Managing User Experience: Usability and Security in a New Era of Software Supremacy

Panagiotis Zagouras[1], Christos Kalloniatis[1(✉)], and Stefanos Gritzalis[2]

[1] Privacy Engineering and Social Informatics Laboratory, Department of Cultural Technology and Communication, University of the Aegean, University Hill, 81100 Mytilene, Greece
{ctdl60l5, chkallon}@aegean.gr
[2] Information and Communication Systems Security Laboratory, Department of Information and Communications Systems Engineering, University of the Aegean, 83200 Samos, Greece
sgritz@aegean.gr

Abstract. Software is now the driving force behind our daily lives. At work, everyone is affected to a greater or lesser extent by software, which exerts a significant influence in every activity of our daily lives. It is clear that software and the way it interacts with humans has a significant impact on the life and future of everyone who uses it. There is thus a self-evident need to balance usability and security, with usability now defined as an outcome of a product's interaction rather than a property inherent to that product, and assessed by means of usability evaluations rather than by measurements. Still, the problems of achieving a balance between usability and security remain. Recent research would indicate that the concept of 'user experience' needs to be broken down into the complementary factors of usability and security to create new methodologies for producing modern, reliable, user-friendly software. The current paper moves into this direction by presenting scientific definitions for the concepts of 'user experience', 'usability' and 'security', their extensions and implications, and the research which has explored ways of harmonizing usability and security in contemporary software. It highlights how hard this is to achieve and how important it is for the software industry to incorporate the concept of 'user experience' into usability-security so as to develop products capable of automatically adapting to any given environment or user.

Keywords: User experience · Usability · Security · Interaction

1 Introduction

The need for user-friendly, high-quality software is now axiomatic. However, a large number of information systems [1] are rejected, despite the large sums invested in their development, due to their failure to interact with the system or fulfill their task. Usability thus has a significant impacts on the success of a given software package.

T. Tryfonas (Ed.): HAS 2017, LNCS 10292, pp. 174–188, 2017.
DOI: 10.1007/978-3-319-58460-7_12

And while it remains a highly complex concept with fuzzy characteristics, it is crucial that we manage usability correctly in every function of a given piece of software. User involvement plays an important role in defining the usability of software and the level of security it can provide when it is operated by a given user.

We live in a world in which privacy and security have assumed greater importance than ever before. In this post-Snowden era [2], companies assign greater significance to data security and as software users, we are increasingly aware of threats to our privacy. And yet we still demand software that is personalizable, user-friendly, intuitive and flawless. As users, we demand 24/7 access to out information without constantly having to log back in to the system.

This is the contradiction between security and user experience (UX). We do not want our data falling into the wrong hands, but we do not want our added security to impact on our user experience, either. In fact, we want ever-easier access to our data.

Everyone has a double-edged relationship [3] with the products and services they make use of: they simplify but complicate our lives; they divide us and bring us together. But all this software is made by people who will take the credit if it works well and the blame if it fails to do so. This means that, in order to design software that provides a better user experience, we need to foresee every action a user may conceivably take and understand their intentions at every stage in every process they execute, all of which must take place in a secure and private environment. Correctly applying UX principles and guidelines will boost security.

This is encapsulated in the following formula:

$$\text{Security} + \text{UX} = \text{Security}^2$$

There is a need to create user-centric information systems. The concept of 'user experience' is crucial to such developments and its implications for usability—security has still to be fully explored by researchers.

This paper illustrates the need to research the concept of user experience in tandem with usability and security, given the lack of scientific methodologies, which consider all three concepts in parallel in order to produce easy to use and secure software meeting contemporary requirements.

Specifically, Sect. 2 provides a thorough analysis of the qualities of user experience-usability-security and their lines of scientific demarcation. Section 3 reviews scientific methodologies, which have been developed to complement usability-security as well as respective tools that have been developed in this direction and provides useful findings. Finally Sect. 4 argues the case for more specific, categorized solutions in the design of secure and usable information systems and concludes the paper and provides guidelines for future research.

