process of integrating security and usability and managing their conflicts, since security itself is considered a constraint to systems functional requirements (Haley *et al.,* 2006*)*. If security is considered as a constraint, then could usability of security be considered as a constraint to a constraint? Therefore, as stated earlier this research advocates handling usable security as a requirement, more details on how to handle usable security concerns are presented in the Section 3.

2.2 Related Work

Much of research work on the topic identifies a tactical approach for addressing the usable security problem (Garfinkel and Lipford, 2014). Tactical in a sense that the solutions are focused at addressing specific problems, therefore, they have a limited impact. What is required are the generic solutions that are addressing the core of the problem.

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

Naqvi and Seffah (2019) present a framework for aligning security and usability during the development of security systems and services. The framework governs aspects from identification of the conflicts to their

resolution in form of suitable trade-offs and documentation as patterns. However, the aim in that paper was to introduce the concept of handling security and usability conflicts during the development lifecycle of security systems, and potential use of design patterns for elicitation of suitable trade-offs. However, the framework does not provide details such as mechanism for identification of the conflicts, elicitation of suitable trade-offs, dissemination mechanism for usable security patterns; all of which have been considered in the current work.

Mujinga *et al.,* (2019) proposed "Socio-technical Information Security Framework". The framework has been designed while considered both technical and social aspects of information security. The authors claim that the development of security application can be improved by applying 12 design principles presented in the framework. The framework is more about providing a list of usable security design principles rather than contributing towards improving industrial processes.

Parveen *et al*., (2014) presented a process-oriented approach for incorporating usability during the security system development lifecycle. In the proposed approach, all the requirements are assessed from which security requirements are extracted. For the security requirements threats, vulnerabilities are identified. The next phase involves the identification of usability requirements against a specific set of characteristics. Finally, security and usability analysis tests are performed to access the outcomes of requirement engineering process are highly secure and highly usable. The aspect which remains unaddressed is the practicality to adopt such a methodology in real industrial contexts such as identification of roles involved during the methodology.

Hausawi and Allen (2014) 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 characteristic of security and usability. The weights determine the relative importance of characteristics to be considered for the requirement specification of software. The authors claim that requirements specified after AFUS have a balance between usability, security and usable security.

Mairza and Zowghi (2010) presented an ontological framework for catering to the usable security conflict. The framework is based on the identification of usability/security requirements, identifying meaning and system context. After that, the conflicts are identified based on system requirements, characterized on the basis of their impact and listed. Then the nature of the identified conflict is determined, and the conflict resolution strategy is devised in line with the system requirements.

Flechais *et al*., (2007) proposed "Appropriate and Effective Guidance for Information Security" (AEGIS). AEGIS presents a way of integrating usable security in the SDLC. It relies on involving the users and other stakeholders while designing the security features of the system. The major focus of the proposed approach is on involving the end-user while designing the system. This method has some limitations in a way that, the user at the requirement and design stage might not be able to perceive a clear picture of the system being developed. Users also have different aptitudes, involving a particular or a group of users does not mean that it incorporates the view of all users around the world.

Moreover, from quality engineering perspective, Feitosa *et al.,* (2015) investigated the trade-offs between sub-characteristics of a critical embedded system. Their investigation shows that the trade-offs are usually in favor of critical quality characteristics. However, the work is limited to identification of conflicts. Zhu *et al*., (2012) proposed a model of fuzzy soft goal interdependency graphs. 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.

From the literature referred above, it is evident that different frameworks exist for aligning human aspects and security, but each one has its own limitations. Some of the limitations include, (1) no mechanism for identification of suitable trade-offs, (2) formulated to be applicable in specific contexts and scenarios, (3) challenges and limitations in applying the findings in a real context. Therefore, there is a need for a framework which aims at incorporating the human facet of security in the development of systems and service while also addressing the limitations in the existing work.

3. Integrative Framework for Usable Security (IFUS)

A framework is an abstraction, underlying a system or a concept. The intent behind the creation of IFUS was to formulate means for facilitating system designers and developers in managing the conflicts. IFUS provides a mechanism for management of conflicts starting from identification of the conflicts to elicitation and documentation of the suitable trade-offs. The documented suitable trade-offs are disseminated among the designers and developers who are experts either in security or usability. It is expected to positively influence their decision-making abilities when it comes to the conflicts between security and usability in other but similar contexts. As stated earlier, the concerns raised after a series of interviews with security and usability practitioners working with our industrial partner were a key driving factor in development of the IFUS. It is pertinent to mention that the IFUS was created as a part of 'co-creation' project funded by a Finnish agency (Business Finland). The roles interviewed during the project included, (1) project manager, (2) lead architects, (3) security engineers, and (4) UX experts. Despite different viewpoints on importance of security and usability independently in their product lines, there was consensus among the interviewees on the following.

1. There are no practices and methods in place for integrating security and usability during the SDLC.
2. Alignment between security and usability has a direct business impact in terms of number of users using the product.
3. Whenever, there are conflicts, informal meetings are conducted to address those conflicts. The outcomes of the meeting are considered as one-off event and are not documented.
4. In conflict situations where the security and usability professionals cannot reach an agreement, the trade-offs are in favor of security because no one is willing to take the risk, and that users should be able to use it as it is.
5. The rule of thumb is, "if it is not possible to design it the easiest way, then try doing it the next easy way".

These concerns are consistent with some of the existing industrial case studies, for instance, the one reported by Caputo *et al.,* (2016). Moreover, in conjunction with the concerns raised after the interviews, we also referred to the limitations in the methodology (Naqvi and Seffah, 2018) identified after presentation at the conference. The research method used for development of the IFUS is design science research (DSR). IFUS could also be viewed as an evolved and extended version of the methodology (Naqvi and Seffah, 2018). This evolution is in line with the fundamentals of design science research, which supports iterative model of development. In this case feedback during the conference and industrial partner's concerns formed a basis for planning and executing the iteration that led to creation of the IFUS.

