2016

Misalignment challenges when integrating security requirements into mobile banking application development

Memory Machiridza
University of Cape Town, mchmem001@myuct.ac.za

46. Misalignment challenges when integrating security requirements into mobile banking application development

Memory Machiridza
University of Cape Town
mchmem001@myuct.ac.za

Abstract

This study identifies and explores the core challenge faced when integrating security requirements into the mobile application software development life cycle. Studies on key issues in Information Systems (IS) have been on-going in the past decades, with security moving up the ranks of top issues in IS. Security requirements can be added into mobile application development processes by practising secure coding or by adding a third party security tool. This study gathered data from a single case study and employs grounded theory methodology to reveal misalignment as the core challenge to integrating security requirements into mobile banking application development. Identified forms of misalignment include that between security requirements and (1) external entities, (2) roles, (3) skills and (4) system requirements. Some of the findings indicate the need for further research. Research indicates that mobile application development follows agile methods for development. Agile methods have been compared with Complex Adaptive Systems (CAS). For this reason, research in IS could benefit from studies that focus on CAS as a theory to provide a better explanation on the misalignment issues in mobile application development. From the current study, the research also identified the need to address misalignment issues before embarking on a project involving integrating of security requirements.

Keywords

Security, Mobile Application Development, Misalignment

1. Introduction

Mobile banking has grown rapidly in the last decade, with more banking institutions investing towards mobile banking technologies. Mobile banking is defined as the provision of banking services via a mobile device (Angelakopoulos & Mihiotis, 2011). There are issues that affect the full potential use of mobile banking. Issues around security have been identified as major factors prohibiting the full adoption of mobile banking (Angelakopoulos & Mihiotis, 2011). Angelakopoulos and Mihiotis (2011) state that "people see and hear everywhere about hackers, crackers, computer virus, identity theft, phishing attacks, spyware, malware and many other terms that refer to security issues". Operating systems running on most smartphones and tablets are almost as advanced as the operating systems on desktop computers. This makes smart devices as susceptible to security attacks such as hackers, viruses, spyware and other security issues as much as desktop computers (Bickford, O'Hare, Baliga, Ganapathy & Iftode, 2010). Software developers, therefore, need to take cognizance of these issues in their software development processes.

Software development methodologies are a set of guidelines that are followed during the software development process. It is important to be aware of the software development methods and how they fit in the mobile application development process. There are two main categories of software development methodologies namely traditional development methodologies and agile methods. Traditional software development methodologies are founded on following a series of sequential steps from the requirements gathering to the maintenance of the software product. The most common ones include the waterfall model, spiral model and the unified model. On the other hand, agile methods are some of the most important advancements in rapid software development methodologies realised in the last few years. Their main goal is to address the limitations of the traditional software methodologies. Agile methods focus on quick response to the customer requirements. The more popular agile methods are scrum and extreme programming (XP) (Pikkarainen, Haikara, Salo, Abrahamsson & Still, 2008).

Schadler and McCarthy (2012) provide a comparison of the 'PC era' and the 'mobile age'. The preferred development methodologies of these two periods are different. The 'PC era' is distinguished by its use of the waterfall development method while the 'mobile age' employs agile methods. A number of factors have been identified as the driving force behind the adoption of agile methodology in the development of mobile applications, for example, competitiveness in the market, shorter delivery cycles and the ever changing customer requirements (Pikkarainen et al., 2008). Agile methods have been found as appropriate for mobile application development, but not much research has addressed issues related to security requirements and their integration into agile software development (Abrahamsson, 2007).

McGraw (2006) defined software security as "…building secure software: designing software to be secure, making sure that software is secure, and educating software developers, architects, and users about how to build secure thing". The securing of software is about building secure software. This includes ensuring correct coding standards and following prescribed standards and guidelines. When developing software, it is important to ensure that the whole development process complies with both internal and external security policies (Oueslati, Rahman & Othmane, 2015). Software security should be a consideration from the early phases of the development lifecycle as issues that are undetected may become apparent later in the development cycle (Daud, 2010).

Security requirements are major concerns when developing mobile banking applications. Security has been identified as one of the top issues and influences the quality and usability of an application (Daud, 2010). Therefore, it is important to look at the challenges faced in integrating security into the mobile application development process and how these challenges can be addressed. The key research question posed in this study is:

- What is the core challenge faced when integrating security requirements into mobile banking native applications?