2 Core Concepts

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human

or human-to-computer interaction [4]. Yet along with the many societal, environmental and economic benefits of the IoT, the rapidly-expanding connected world represents a growing surface which adversaries of all stripes can attack, and IoT vulnerabilities are being exploited with malicious intent every day.

The IoT (Internet of Things) has changed the way consumers behave in the marketplace. A large number of different devices now interact using a range of technologies. Major corporations are investing heavily in connectable devices. In this context, it is self-evident that the way in which businesses and individuals interact with the IoT will impact significantly on user experience-usability-security.

2.1 User Experience

Recent years have seen vast changes [5] in the systems architecture sector. The data organization and search landscape has changed utterly over the last decade or so, and the nineteen nineties are now most definitely a bygone era. Complex information systems with different technologies, users and goals now interact and exchange personal data and financial figures under regulatory systems of varying strictness, creating obstacles, issues and delays for users. Figure 1 below illustrates why we must strike a unique balance on each project between business goals and context, user needs, behavior and content.

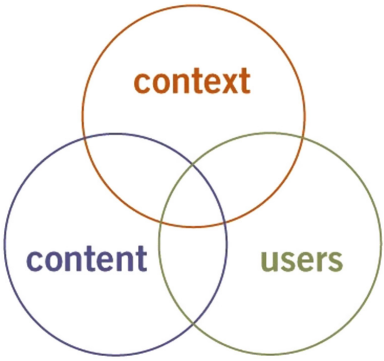

Fig. 1. Three circles of information architecture [5]

The above figure represents the concept of Information Architecture, but it is also useful for understanding user experience as well. If the content has structure and all the information a user needs, it will help create a good user experience [6]. The context refers to the physical, digital, and social structures that surround the point of use.

Users have several characteristics including their age, professional responsibilities, software, hardware, environment (home, shared office, private office, shared public terminal), computer experience and Web experience. User characteristics can also include types of disability, adaptive strategies used, and experience with specific assistive technologies.

User experience has multiple significant extensions, which make it clear that we have to expand into a large number of parameters beyond usability. The following diagram presents this in detail (Fig. 2):

Fig. 2. User experience honeycomb [5]

This is how the facets or qualities of the user experience can be explained [5]:

Useful. Practitioners cannot simply paint within the lines drawn up by managers. They must be brave and creative enough to question the degree to which products and systems are useful and use their expertise in both the craft and the medium to come up with innovative improvements.

Usable. Usability remains crucial, but there are aspects of web design, which go beyond methods of, and perspectives on, human-computer interaction which center on the interface. We can thus say that ease of use, while necessary, is not sufficient in itself.

Desirable. Efficiency is not the be-all-and-end-all of design. It is important not to underestimate the power of emotional design and the value of image, identity, brand etc.

Findable. Users must be able to find what they are looking for on a web site easily, so objects need to be locatable and navigable.

Accessible. Given that people with disabilities account for upwards of 10% of the population, web sites need to ensure they are accessible to this group. It is not only the ethical thing to do, it is good for business, too. E-accessibility is sure to be required by law at some point, in the same way that physical accessibility is now.

Credible. The Web Credibility Project is helping us understand how design elements can impact on the degree to which users believe web content.

Valuable. Site sponsors need to receive value for their money. In the case of not-for-profit organizations, the experience of site users must contribute to the fulfillment of the organization's mission. In the case of for-profit businesses, the on-line experience must enhance customer satisfaction and profits.

UX specialists, designers and developers [7] no longer work in a one-way workflow (or waterfall). Rather, two-way communication is the norm between UX specialists and designers, UX specialists and developers, and developers and designers, given that the

definition, design and development processes can be concurrent. This does, however, create the need for an integrator to coordinate this interaction. The coordinator will thus work closely at different times with UX specialists in realizing the UI architecture (e.g. screen layout), with designers in providing technical support to generate XAML or MXML code, and with developers in ensuring that functionality is integral to the design (Fig. 3).