It is worthwhile to mention that IFUS presented in this paper was validated during a workshop involving the same practitioners who had participated in the interview stage. During the workshop, IFUS was exposed to comments and review by the participants, however other relevant issues discussed during the workshop

do not fit within the scope of this paper. Concerning IFUS, the participants were of the view that IFUS provides a mechanism for communication between the security and usability practitioners, and documentation of the trade-offs as patterns would support dissemination of the solutions among the designers and developers working on different product lines and in other offices of the organization. Moreover, it was also noted that a catalog of patterns identified using IFUS could be helpful in training junior developers. Before describing IFUS, it is worthwhile to explain the process followed for its development.

3.1 Process of creation

Design science is a paradigm involving the design and investigation of the artifacts in a particular context (Wieringa, 2014). The design science paradigm guides the design of *artifacts* and *processes*. The *artifacts*, in this case, are the design patterns; however, the *process* is the IFUS. The development process for the IFUS involved three cycles in line with the principles of design science research identified by Henver (2007).

- **The relevance cycle**: The motivation behind this phase is to improve the environment by introducing new artifacts and processes to build these artifacts. The process is initiated considering a specific problem context and the criteria for evaluation. Then there is a testing phase to assess the impact of the design artifact on the environment. The cycle iterates as much as it is required. In the context of this paper, the problem is the conflict between security and usability, the artifacts are the patterns, and the process is the IFUS.
- **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 experience, existing case studies, existing frameworks, interviews of experts and feedback on the methodology (Naqvi and Seffah, 2018) during the conference.
- **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. The elements in the knowledge base as identified earlier were the key factors considered in planning and executing the iteration of this cycle which led to development of the IFUS. Moreover, it is important to note that the artifacts (patterns) were also validated by conducting a survey, however, due to patterns ability to evolve with time, a pattern after its creation and dissemination is continuously under review by the developers who use the pattern. The request for modification is considered and based on that, either existing patterns are evolved or new ones with different contexts are documented.

Moreover, the design science research method used for the creation of IFUS (*process*) and identification of patterns (*artifacts*) is presented in Figure.1.

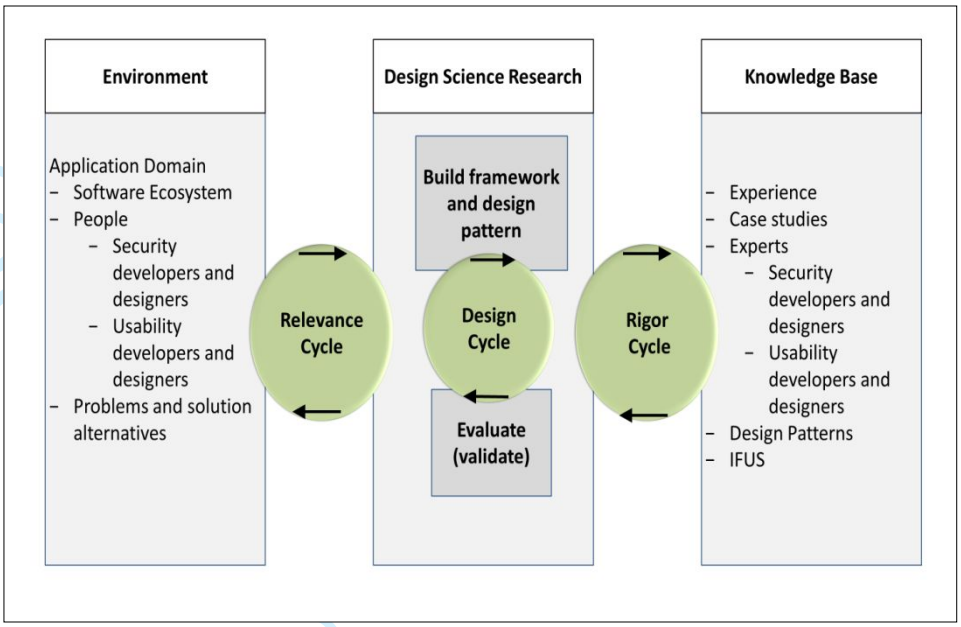

Figure.1. Design Science Research Method (Adopted and re-drawn in particular context from (Henver, 2007))

The three cycles of the design science research process are presented in Figure.1. The arrows with the cycles are representing the iterative nature of the process. The environment is the software ecosystem, which includes systems and services with security and usability issues and the people involved in handling those issues. Moreover, in the design cycle, the IFUS was created and it contributed to the existing knowledge base with an addition of a process (IFUS) and artifacts (patterns identified using IFUS).

3.2 Integrative Framework for Usable Security (IFUS)

The IFUS has three layers, different elements and activities of the IFUS along with the participants is presented in Figure. 2. A bottom-up approach has been applied to construct the elements of IFUS. The participants during various activities in the IFUS are system designers and developers from security and usability domains. In line with the discussion in Section 2, the IFUS is adopted during the requirement-engineering phase of the SDLC. The outcomes of employing the IFUS are documented as usable security design patterns, which are disseminated among the community of developers for use in other but similar contexts.