An inductive grounded theory methodology was employed to reveal the core challenge. Such an approach is appropriate in addressing a broad, open-ended question such as this. Literature review around this core challenge was only possible after the core challenge had emerged

through data analysis; hence, the literature concerning this challenge is weaved into the discussion of the findings after the fashion of Volkoff and Strong (2010).

2. Research Methodology

The study adopts a form of the grounded theory methodology often termed Glaserian or classical (Matavire & Brown, 2013). Grounded theory methodology is a composition of techniques that involve data collection and data analysis simultaneously. The researcher focuses more on data collection and analysis (Pickard, 2007). Grounded theory methodology follows three set principles; emergence, constant comparative analysis and theoretical sampling. The principle of emergence involves the researcher(s) having no theoretical framework as a study lens. Belief is that the research process and the research product should unfold during the research (Matavire & Brown, 2013). Unlike deductive methods which commence research with a predefined theory and collecting and analysing data based on that theory, grounded theory methodology is used to collect data and generate theory from the data (Matavire & Brown, 2013). However, it is worth mentioning that grounded theory methodology is not always used for theory generation. In some cases, it can be used as a foundation study for a more extensive project in order to gain initial knowledge (McCallin, 2003).

In the current study, a single case study involving a team that is within a software development organization was employed. The team is involved in developing mobile applications for the retail banking sector. Theoretical saturation was reached with thirteen participants, i.e., after thirteen interviews all major conceptual categories identified were sufficiently supported by the data. Data collection and analysis was performed in an iterative manner. After each interview, the researcher made notes on possible concepts emerging from the data. From the transcription of the interviews and the notes, the data was broken down and relevant parts of the data were given initial labels using the process of open coding (Glaser & Strauss, 1967). The researcher documented thoughts on the concepts that were emerging in the form of memos. When the process of coding had been completed, the results of the data analysis were documented in a spreadsheet. The column which contained the answers to the question ‘What concepts does this incident indicate?’ had over 30 open codes. In some cases, codes such as “complexity”, “misalignment” and “ignorance in users” were taken from the words of the participants. Some of the categories that emerged during the data analyses which were initially thought to be important to the study but were later discarded included supportability, project scheduling and profitability due to less than three participants making mention of them. Selective coding, a process of limiting coding to only the concepts around the core category (Glaser & Strauss, 1967), proceeded once the data revealed the concept of ***Misalignment*** as the core category. Through this process, different forms of misalignment were identified. According to the Oxford dictionary, misalignment is “the incorrect arrangement or position of something in relation to something else”. Table 1 gives definitions for the major forms of misalignment identified.

Table 2 is a representation on the popularity of the categories that were identified as forms of misalignment. The second column indicates the number of data incidents related to the category and the third column the number of interviewees who mentioned the data incidents. External misalignment had the highest frequency, with twelve of the participants indicating external entities such as customer requirements, standards and guidelines, regulatory requirements and third party applications as challenges to the integrating of security requirements. Requirements

misalignment had the second highest frequency followed by skills misalignment and finally, role misalignment with four of the participants indicating the differences in roles as challenges to integrating security requirements to mobile banking applications.

Category	Definition
External Misalignment	External misalignment occurs when the software development processes conflict with any other elements that are eternal and out of the control of the development team such as the customers, regulations and third party applications
Role Misalignment	Role misalignment occurs between specific roles such as developer and tester misalignment
Skills Misalignment	Skills misalignment occurs when the current skills do not match the required workload leading to mismatch in responsibilities and incorrect implementation
Requirements Misalignment	Requirements misalignment occurs when there are conflicting issues between the security requirements and the general system requirements

Table 1: Misalignment Categories

Category	Occurrence	Participants
External Misalignment	49	12
Role Misalignment	6	4
Skills Misalignment	28	11
Requirements Misalignment	32	8

Table 2: Misalignment Categories Statistics

3. Discussion

To understand challenges faced during the integration of security requirements, interviews were conducted with business analysts (BAs), developers, testers and a project manager. The results indicate the concept of misalignment as a core challenge to the integration of security requirements into the mobile application development lifecycle. Forms of misalignment identified in the study include; external, role, skills and requirements. In this section, the identified forms of misalignment will be discussed in relation to literature within the context of integrating security requirements to mobile application development.