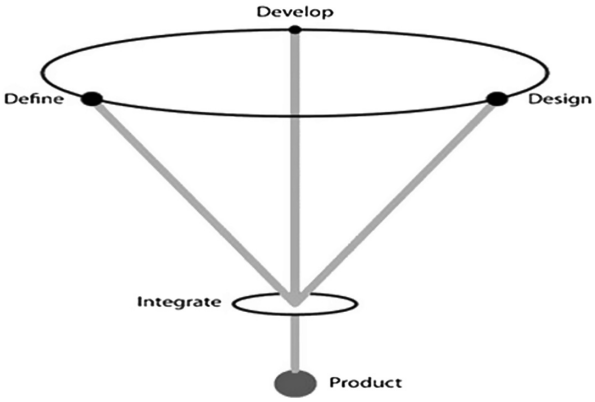

Fig. 3. An illustration of the new UX design and development workflow [7]

Petrie and Bevan [8] found that users of new technologies are less intent on completing a task than on amusing and entertaining themselves. Table 1 illustrates that, rather than being distinct concepts, UX and usability have interrelated aspects that contribute equally to a system's overall UX and usability.

Table 1. Factors contributing to UX [8].

Quality characteristic	UX	Functionality	User interface usability	Learnability	Accessibility	Safety
Product attributes	Aesthetic attributes	Appropriate functions	Good UI design (easy to use)	Learnability attributes	Technical accessibility	Safe and secure design
UX pragmatic do goals	To be effective and efficient					
UX hedonic be goals	Stimulation, identification and evocation					
UX: actual experience	Visceral	Experience of interaction				
Usability (=performance in use measures)	Effectiveness and productivity in use: effective task completion and efficient use of time			Learnability in use: effective and efficient to learn	Accessibility in use: effective and efficient with disabilities	Safety in use: occurrence of unintended consequences
Measures of UX consequences	Satisfaction in use: satisfaction with achieving pragmatic and hedonic goals					
	Pleasure	Likability and comfort				Trust

2.2 Usability

More recently, usability experts [9] have worked with the ISO/IEC JTC1 SC7 Software Engineering subcommittee to integrate usability into software engineering and software quality standards.

Standards relating to usability are primarily concerned with:

- The product in use (the effectiveness, efficiency and satisfaction derived from a particular use).
- The user interface and interaction.
- The process used to develop the product.

The capability of an organization to apply user- centered design.

The above are inter-related: the product's purpose is to be effective, efficient and satisfying when it is used to achieve the desired result. That it has a suitable interface and interaction is a prerequisite for all of these. This requires a user-centered design process, which, if it is to be consistent, requires an organizational capability to support user-centered design.

The most challenging aspect of software development is not simply providing the required functionality; it is fulfilling specific properties such as performance, security and maintainability, which contribute to software quality [10].

Usability engineering has several benefits. Specifically:

- It improves software
- It saves customers money
- It minimizes engineering costs

Proper usability engineering leads to software that is usable, which translates itself into productive, satisfied customers, a better reputation for the product, and hence increased sales. Proper usability engineering can reduce the cost overruns in software projects.

A framework is presented which visualizes how and to what extent usability can be integrated at the architectural level using specific methods of design, and how and to what extent we can assess architectures in terms of the degree to which they support usability. Usability should drive design at all stages, but current usability engineering practices fail to fully achieve this goal. Our survey shows that there are no design techniques or assessment tools that allow for usability to be integrated at the architectural level.

Figure 4 below illustrates an integrated approach to the extensions/implications of usability.

2.3 Information and Computer Security

Information security [11] is described as a set of properties that must be upheld.

The ISO/IEC 27000:2016 [12] provides an overview of information security management systems and describes terms and definitions.