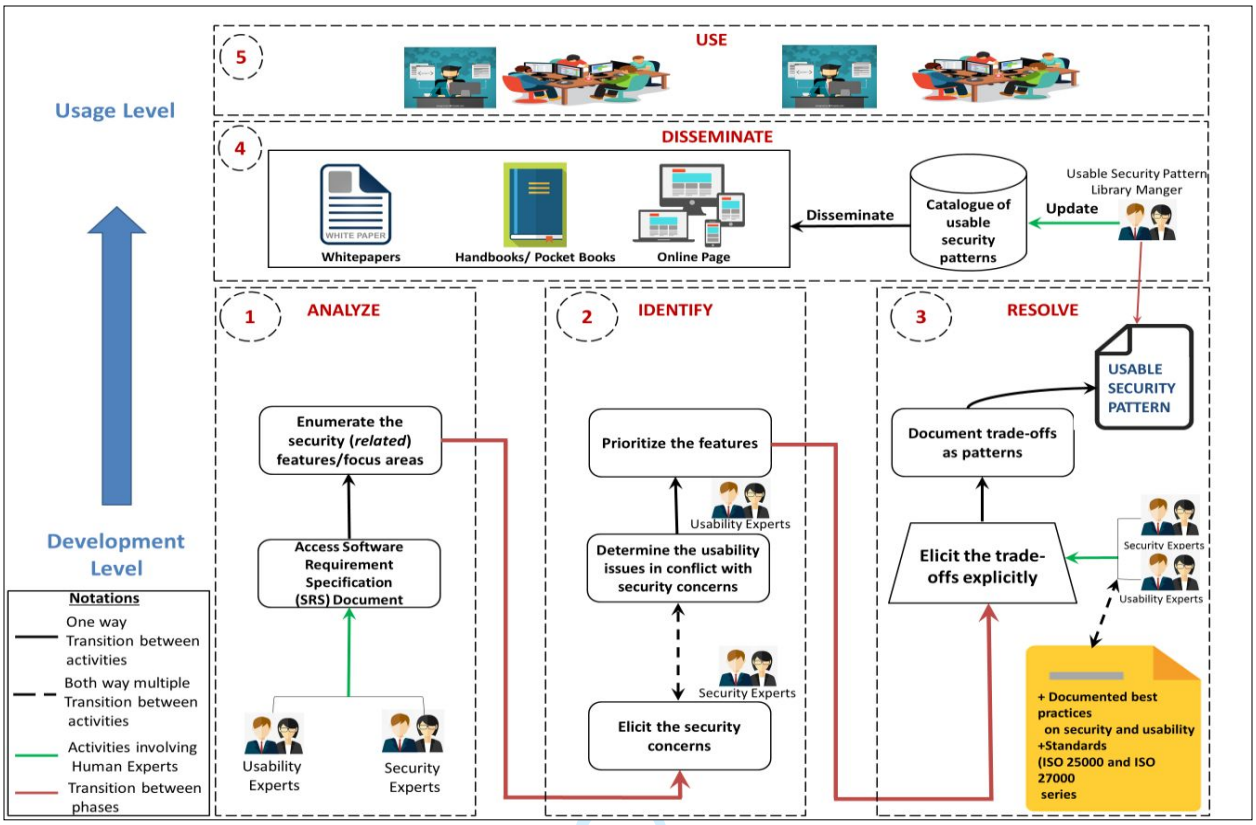

Figure. 2. Integrative Framework for Usable Security (IFUS)

It is worth mentioning here that IFUS does not automate the process but governs the activities to be performed at the requirements and design stage of the system development lifecycle. The key activities of the IFUS are grouped in five distinct phases which are, analyze, identify, resolve, disseminate and use. Feedback on the methodology during the conference and concerns raised during the interviews were considered while arranging the activities in different phases primarily to, (1) group the activities of similar nature, (2) support the layered-architectural pattern for the IFUS. *Analyze* refers to the phase where the requirements are collected and analyzed. In the *identify* phase, the 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, which are documented as patterns to be disseminated in the *disseminate* phase. However, in the *use* phase, the developers use the patterns in other similar contexts.

The details of each activity during the five phases of IFUS are as follows.

- **Access Software Requirement Specification Document**: To initiate the process, security and usability experts access the Software Requirement Specification (SRS) to enumerate security requirements. Different knowledge sources can be used to initiate pattern writing including individual experiences, standards and best practices, specifications, and documents (Riaz *et al*., 2016). IFUS uses the SRS document for the said purpose.
- **Enumerate the security features and focus areas**: The security and usability experts access the SRS document to identify the security requirements of the system. This is done to ensure a specific focus on requirements directly affecting security and its usability.

- **Elicit the security concerns**: For the enumerated features, a specification of what is required from the security point of view is explicitly identified. IFUS works at the granular level, which involves identification of affected sub-characteristics of security (including confidentiality, integrity, and availability, among others). While eliciting the concerns, it is important to consider both internal and external threats.
- **Determine the usability issues in conflict with security concerns**: Once the security concerns are known, the requirement associated with each of the security concerns is subjected to usability analysis to identify instances of potential conflicts. A matrix of sub-characteristics of security (rows) and sub- characteristics of usability (columns) are created (see Figure. 3). Each element of the matrix describes a potential conflict.

Security	Usability		
	Effectiveness	**Efficiency**	**Satisfaction**
Confidentiality			
Integrity			
Availability			
Authentication			

Place an "X" in the cell where there is a potential conflict

Figure. 3. Matrix for describing a potential conflict between at sub-characteristic level

The sub-characteristics of security and usability are presented for exemplary purposes; other characteristics can be added to the matrix based on requirements.

- **Prioritize the features**: To prioritize the requirements in order of severity in terms of usable security consequences, a five-scale schema is presented. The schema for prioritizing the requirements is inspired by the scale suggested by Nielsen and Norman (NN) Group [1]. Severity of usable security problem will be determined based on following factors.

  1. *Frequency* of occurrence of a particular usable security problem
  2. *Possible Impact* that a usable security problem has, if it occurs.
  3. *Persistence* of usable security problem among the stakeholders and potential users, either it is identified once or repeatedly.

  The software requirements analyst is the role associated with this task, who is the one looking after the prioritization task based on the input from security and usability experts. Various system stakeholders including potential end users are involved during this stage to assign values for each of these factors. These factors are combined into a single rating (1-5) to facilitate prioritization and decision making. It is relevant to mention that all these factors have equal weight in determining which scale is to be assigned. Using this criterion, values from 1-5 are assigned to each requirement based on the combined weight of all factors. Further information about the prioritization scale is presented in Naqvi and Seffah (2018).