Misalignment

Misalignment arises when the intended purpose or design is somewhat conflicting with the real outcome. The concept of alignment in IS has been explored especially in IT-Business alignment (Chan & Reich, 2007; Luftman & Kempaiah, 2007). The concept of alignment has also been investigated in software development to address issues around alignment between development and testing (Zhang, Stafford, Dhaliwal, Gillenson & Moeller, 2014). The concept of alignment, especially in IT is complex as it is quite fragmented and relates to different facets. Hence, in order to achieve appropriate alignment, it is important to ensure "focus is on specific components of alignment rather than on the overall alignment" (Dhaliwal, Onita, Poston, & Zhang, 2011). For this reason, the lack of alignment which is referred to in this study as misalignment, is discussed in the context of firstly; external entities such as customers, standards and guidelines, regulations and third party software; the different roles involved in the software development process; the current and required skills for integrating security requirements and lastly the general system requirements. All the identified forms of misalignment pose as challenges to the

integration of security requirements in mobile application development. The section that follows gives an overview of the different forms alignment.

External Misalignment

In the section that follows, the discussion will be around the four aspects that make up external misalignment namely customer requirements, standards and guidelines, regulatory requirements and lastly, third party software.

Customer Requirements

The data analysis reveals the extent to which customer requirements drive the software development process. For the BA, the important subject is ensuring customer satisfaction, however, still maintaining the quality and credibility of the software. Both the BAs and the developers indicated the need to focus on customer needs and preferences when adding security features. However, it is clear from the data analysis that customer preferences can result in security vulnerabilities. In the study, one example which can be used to illustrate this aspect relates to a customer requesting the introduction of web banners inside the mobile banking application as a means to advertise the other products offered by the customer.

"Customers wanted web views but this is a security issue" PARTICIPANT 1, DEVELOPER

Regardless of the advancements in technology, security vulnerabilities are still at large due to the "influence of people" (Lacey, 2011). Security practices should not only be within the organisation's domain but should extend to external entities such as customers (Lacey, 2011). In a study carried out by Zhu (2015), it was noted that customers are not concerned or familiar with security technologies and possible threats. Although Zhu (2015) primarily focused on customer security awareness on Internet banking, the same principles can be applied to mobile banking security awareness as both channels access banking via the Internet. Customers may not be aware of possible security threats and vulnerabilities arising from requested requirements such as the need for advertisement links inside a mobile banking app.

Standards and guidelines

External misalignment can also occur in terms of variability in security guidelines and standards. It is important to note that security in mobile apps can be achieved by an additional tool provided by a third party company that specialises in security, by building security components in-house such as authentication or by implementing both security mechanisms. According to the BAs, there is a lack of set guidelines available on selecting a security vendor. BAs indicated that there was a lack of guidelines for selecting a security vendor both internally and externally by the banking institutions. The first BA who was involved in acquiring a third party tool for device security stated that:

"*There are no clear guidelines on what vendors they should select and support. So certainly within the industry, there is work that should be done to identify which vendor someone [organisation] would choose"* PARTICIPANT 11, BUSINESS ANALYST

According to Lacey (2011), "It's vital also to ensure that project managers and development staff appreciate the importance of developing secure systems, based on intrinsically secure protocols

and coding standards". Standards and guidelines are a form of security requirements and help in realising the overall security of software (Rindell, Hyrynsalmi & Leppänen, 2015). Existing security guidelines prove to have some misalignment among them. In addition to this, use of guidelines and standards indicate some form of process following, which in the case of an organisation following agile methods might be difficult as agile methods follow an informal approach to working (Rindell et al., 2015).

Regulatory requirements

The data analysis indicated complexities around the understanding of government regulations that relate to security of information. In addition, one may need to consider regulations from different countries as software development is global. A company may be providing software to customers in different countries, with different laws and regulations. According to one developer, several regulations are in place which makes following and aligning them the to the development process difficult.

"Many countries have different regulations from others" PARTICIPANT 4, DEVELOPER

Thus, it is important for the stakeholders involved to be aware of the important regulations for the countries the mobile apps will be deployed and mitigate any issues that may result in violating any of these rules and regulations. Governments are expected to impose laws and regulation governing the security of personal data. One of the challenges faced when integrating security requirements for smartphones is the lack of security policies. Most, if not all applications running on the smartphones require the use of the Internet. The Internet is borderless, which makes the formulation of security policies challenging. Additionally, the formation of comprehensible security policies by any government is difficult as there are no frameworks available. Furthermore, most governments are not well equipped to deal with security issues (Harknett & Stever, 2011).