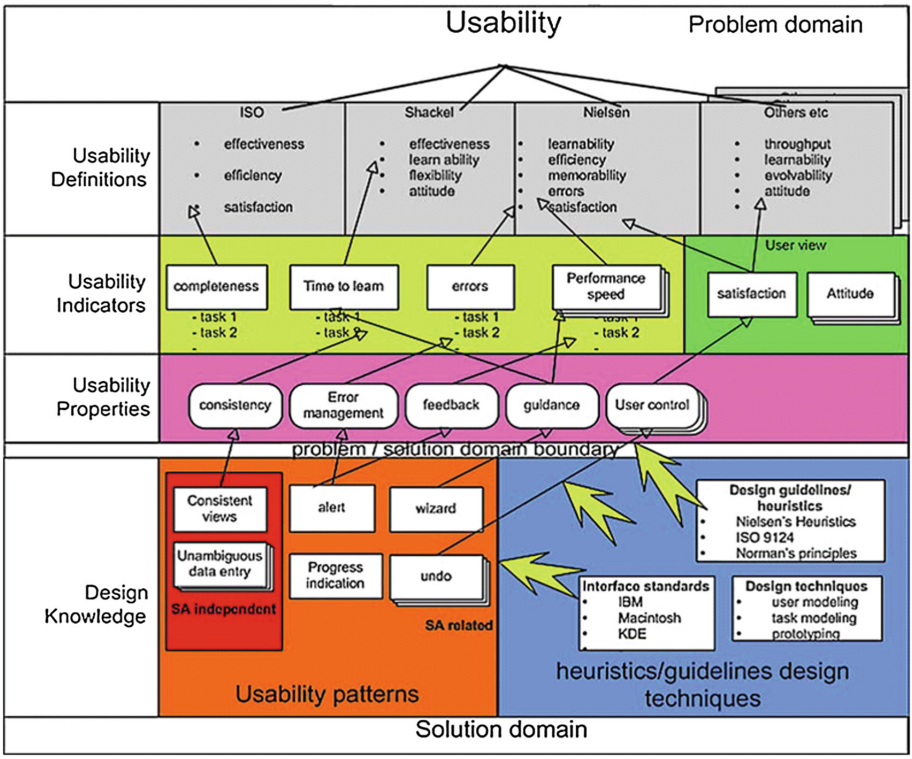

Fig. 4. A usability framework approach [10]

Computer Security can be defined [13] as the technological and managerial procedures applied to computer systems to ensure the availability, integrity and confidentiality of the information managed by the computer system.

Confidentiality, integrity and availability—or CIA–are the fundamental elements of any IT security system. However, their inclusion often detracts from the usability of IT systems. Other researchers have offered more complex variations on the classic CIA triad; Don Parker [14], for instance, has written about the Parkerian Hexad of Confidentiality, Integrity, Availability, Possession, Authenticity and Utility. Other scientists have proposed other desirable properties.

Gollman [15] proposed accountability and dependability. He defined security as the protection of assets and introduced the concepts of prevention, detection and reaction. Two equally important concepts relevant to security are avoidance and deterrence.

A threat is something that has the potential to cause us harm. Vulnerabilities are weaknesses that can be exploited in order to harm us. Risk is the likelihood that something bad will happen.

The table below lists the potential consequences of various types of threats to an information system's security attributes [16]. An interception means that some

unauthorized party has gained access to an asset. In an interruption, an asset of the system becomes lost, unavailable, or unusable. If an unauthorized party not only accesses but tampers with an asset, the threat is a modification. Finally, an unauthorized party might fabricate counterfeit objects on a computing system (Table 2).

Table 2. Various types of threats

Security attribute	Threats
Confidentiality	Interception
Integrity	Interruption Modification Fabrication
Availability	Interruption Modification Fabrication

There are various ways of building increased security into information systems. These break down into:

- Descriptive and *ad hoc* methods.
- Checklists.
- Guidelines.
- Risk management.

Providing security for, but also against, the different people around our information is one of the most difficult aspects of information security. These can include service providers, employees, partners, contractors, customers and many others. We can expect all these groups to behave in different unforeseen and unexpected ways, doing so innocently, ignorantly or maliciously. In all cases, it can be a challenge to provide security in this area. Given that humans are the weakest link, they must be taught to be more aware of security.

Applying design patterns can have multiple benefits in the security sphere. The seven security patterns as shown in Fig. 5 and proposed by Yoder and Barcalow [17] can be applied when developing security for an application.