- **Elicit the trade-off**: Once the prioritized requirements along with security and usability concerns are known, the trade-offs are elicited explicitly while ensuring that both security and usability concerns have been catered with minimum possible compromise to any of these characteristics and their respective sub-characteristics. The main objective served during this activity is that it provides an avenue for integrating security and usability concerns, rather than developing a security centric

[1] https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

system, where the trade-offs are always favoring security**.** For eliciting the trade-offs, the security and usability experts consider the goals from security and usability perspectives concerning the requirement under consideration, and the standards and best practices concerning security and usability. Once goals from security and usability perspective are known, standards and best practices are accessed to tailor a solution causing minimal compromise to the involved characteristics. In case there is a trade-off, it is ensured that in the first instance no violation of the standards and best practices takes place. After this, a risk-based decision process is applied to determine the most optimal solution. In line with the risk-based decision process, the list of possible solutions (trade-offs) is identified, validated and accepted for documentation as patterns (Risktec, 2005). Even after documentation of the suitable trade-offs as pattern, it is subjected to continuous monitoring and improvement by incorporating the feedback from the developers who apply these patterns. This is in line with pattern ability to be evolve and improve with time. This activity is repeated for all security requirements in order of their priority.

- **Document trade-offs as patterns**: Once the suitable trade-offs have been identified, an effective approach is to document these as patterns. The patterns can be disseminated among the community of developers and designers to assist their decision-making abilities when it comes to the conflicts in other but similar contexts. The implementations based on the same patterns can differ, however, in such cases the suitable trade-off presented in the pattern serves as a starting point to build upon, rather than re-inventing the wheel and spending hours of effort on something which has already been done, but not known. A patterns' template to document usable security patterns is presented in Figure. 4.

  - **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. 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 Characteristics:** The sub-characteristics 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.
  - **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

Figure. 4. Template to document usable security patterns

- **Disseminate**: Whenever a new pattern is documented, it is added to the catalog. The catalog can be disseminated among the community of security developers via online pages, pocketbooks for developers, and release of white papers reflecting updates and newest patterns.
- **Use**: This layer forms the topmost layer and the usage level of IFUS, where the common developers use usable security patterns to deliver usable and secure solutions.

Patterns have benefits like reusability, means of providing shared documentation, availability of common vocabulary, to mention a few (Riaz *et al*., 2012). It is imperative to mention here that one pattern

solves only one problem, therefore, a catalog of patterns can be built by applying the proposed framework to cover other aspects and to deliver usable and secure solutions.

4. Instantiating the IFUS

The past decade has experienced enormous technological and social advancements; however, the power grid industry has arguably not revolutionized to keep pace with advancements in other sectors (Wang and Lu, 2013). With the idea of cyber-physical systems (CPS), researchers have begun to work on revolutionizing the existing infrastructure into next-generation power systems referred to as Smart Grids. Smart Grid involves the integration of computing and communication technologies with the existing infrastructure to enhance efficiency, reliability, etc. Due to safety concerns associated with specific services like advanced metering infrastructure (AMI), it is pertinent to consider the cybersecurity aspect as well. Furthermore, when security is considered, it should not be limited to discussing the network vulnerabilities, attack vectors, countermeasures against attacks, but it is imperative to consider the human facet of these security services as well. It is relevant to state that to the authors' best knowledge no similar work (on usable security) before ours exists related to CPS in general and smart grids in particular.

For this case study, a use case related to Smart Grids is considered. For the identification of relevant use cases, 'Guidelines for Smart Grid Cyber Security NISTIR 7628 Revision 1' were assessed (NISTIR 7628, 2014). The document identifies different categories and scenarios relevant to the security and privacy of users and the infrastructure in Smart Grids. Since the focus of this research is on usable security, therefore, a use case involving interaction with the customers ('meter sends information' within the AMI category) was selected. The use case identifies confidentiality and integrity as objectives from a security point of view. However, the case study is detailed considering a scenario where a customer tries to access the AMI from a trusted device to check current months' meter reading of their house. To do so, the customer has to login to the system to get access. This scenario is discussed from the perspective of the IFUS activities below.

- **Access Software Requirement Specification Document**: In line with the activities of the IFUS, the first step is to access the SRS document, however, for illustration of the approach during the case study, NISTIR 7628 was accessed. For the scenario just described, the use case is considered when a customer accesses the AMI to get information about the meter reading data.
- **Enumerate the security features and focus areas**: The security-related features for the considered scenario are presented in Table 1.

Table 1. Enumerated Security Features for the Scenario

Enumerated security features and focus areas
— The AMI service must be available to the customer when accessed from home area network (trusted location).
— The dialogue between AMI and customer home must be encrypted so that there is no unauthorized disclosure.
— The meter reading data displayed to the customer must be the same as maintained in the records.
— The customer must authenticate to AMI to access the information.

- **Elicit the security concerns**: It is important to identify the sub-characteristics of security associated with the identified security features and requirements (in Table 1). For instance, for the feature/requirement, "the customer must authenticate to the AMI to access the information", essential aspects relevant to security such as goal, tasks are presented in Table 2.

Table 2. Security Concerns for the considered requirement

Requirement	Security Goal	Security Tasks/Sub Tasks
The customer must authenticate to the AMI to access the information (meter record)	Authentication	○ Uniquely identify and authenticate users before allowing access to the system. ○ Allow access to the system only after the user has passed the authentication stage. ○ Explicitly specify and document the authorization to user without authentication. ○ Limit the number of parallel sessions for a particular user account ○ Prevent access to the system by locking the system after certain defined period of inactivity, or wrong authentication attempts.

The details of security tasks to achieve the authentication goal have considered the same way as identified by Riaz *et al.,* (2016).