Third-party software

The use of third party applications brings several alignment challenges in the software development lifecycle. In this case study, the involved organisation integrated a third party security application as one of the means of ensuring the security of the mobile banking application. Challenges manifested during the processes of integrating a third party security application to the mobile banking application. One problem as mentioned by one of the developers was the misalignment that resulted due to the conflicting internal security policies and regulations with those of the security vendor.

"Understanding how to use the third party application in such a way that it does not violate our privacy and security requirements" PARTICIPANT 1, DEVELOPER

The results of the study indicate concerns around the use of third-party software such as issues around the lack of control. These include aspects such as limited or no access to the source code and working with the unknown. Third-party applications are ready-made purchased external software components that are used in software development with the aim of improving the software quality and reducing the cost and cycle of software development (Haddox, Kapfhammer, Colyer & Tsai, 2009). Haddox et al., (2009) identified challenges to integrating

third-party software. Firstly, the recipients of the third party software in most cases do not have control over the source code. In situations where the source code is available, the behaviour of the applications is unknown thus; there is a limited control on the outcome of integrating a third party application. De Jonge, (2009) in support of this view, states that integrating third party software with software built in-house is a challenge as most software built in-house is not standardised and "third-party software does not fit..." (De Jonge, 2009).

Role Misalignment

Role misalignment occurs between different specific roles. The roles found in an agile team are easily distinguishable yet connected. Typically, in a scrum environment, because of the augmented team collaboration, there is needed to understand tasks performed by other roles. This will enable one to identify where they fit in on the team and what each team member needs to do to be able to complement the other roles. One developer noted that:

"The first thing is if you do not have good alignment on people doing the research and people who want to code, you are going to miss stuff" PARTICIPANT 4, DEVELOPER

Any clarification on requirements should be performed by the BA and not the developer. This can result in errors as the developer will only explain the requirements from what was developed instead of what was documented in the requirements documents. One developer stated the need to avoid such misalignment by working collaboratively. The different roles that make up a team must be properly defined and aligned, with each role performing the expected tasks. From the data analysis, there was an indication of misalignment of roles such as the BA and the tester, the BA and the developer and the developer and tester. A clear indication of what each role entails should be specified for each project through effectively communicating to the respective roles. This will ensure that each role performs the expected tasks without assuming dependency on another role (Dhaliwal et al., 2011). In software development, coordination of work processes is determined mainly by the fragmented roles that make up a software development team.

Typically, roles within a software development team include software developers, testers, business analysts, project managers, security engineers and IT managers. Dhaliwal et al., (2011) refer to these roles as an internal IT subunit. Prior research indicates misalignment within the internal IT unit, especially between software developers, testers and the business/ requirements analyst (Ghobadi & Mathiassen, 2015; Zhang et al., 2014). Each role must know what the other role needs to ensure that all the other roles that depend on them can perform their tasks efficiently. This will ensure an understanding of "how their role fits within the entire process" (Dhaliwal et al., 2011), improve communication and reduce conflict among the roles (Liu, Chen, Chen & Sheu, 2011). Ultimately, there will be fewer inconsistencies and changes of requirements by the BA and the developers' and the testers' understanding of the requirements is improved (Ghobadi & Mathiassen, 2015). The same misalignment of roles can be addressed in mobile application development teams when focusing on integrating security requirements so that there is a clear understanding within the teams as to who documents security requirements and test cases.

Skills Misalignment

Skills misalignment occurs when the expected competency level of a specific role does not align with their ability to perform the role. Skills misalignment can result in inappropriateness of

responsibilities, idle time and errors. In the current study, one task that was simple resulted in several errors. One developer indicated that as a result of lack of knowledge on configuring the third party security application to work with the mobile application, more time was taken to complete the task than what was initially anticipated. The lack of skills required to implement security requirements is mainly because security education is not usually a part of a software developer curriculum. Most developers learn how to write code. Security skills are an additional proficiency often acquired through experience.

"In university focus was more on performance rather than security algorithms" PARTICIPANT 5, DEVELOPER

The data collected indicated a lack of skills in security implementation as well as in the understanding of security guidelines and standards. The lack of skills can lead to security concerns being overlooked.