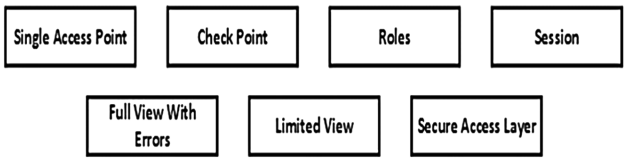

Fig. 5. Yoder and Barcalow's seven security patterns [17].

This is an excellent approach to take into consideration in this research, given that it considers an application to be only as secure as its component parts and its interactions with them. It is a user-focused approach.

3 Academic Approaches

It is extremely difficult to harmonize security with usability. The fundamental goal of security is to protect assets. But protecting infrastructure from risks requires security systems to access these risks. Moreover, it should be noted that security is a process not a product.

3.1 Usable—Security

HCISec is a scientific field which has advanced enormously in recent years. A number of research projects have focused on the usability of password policies and security controls. The HCISec community has approached the design of usable-security from two different directions:

- Design principles and idioms.

Examples include Yee's [18] guidelines and strategies for secure interaction as well as Garfinkel's [19] patterns for usable-security.

- User-centered security

Sasse et al's [20] work on applying HCI design approaches to the design of security mechanisms and Zurko and Simon's [21] work on user-centered security contain significant material.

Very little research has been conducted into the design of usable and secure systems with a view to the designer's, rather than the end user's, needs. Composing HCI with security technics is a development that has inspired a good many researchers.

Gerd tom Markotten's research [22] seeks to connect the security engineering process to usability engineering. It begins with functional analysis, threat and risk analysis, a security strategy and model, and the design and implementation of testing. This corresponds to Analysis, Design, Testing in usability engineering.

AEGIS [23] (Appropriate and Effective Guidance for Information Security) is another example. AEGIS was designed as a lightweight process to provide guidance to developers designing secure systems. AEGIS assets and their relationships are modeled using UML [24]. AEGIS asset models make useful boundary objects, while asset modeling and risk analysis are carried out with respect to different environments.

The lack of a design approach based on the singularity and requirements of individual users and compatible with current scientific approaches to usability and security is obvious. Procedures need to be created which allow software to be designed in line with these singular needs, while sticking to the rules of usability-security.

3.2 Usability—Design

The ISO 9241 on usability definitions is detailed in Table 3 below.

Table 3. Usability Definitions based on ISO 9241

Concept	Description
Product	The part of the equipment (hardware, software, materials) for which usability is to be specified or evaluated
User	Person who interacts with the product
Goal	Intended outcome
Effectiveness	Accuracy and completeness with which users achieve specified goals
Efficiency	Resources expended in relation to the accuracy and completeness with which users achieve goals
Satisfaction	Freedom from discomfort, and positive attitudes towards to the use of the product
Context of use	Users, tasks, equipment (hardware, software and materials) and the physical and social environments in which a product is used

A number of usability professionals have taken the philosophy of user-centered design on board and have created various usability design processes informed by this philosophy, such as: Goal-Directed-Design [25], Contextual Design [26] and Usage Centered Design [27]. The above processes share the following features:

- Tasks and scenarios
- Goals
- Personas and Assumption personas

We have still to produce modern, user-friendly software, and there is a pressing need to adopt a contemporary approach informed by the new balance of priorities in Information Technology.

3.3 Requirements Engineering

Requirements engineering [23] is a research nexus between HCI and information security. It encompasses many approaches, but the best fits for this area are Problem Frames, Goal-Oriented Approaches, and Use Cases. These approaches are valuable in this area, because they have published security extensions and relate to the elicitation and specification of requirements.

Problem Frames: A tool [28] for structuring software problems and analyzing them.

Problem analysis or the problem frames approach is a set of concepts which can be employed when collating requirements and deciding on specifications for software. Its underlying philosophy differs markedly from other methods of collecting software requirements in so far as it:

- Takes a parallel rather than a hierarchical approach to breaking down user requirements.
- Views user requirements as real-world relationships–which is to say in the application domain, rather than within the software system itself or its interface.