- **Determine the usability issues in conflict with security concerns**: This stage involves determining the usability issues arising with implementation of a specific feature and requirement. This activity is to be repeated for all security features identified during the first activity and for all system security tasks/sub-tasks identified during the second activity. For example, from the perspective of the case study being considered, as a consequence of the security task "prevent access to the system by locking the system after a certain defined period of inactivity, or wrong authentication attempts", a user may find that the account was locked when s/he is back after responding to the doorbell or switching off the stove in kitchen. In this case, the user has to login again. From a security perspective, this is in line with the practices and recommendations. However, from the usability perspective, such implementations add to the cognitive load as for the user going through the security procedures takes longer than the task itself, which is to record meter reading. During system security development, the cognitive burden on the user end is given less priority compared to CIA goals of security leading to cases where the users evade the security procedures or give up on using a service rather than suffering from security (Glass *et al*., 2016). Considering the example just discussed, the matrix describing the potential conflict is presented in Figure. 5.

Security	Usability		
	Effectiveness	Efficiency	Satisfaction
Confidentiality			
Integrity			
Availability			
Authentication		X	X

Figure. 5. Matrix for describing a potential conflict between authentication and satisfaction

The matrix presents conflicts between authentication (security) and efficiency (usability), and between authentication (security) and satisfaction (usability).

Another instance of the conflict arises with the security task "uniquely identify and authenticate users before allowing access to the system". Such tasks leave room for developers and security engineers to make authentication implementations as robust as possible. In such a case, a system

might be implemented in which authentication with password requires "the password must be: at least 16 characters; has not been used in last 5 passwords; should not be a common dictionary word; should not be your name/surname; contains at least three of the fours character groups: English uppercase letters (A-Z), English lowercase letters (a-z), Special characters (@, $, #), Numerals (0-9)". This is all right from the security perspective, but these requirements add the cognitive load on users' mind and have an impact on effectiveness in use and satisfaction. The potential conflicts arising in this regard are presented in Figure.6.

Security	Usability		
	Effectiveness in use	Efficiency	Satisfaction
Confidentiality			
Integrity			
Availability			
Authentication	X		X

Figure. 6. Matrix for describing a potential conflict between authentication and effectiveness, satisfaction

After following the activities just described, the requirements are prioritized using the prioritization scale discussed in the previous section. The output is sorted requirements in order of their priority.

- **Elicit the trade-off**: For the conflicts identified in the previous stage, the standards and best practices concerning security and usability are accessed to elicit the trade-off and document a solution that maximizes both characteristics. For the example under consideration, i.e. when a customer logs in to the AMI service to check the meter reading, and due to some unavoidable event, there is a premature automated logout due to inactivity, the customer is made to login to the service. Such events increase customers' disengagement with the service and have an impact on usability elements like satisfaction and efficiency. Therefore, to elicit the trade-off a solution "whenever there is a premature timeout, the service should notify the customer about the requisite detail using a popup on the screen or a via text message or email" was identified, which can improve the usability and user's experience without any impact on security. The risk-based decision process was considered during the case study while identifying a suitable trade-off (solution). In line with the design science process and as part of monitoring and improvement, the pattern was subjected to a validation study involving the developers, who encounter similar situations. Once a trade-off has been elicited it is documented as a pattern for developers to use.
- **Document trade-offs as patterns**: For the example just discussed, the trade-off was documented as a usable security pattern called "*usable secure record inquiry pattern*" and presented in Figure.7. Each developed pattern is added to the catalog and made accessible to common developers (through online pages, pocketbooks, and whitepapers) for use.

- **Title:** Usable Secure Record Inquiry
- **Classification:** Authentication mechanism, device protection
- **Prologue:** To increase customer satisfaction while using AMI or related services
- **Problem statement**: While using AMI service, the customer due to certain period of inactivity on the service might find the account as locked to ensure security of the service.
- **Context of Use:** Whenever there is a premature and automated logout from the system with the user unable to perform the main task, and with login the customer finds that the security procedures take longer than the task itself.
- **Affected Sub-characteristics:** The sub-characteristics of usability and security being affected/involved when this pattern is applied.
    - *Usability:* satisfaction, efficiency
    - *Security:* authentication, confidentiality
- **Solution:** The customer should be notified of details about the record using a popup on the screen or via email or text messages, so that s/he does not have to login from to system.
- **Discussion:** In this case, premature log out due to inactivity would have caused a negative impact on the user experience (satisfaction, desirability). So rather than doing so, this pattern suggests to transmit the requisite information so that the user does not need to login again, thus balancing between security and usability, thereby increasing both of them.
- **Type of service**: CPS, smart grids, AMI
- **Epilogue:** Increased user satisfaction with no impact on security or in other words a usable secure authentication process.
- **Related Patterns:** Can be added later from the catalogue

Figure. 7. Usable Secure Record Inquiry Pattern

5. Validation Study

In line with the principles of design science research, the artifacts are created and validated (Wieringa, 2014). The artifacts, in this case, are the patterns created using IFUS. A validation study was conducted to validate the patterns' template and the usable secure record inquiry pattern. The methodology for the study was adopted from Arteaga *et al*., (2009). The details of participants as specified during the study are presented in Table 3. The participants were recruited during a developer's workshop held at the Lappeenranta University of Technology, and participation in the study was voluntary. The respondent's consent was gathered before the study, and personal information (if any) shared during the study will not be disclosed at any stage. The results of the study were made available on request. To validate the patterns, the survey instrument was used to record the data from the participants. More details about the study are as below:

- **Stages and Apparatus**: The study was conducted in two stages. Stage-I involved responses from the participants on the usable security patterns' template. However, Stage-II involved the validation of the Usable Secure Record Inquiry Pattern. In addition to the two stages, the study included briefing the participants about the aims and objectives of the study, the usable security problem, etc. The data was recorded on a questionnaire using a Likert scale (a scale of *five* - '*Strongly Agree*' to '*Strongly Disagree*'). Feedback on characteristics such as understandability, completeness, genericity of the patterns' template was desired during the Stage-I. However, during the Stage-II in addition to the feedback on characteristics just mentioned, for the pattern under consideration, it was intended to have a response on problem-solving ability, comprehension, among others.

Table 3. Details of Participants in the Validation Study

Expert	Background	Company Type	Experience (Number of Years)
1	Software Developer	Software Services	2+
2	Software Developer	Software Services	1
3	Software Developer	Startup	7
4	IT Analyst	Telecom	1+
5	Software Developer	Design and Development	2
6	Software Developer	Software Development	2.5
7	Technical Consultant	Business Intelligence	2.5
8	Software Developer	Software Development	4
9	Software Developer	Software Solution Startup	3
10	Software Developer	Information Technology	6
11	IT Analyst	Information Technology	6
12	Software Developer	Banking Services	7
13	UI/UX Expert	Computer Software Company	8
14	Software Quality Engineer	Software Development	9
15	Information Security Manager	Information Technology	5

- **Training Materials**: The participants were provided with a document including the following: the objectives of the study, set of activities to follow during the study, background information concerning the patterns and usable security, and the problem scenario (for Stage II).
- **Tasks:** After the initial briefing, the participants were asked to fill the Study Questionnaire-I, followed by a briefing on the problem statement and what was intended in Stage II. After the briefing and the question-answer session, the participants were provided the Study Questionnaire II for their responses.
- **Survey Questionnaire and Results**: Two sets of questionnaires were designed, one for each stage. As mentioned earlier, during Stage-I it was desired to have a response on certain qualitative characteristics relevant to patterns' template. The question statements included in the questionnaire for Stage-I are presented in Figure.8.

TASK

Please examine the pattern template before answering the questions. The items in the bullets are elements of the patterns e.g. title, classification, etc. followed by the details of what each item refers to.

Question 1- Understandability
The basic intent behind patterns is to assist developers in solving common design problems. For this purpose, the patterns has to be understandable so that it is not used in a wrong way. Did you find the terms used in the pattern template understandable?

Question 2- Completeness
The template should be detailed enough to let the developer understand the problem, solution, consequences, context of use etc. Did you find the terms used in the pattern template complete?

Question 3- Genericity
Did you find the pattern template generic to be applicable in varying contexts of use and cover different problems in the domain?

Question 4- Providing Common Vocabulary
One advantage of patterns is to provide developers with shared documentation and common vocabulary. Did you find the terms used in the pattern template effective in this regard?

Question 5- Usefulness
Considering specific case of usable security which require expertise in two different field i.e. usability and security, which are hard to find in one person. Do you consider the patterns template useful in a way to assist security developers assess the usability of their security option?

Figure.8. Study Questionnaire-I to record responses on patterns' template

The tendency of participant's opinion to Study Questionnaire-I is presented in Figure.9. Most of the participants found the patterns' template understandable. Similarly, the tendency of responses to questions of completeness, genericity, means of providing a common vocabulary, usefulness respectively is presented in Figure.9.

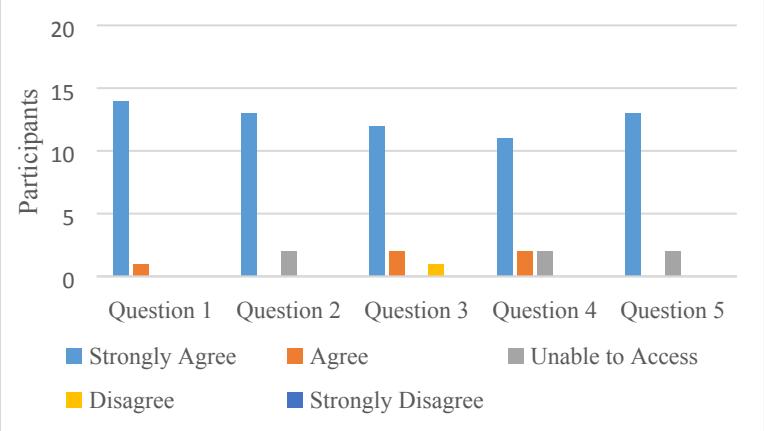

Figure.9. Tendency of the participant's opinion to Study Questionnaire-I

Furthermore, to have responses on the Usable Security Record Inquiry pattern, the Study Questionnaire II (Figure.10) was developed.

TASK

This is the second part of the survey, which is being conducted to validate 'Usable Secure Record Inquiry' pattern. Consider the scenario and the user problem you have been briefed about in the briefing session while answering the questions.

Question 1
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. Do you think this holds true for the pattern under consideration?

Question 2
Patterns describe the thing "what" of the problem and process "how" to solve it. Was this criteria respected in the pattern under consideration?

Question 3
Did you find the problem statement detailed enough to let the developer understand the user's problem?

Question 4

Did you find the context of use clear enough to let the developer apply the pattern in correct context?

Question 5

Do think the solution and discussion would enable the security experts to assess the usability of their security options?

Question 6
Do you think the solution (by the pattern) manages a suitable trade-off between usability and security, without compromising anyone of them?

Question 7
Please respond as per your opinion, do the following characteristics are catered in the pattern under consideration?

I. Understandability
II. Consistency in documentation/template
III. Providing Common Vocabulary
IV. Completeness
V. Usefulness

Figure.10. Study Questionnaire-II to record responses on Usable Secure Record Inquiry

In response to the Study Questionnaire-II, the tendency of participant's opinions is presented in Figure.11. Most respondents believed that multiple implementations could be derived from the solution presented in the Usable Secure Record Inquiry pattern. Furthermore, 12 out of 15 participants 'Strongly Agreed' for the problem statement being enough to let the developers understand the user problem. The similar trend prevailed in terms of clarity of context of use as the `context of use has a crucial role to ensure that patterns are applied in the right scenarios. However, 11 out of 15 participants 'Strongly Agreed' that the patterns managed the trade-off effectively, two participants 'agreed', whereas the other two participants were unable to assess.