Goal-oriented approaches: Goal-Oriented Requirements Engineering (GORE) [29] is about the use of goals for requirements evaluation, elicitation, documentation, quality assurance and evolution. Two goal-oriented RE Frameworks emerged independently for GORE, KAOS [30] and NFR/i* (Non-Functional Requirements/Intention Strategic Actor Relations) [31]. Both frameworks address common targets such as goal refinement and conflicts, but while there are complementarities there are also differences between them. KAOS was more focused on semi-formal and formal reasoning about behavioral goals for deriving goal refinements, goal operationalization, goal-based risk analysis and conflict management. In NFR/i*, too, the focus was more on qualitative reasoning and soft goals for analyzing goal contributions, evaluating alternative goal refinements, and reasoning about goal dependencies among organizational agents.

Use Cases: Scenario-based approaches to specifying, validating and eliciting are popular in Requirements Engineering. The best-known approaches are Use Cases [32]. Sindre and Opdahl [33] proposed Misuse Cases, a sequence of actions including variants that a system or entity can perform, interacting and causing harm to stakeholders.

Castro et al. propose yet another approach, which combines Usability with Requirements Engineering [34]. In order to take usability into account at early stages of software development, he adds various new activities: relating behavior patterns to usability mechanisms, building use cases with usability mechanisms, and building mock-ups with usability mechanisms. The activities that gained the most were the elicitation and analysis of requirements relating to user knowledge and user modeling respectively.

The above approaches relate to the users' goals and knowledge, but do not return the expected results, primarily because neither the individual user nor their behavioral characteristics have as yet been properly and fully researched.

3.4 Methodologies—Frameworks

The community adopts social science research methodologies to examine the difficult issue of secure information system research. The two basic methodologies are (i) Action Research [35, 36] and (ii) Grounded Theory [37, 38].

The Action Research approach has a five-phase process:

- Diagnosing
- Action Planning
- Action Taking
- Evaluating
- Specifying learning

Grounded Theory has three basic stages:

- Open Coding
- Axial Coding
- Selective Coding

Hausawi [39] proposed the Usable-Security Engineering Framework (USEF), which consists of three components (Assessment Framework for Usable Security—AFUS, Usable-Security Guidelines, and a Usable Security Measuring Matrix). Each component focuses on one of the three phases (Requirements Engineering, Design, and Evaluation/Testing) of the Software Engineering Development Life-Cycle (SDLC) in order to enhance the alignment of security and usability for better usable-security (Fig. 6).