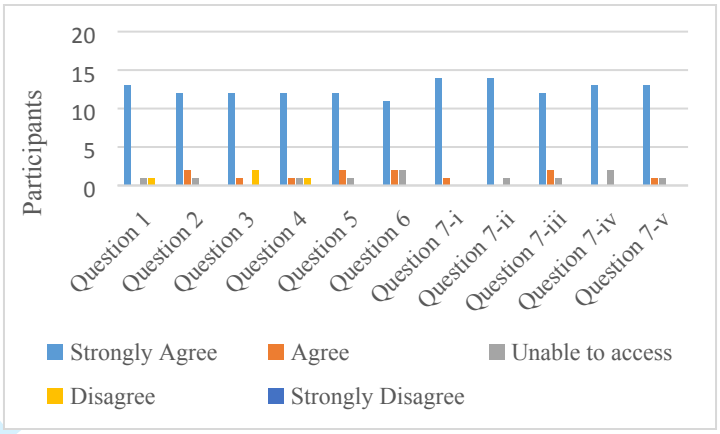

Figure.11. Tendency of the participant's opinion to Study Questionnaire-II

Furthermore, question 7 (I-V) desired response on the characteristics as mentioned during Stage-I of the study. The difference lies in the fact that during the Stage-I the response was recorded for the patterns' template, whereas during Stage-II it was meant for Usable Secure Record Inquiry pattern.

6. Discussion

The IFUS provides a mechanism for identification of the security and usability conflicts right from the start of system development lifecycle. The gaps in state of the art identified after literature review were also considered during the rigor and design cycle of the DSR methodology. As stated earlier, the IFUS is an extension and evolved version of the methodology (Naqvi and Seffah, 2018). Though the fundamentals remain the same but the main limitations of the methodology which have been addressed in IFUS include , grouping of similar activities in line with the practices used in the industry, identification of various roles involved in different activities during various phases, incorporating the elements of risk-based decision making for identification and elicitation of the suitable trade-offs, and, identification of a mechanism for dissemination of the design patterns.

It is pertinent to state that in line with the principles of the DSR method for development of new processes and artifacts, two aspects demonstration, and evaluation must be considered (Peffers *et al.,* 2007). The evaluation was done during the workshop where a group of participants reviewed the IFUS, however, to demonstrate the IFUS we conducted a case study, which is presented in the Section 4. The results of the study were later validated by involving a group of developers. However, some of the strengths and limitations of the IFUS are presented in Table 4.

Table 4. Strengths and limitations of proposed approach

	Strengths
+	Provides a mean to integrate security and usability during the phases of the SDLC
+	Provides means for identification of conflicts and their resolution
+	Supports continuous evaluation and evolution of the artifacts (patterns) based on elements of risk-based approach
+	Provides a communication mechanism for practitioners from independently evolved domains (security and usability) for integrating their concerns in the design process
+	Supports saving costs and efforts associated with re-work in case where conflicts are identified later during the SDLC
	Limitations
-	Does not consider use of metrics and measures for the identification of conflicts
-	Does not provide a severity scale of evaluate degree of the conflict
-	No means for determining the degree of a trade-off, for example the units of usability which are being comprised for security and vice versa.

Moreover, a comparison of the current work with some of the existing frameworks is presented in Table 5. The criteria used for comparison between the works are as below.

- Are security and usability considered together in the proposed solutions?
- Does the proposed approach focus on assisting the developers in handling the security and usability conflicts?
- Does the proposed approach emphasize handling usable security during the early phases of system development?
- Does the proposed approach ensure the adequacy of the usability of security?

Table 5. Comparison of Research Works

Criteria	Researches				
	Flechais *et al*., (2007)	Mairiza and Zowghi., (2010)	Parveen *et al*., (2014)	Hausawi and Allen, (2014)	Current Work
Are security and usability considered together?	X	X	X	X	X
Does the proposed approach focus on assisting the developers in handling the conflicts?					X
Does the proposed approach emphasize handling usable security during the early phases of system development?	X	X	X	X	X
Does the proposed approach ensure the adequacy of the usability of security?	X				X (qualitatively)

IFUS provides an opportunity to handle security and usability concerns from the early phases of the SDLC. Moreover, the documentation of outcomes as patterns also supports the learning aspect thereby assisting the developers in assessing the usability of their security options in similar contexts of use. The adequacy of usability of security is ensured in the solution due to the knowledge and experience of usability experts involved during elicitation of the suitable trade-offs and ensuring compliance to the standards and best practices while eliciting the trade-offs. Therefore, within the scope of IFUS, the adequacy of usable security is ensured from qualitative perspective, however, as a future work there is room for integration of a methodology to ensure the usability of security from a quantitative perspective. Lack of quantitative assessment of the conflicts and trade-offs is one of the limitations of the IFUS. As part of future work, it is intended to add the quantitative aspect to identify, (1) the degree of conflict between security and usability in a particular context; this would require a set of metrics and measures to determine such a degree. Once the degree of conflict is known, it can also be considered as a factor in the prioritization of the requirements associated with potential conflicts. (2) metrics to measure and evaluate the trade-offs when a solution is identified after the risk-based decision process.

7. Conclusion

In this paper, a novel framework (to align security and usability during SDLC) was presented. The exemplar discussed to instantiate the framework features the first effort for aligning security and usability in CPS and smart grids. The results of a study to validate the patterns' template and the patterns are also presented. Referring to some of the previous work as well, we also justified the importance of handling usable security concerns early in the development lifecycle.

Though a case study regarding CPS and smart grids was discussed, this framework holds good for all kinds of systems including information systems, cloud infrastructures. Since usable security requires two sets of expertise, that are generally difficult to have within a single individual, patterns can be seen as a bridge between the security and usability world, between requirement engineering and usability.

Given the proliferation of data-intensive systems, mobile devices, sensors and embedded networks creating new forms of systems where security incidents are more and more trigged by humans, the proposed approach offers up a novel solution to overcome core issues surrounding the development of secure and usable systems. The approach concerning the development of security systems and services needs to evolve from "user is the problem" to "user must be part of security technology-based solution".

References

RSAConference,(2019), "The Future of Companies and Cybersecurity spending" available at: https://www.rsaconference.com/industry-topics/blog/the-future-of-companies-and-cybersecurity-spending (accessed 08 November 2019)

Caputo, D.D., Fleeger, S. L., Sasse, M. A., Ammann, P., Offutt, J., and Deng, L. (2016), "Barriers to Usable Security? Three Organizational Case Studies", *IEEE Security Privacy*, Vol.14 No. 5, pp. 22–32.