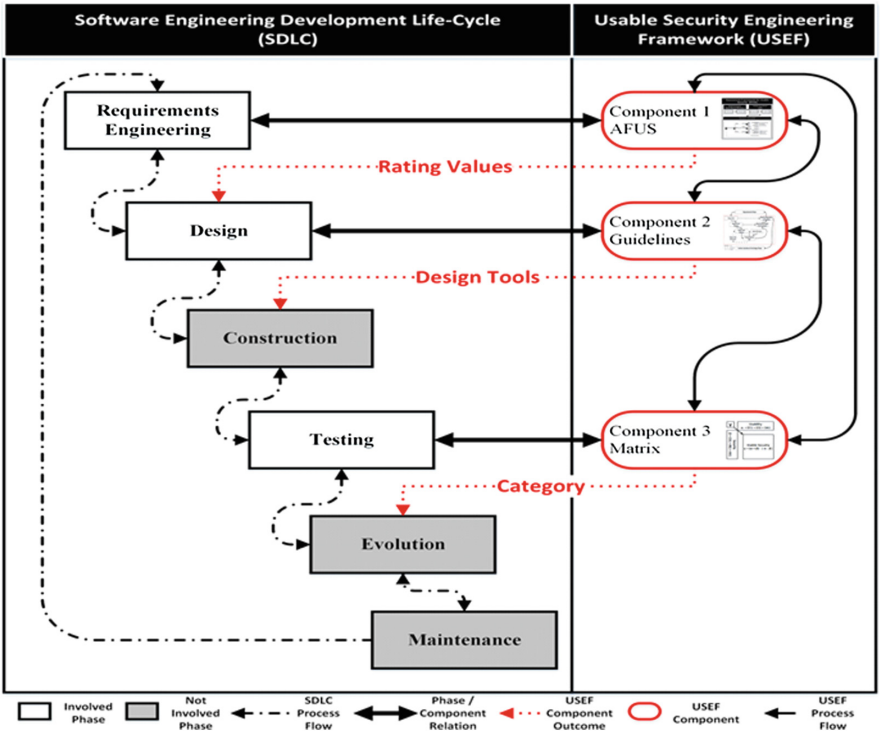

Fig. 6. The Usable-Security Engineering Framework (USEF) [39]

Faily [23] proposed the IRIS (Integrating Requirements and Information Security) framework as a paradigm for integrating existing techniques and tools with the design of usable and secure systems. The IRIS medal model is sub-divided into six views—environment, asset, task, goal, risk and responsibility—which correspond with the different perspectives associated with a secure system's use contexts. Each view is modelled using a UML class diagram.

The meta-model facilitates the specification of requirements for usable and secure systems by stipulating the concepts that need to be elicited. However, the meta-model is agnostic about how this is to be done, which makes it necessary to provide guidance on how the concepts should be elicited and specified. To this end, we have broken the meta-model up into three intersecting groups (which Faily calls perspectives): Usability, Requirements and Security.

The same researcher has also developed CAIRIS (Computer Aided Integration of Requirements and Information Security), a software tool which embodies the characteristics required to support the IRIS framework. The design principles of CAIRIS are: Familiarity, Extensibility, Process, Centricity, Security and Usability centricity.

These two approaches deliver excellent results in the management of usability and security, but there would still appear to be considerable scope for further research into how user experience relates to the characteristics in question, given that the user now plays a key role both during the development of modern software and after its release in determining its success or otherwise.

4 Conclusions

Our research presented the approaches taken by the academic community to the design of information systems that are both user-friendly and secure. It became clear that none of the approaches presented actually took into account the profile of the users who would be using the system—i.e. their knowledge, weaknesses and the environment with which they interact. Thus, while user-friendly security is essential, none of the proposals use detailed user features as input into the process for designing useable security. Usable design seeks to achieve its goals in various ways, but we believe there is a need for more specific, categorized solutions. Requirements engineering has not borne fruit, because researcher have still to thoroughly and properly research the individual user and their behavioural characteristics.

Our research has highlighted the lack of interaction between User Experience, Usability and Security. It must be understood that User Experience and Usability are two different concepts. Usability is a subset of user experience. The community has focused in the past on the relationship between usability and security, ignoring the user who interacts with the program and influences the way in which it behaves through their behavior and use experience. This could be captured in the study on User Experience. The need for investigating and testing scientific routes that could help the community develop software building methodologies that will interact securely with both the mind and the mood of every user is more than immense.

We believe that the future of software engineering lies in the creation of software whose nature adapts to every context and user it finds itself in. The need for establishing rules and procedures that will enable everyone involved in software development to work together and create safe and user-friendly software which will draw its character from the environment in which it is operating, respect every user, and interact in a different way with different users depending on their needs and requirements is a goal that the new technological trends along with users' expertise demand to be implemented as soon as possible.