Yee K.P. (2004), "Aligning security and usability", *IEEE Secur Privacy*, Vol. 2 No. 5, pp. 48–55.

Parveen, N., Beg, R. and Khan, M.H. (2014) "Integrating security and usability at requirement specification process", *Int. J. Comput. Trends Technology,* Vol. 10 No. 5, pp. 236–240.

Garfinkel, S. and Lipford, H.R. (2014), *Usable Security: History, Themes, and Challenges,* Morgan & Claypool Publishers, USA.

National Institute of Standards and Technology (2016), "Usability and Security Considerations for Public Safety Mobile Authentication", NISTIR- 8080.

IBM (2018), "Cost of Data Breach Study: Global Analysis*", Ponemon Institute LLC*.

Whitten A., Tygar J. (1998), "Usability of security: A case study" Technical Report CMU-CS-98-155, School of Computing Science. Carnegie Mellon University, USA.

Asher B., Kirschnick, N., Sieger, H., Meyer, J., BenOved, A., and Möller, S. (2011), "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*, ACM, pp. 465–473.

Riaz, M. and Williams, L. (2012), "Security requirements patterns: understanding the science behind the art of pattern writing", in *Proceeding of Requirements Patterns* (RePa 2012), IEEE, pp. 29–34.

Riaz, M., Elder, S., Williams, L. (2016), "Systematically Developing Prevention, Detection and Response Patterns for Security requirements", in *Proceedings of Evolving Security & Privacy Requirements Engineering (ESPRE) Workshop held during the 24th IEEE Requirement Engineering Conference Workshops*, IEEE, pp. 62–67.

Alshamari, M. (2016), "A Review of Gaps between Usability and Security/Privacy", *Int. J. Commun. Netw. Syst. Sci*, Vol. 9 No. 10, pp. 413–429.

Morville, P. (2004), "User Experience Design", Semantic Studios, available at: http://semanticstudios.com/user_experience_design/ (accessed 08 November 2019).

Flechais, I., Mascolo, C., and Sasse, M.A. (2007), "Integrating security and usability into the requirements and design process", *International Journal of Electronic Security and Digital Forensics*, Vol.1 No. 1, pp. 12–26.

Glass, B.D., Jenkinson, G., Liu, Y., Sasse, M.A., Stajano, F.A., 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 1st European Workshop on Usable Security*, Internet Society, DOI:10.14722/eurousec.2016.23007

Mairiza, D. and Zowghi, D. (2010), "An Ontological Framework to Manage the Relative Conflicts between Security and Usability Requirements", in *3rd International Workshop on Managing Requirements Knowledge (MARK)*, Sydney, Australia, pp. 1-6.

Wang, W., Lu, Z. (2013), "Cyber security in the Smart Grid: Survey and challenges" *Computer Networks*, Vol. 57, pp. 1344–1371.

National Institute of Standards and Technology (2014), "The Smart Grid Interoperability Panel – Smart Grid Cybersecurity Committee, Guidelines for smart grid cyber security", NISTIR 7628 Revision 1.

Hausawi Y., 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*, pp. 33-44.

Kirlappos I., Sasse M.A. (2014), "What Usable Security Really Means: Trusting and Engaging Users", in *Tryfonas T., Askoxylakis I. (eds) Human Aspects of Information Security, Privacy, and Trust HAS 2014,* Lecture Notes in Computer Science, vol 8533, Springer, pp. 69-78.

Naqvi. B., Seffah, A. (2018), "A Methodology for Aligning Usability and Security in Systems and Services", in *Proceedings of International Conference on Information Systems Engineering* (ICISE) 2018, IEEE, pp. 61-66.

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

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,* Vol 40, pp. 1231-1241.

International Standardization Organization (2011), "Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models", ISO 25010.

Hevener A. (2007), "A Three Cycle View of Design Science Research", *Scandinavian Journal of Information Systems*, Vol. 19, No. 2, pp. 87-92.

Risktec, (2005), "Risk-Based Decision Making", *RISKworld* - the Newsletter of Risktec Solutions Limited, Risktec, Issue 7, Spring 2005, pp. 2-3, available at: https://www.risktec.tuv.com/wp-content/uploads/2018/09/risk-based-decision-making.pdf

Al-Darwiash, A.I, Choe, P. (2019), "A Framework of Information Security Integrated with Human Factors", in *International Conference on Human-Computer Interaction* (HCII) 2019, Springer, pp. 217-229.

Mujinga, M., Eloff., M.M., Kroeze, J.H., (2019), "Towards a framework for online information security application development: A socio-technical approach", *South African Computer Journal*, Vol. 32, No. 1, pp. 24-50.

Feitosa, D., Ampatzoglou, A., Avgeriou, P., 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*, ACM, pp. 113-122.

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

Naqvi, B., Seffah, A., (2019), "Interdependencies, Conflicts and Trade-offs between Security and Usability: Why and how we should engineer them?", in *International Conference on Human-Computer Interaction* (HCII) 2019, Springer, pp. 314-324.

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,* ACM, pp. 35-42.

Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., (2007) "A Design Science Research Methodology for Information Systems Research", *Journal of Management Information Systems,* Vol. 24, No.3, pp. 45-78.

Zagouras. P., Kalloniatis C., and Gritzalis, S., (2017) "Managing User Experience: Usability and Security in a New Era of Software Supremacy", *in T. Tryfonas (Ed.) Human Aspects of Information Security, Privacy and Trust, HAS 2017*, LNCS 10292, pp. 174–188.

Mahlke, S., (2007). "User experience: usability, aesthetics and emotions in human-technology interaction", in *Towards a UX Manifesto, COST294-MAUSE affiliated workshop,* September 2007, Lancaster, UK, pp. 26-30.