References

1. Ankita M., et al.: Usability evaluation methods: a literature review. Int. J. Eng. Sci. Technol. **4**(2) (2012)
2. Jasper, O.: Security + Usability = Security^2 (2015). http://www.thenextview.nl/blog/security-user-experience. Accessed 11 Jan 2017
3. Garret, J.J.: The Elements of User Experience: User-Centered Design for the Web and Beyond. New Riders, Berkeley (2011)
4. http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT
5. Morville, P.: User Experience Design (2004). http://semanticstudios.com/user_experience_design/. Accessed 11 Jan 2017
6. Techved: (2016). https://uxmag.com/articles/the-user-experience-of-good-content. Accessed 15 Jan 2017
7. Zheng, X.S., Wang, M., Matos, G., Zhang, S.: Streamlining user experience design and development: roles, tasks and workflow of applying rich application technologies. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part I, HCII 2011. LNCS, vol. 6761, pp. 142–151. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21602-2_17
8. Petrie, H., Bevan, N.: The evaluation of accessibility, usability and user experience. In: Stephanidis, C. (ed.) The Universal Access Handbook, pp. 20.1–20.14. CRC Press, Boca Raton (2009)
9. Bevan, N.: International Standards for HCI. Idea Group Publishing, Hershey (2006)
10. Folmer, E., Bosch, J.: Architecting for usability: a survey. J. Syst. Softw. Issue **70**(1) (2002)
11. Fléchais, I.: Designing secure and usable systems. Ph.D. thesis, University College London (2005)
12. ISO: ISO/IEC 27000:2016 Information technology – Security techniques - Information security management systems - Overview and vocabulary. ISO/IEC (2016)
13. http://www.digitalguards.com/glossary.php
14. Parker, D.B.: Fighting Computer Crime. Wiley, Hoboken (1998)
15. Gollman, D.: Computer Security. Wiley, Hoboken (1999)
16. Andress, J.: The basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice. Elsevier, Amsterdam (2014)
17. Yoder, J., Barcalow, J.: Architectural patterns for enabling application security. Urbana **51**, 61801 (1998)
18. Yee, K.-P.: Guidelines and strategies for secure interaction design. In: Cranor, L.F., Garfinkel, S. (eds.) Security and Usability: Designing Secure Systems that People Can Use. O'Reilly Media, Newton (2005)
19. Garfinkel, S.L.: Design principles and patterns for computer systems that are simultaneously secure and usable. Ph.D. thesis, Cambridge (2005). (Adviser-David D. Clark and Adviser-Robert C. Miller)
20. Sasse, M.A., et al.: Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. BT Technol. J. **19**(3), 122–131 (2001)
21. Zurko, M.E., Simon, R.T.: User-centered security. In: Proceedings of the 1996 New Security Paradigms Workshop, pp. 27–33 (1996)
22. tom Markotten, D.G.: User-centered security engineering. In: Proceedings of the 4th EurOpen/USENIX Conference (2002). Unpublished workshop proceedings
23. Faily, S.: A framework for usable and secure system design. Ph.D. thesis, University of Oxford (2011)
24. Rumbaugh, J., et al.: The Unified Modeling Language Reference Manual, 2nd edn. Addison-Wesley, Boston (2005)

25. Cooper, A., et al.: About Face 3: The Essentials of Interaction Design. Wiley, Hoboken (2007)
26. Beyer, H., Holtzblatt, K.: Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, Burlington (1998)
27. Constantine, L.L., Lockwood, L.A.D.: Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. Addison-Wesley, Boston (1999)
28. Jackson, M.: Problem Frames: Analyzing and Structuring Software Development Problems. Addison-Weslay, Boston (2001)
29. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, Hoboken (2009)
30. Dardenne, A., et al.: Goal-directed requirements acquisition. Sci. Comput. Program. **20**(1–2), 3–50 (1993)
31. Chung, L., et al.: Non-functional Requirements in Software Engineering. Kluwer Academic, Dordrecht (2000)
32. Jacobson, I.: Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Boston (1992)
33. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requir. Eng. **10** (1), 34–44 (2005)
34. Castro, J.W., et al.: Integrating the personas technique into the requirements analysis activity. In: Proceedings of the 2008 Mexican International Conference on Computer Science, pp. 104–112. IEEE Computer Society (2008)
35. Baskerville, R.: Investigating informations systems with action research. Commun. AIS **2** (1999)
36. Stringer, E.T.: Action Research. SAGE Publications, Thousand Oaks (2007)
37. Martin, P.Y., Turner, B.A.: Grounded theory and organizational research. J. Appl. Behav. Sci. **22**(2), 141–157 (1986)
38. Goulding, C.: Grounded Theory: A Practical Guide for Management, Business and Market Researchers. SAGE Publications, Thousand Oaks (2002)
39. Hausawi, Y.: Towards a Usable-Security Engineering Framework for Enhancing Software Development. Florida Institute of Technology (2015)