"If you are not an expert in this area it will be difficult to pick the best practices- it's a very specialised area and you may end up having a false sense of security" PARTICIPANT 13, DEVELOPER

Team members' competence or the lack of competence in dealing with issues of security is related to an individual's level of security awareness. Poor understanding or awareness of security matters is not an issue which involves end users alone. Siponen (2002) describes various dimensions of security awareness which include organizational, general public, socio-political, computer ethical and institutional education dimensions. The public dimension includes IT professionals and end-users. It is unlikely for one to take into consideration security standards if they are unaware of these standards and guidelines. Data collection and analysis showed a deficiency in skills to document, develop and test security requirements. Mouratidis, Giorgini and Mansona (2005) insist that secure software development is a specialist area. They point out that many developers do not have the right skills to develop secure applications.

In organisations that adopt an agile method, the developers are likely to take on the role of the security specialist. This situation is far from ideal as most developers do not have the correct skill set (Rindell et al., 2015). This is evident in the study as the developers were involved in documenting the security requirements and simulating the test cases. This is evident in the statement below:

"Security on its own is very complex. We needed the help of the developers to set up the testing scenarios" PARTICIPANT 9, TESTER

Role-based training must be offered to all the members of the team as this will ensure that the security requirements are correctly aligned in the software development lifecycle. The product owner or the BA would then know how to include security requirements when documenting the business requirements. The developers and tester would have a good foundation from which to work (Rindell et al., 2015).

Requirements Misalignment

Requirements misalignment occurs when there are conflicting issues between security requirements and the general system requirements. Requirements can either be functional or non-

functional requirements with security requirements categorised as non-functional. Regardless, functional and non-functional requirements are equally important and must be taken into consideration during software development. Fragmentation in requirements classification is important but can result in alienating the different types of requirements, with non-functional requirements having less priority and considered after the design stage (Mouratidis et al., 2005).

Misalignment of security requirements can occur with functional requirements that would have been stated from the beginning of the software development life cycle "since security mechanisms would have to be fitted into a pre-existing design, therefore leading to design challenges that usually translate into software vulnerabilities" (Mouratidis et al., 2005). Security requirements are supposed to mitigate vulnerabilities on functional requirements. However, "security requirements and functional requirements clearly crosscut each other" (Haley, Laney & Nuseibeh, 2004).

4. Conclusions

The study has given insight into the core challenge faced by an agile development team when integrating security requirements into the development of mobile banking applications. Security requirements can be added to the development process by either defining individual security requirements or acquiring a third party security application. Misalignment was identified as the core challenge. The main forms of misalignment identified included external misalignment, role misalignment, skills misalignment and requirements misalignment.

According to Lacey (2011), the field of security in software development is relatively new. Thus, this research adds more theory to the field of security especially the non-technical aspect of security as well as mobile application development. Findings from the study indicated four forms of misalignment that result as challenges in integrating security requirements to mobile applications. Organisations can address the four forms of misalignment to ensure that the process of adding security requirements is less challenging. This research has contributed to practice by pointing out that misalignment issues must be considered before commencing with a software development project, especially one that is considered as a specialist area such as security.

It is important to expand on the current study and focus on additional research to develop descriptive theory and explanatory theory. Further research can enhance the current study by developing propositions that provide a deeper explanation on the relationships between the categories of misalignment. The future studies can follow a similar method as carried out by Volkoff and Strong (2010) in their work on misfits in ERP systems using the critical realism approach, to reveal the deep structures that give rise to misalignment of security requirements.

Misalignment of roles in software engineering has mostly focused on the roles of the developer and tester, citing the interdependence between the two roles as well as the conflict encountered (Zhang et al., 2014). However, the data analysis indicated the importance of the role of the BA in ensuring the understanding of the security requirements in such a way that the developers and testers understand the requirements for their individual roles. Thus, it is important for more researchers to focus on the agile role of the product owner/ BA and how it aligns to roles of the developer and the tester in ensuring a clear outlining of requirements, especially non-functional

requirements such as security requirements. This is supported by Dhaliwal et al., (2011) in pointing out the need for academics to focus on role alignment within an IT unit.

The nature of the mobile application domain suggests that the investigation of this domain may benefit from a Complex Adaptive Systems (CAS) theory view. Few researchers in IT related fields such as project management and software engineering have looked at CAS. Highsmith and Cockburn (2001) looked at CAS and its relationship with the agile methods. However, there is no evidence on the application of CAS in solving misalignment challenges in agile software development teams

References

Abrahamsson, P. (2007) "Agile software development of mobile information systems", *Advanced Information Systems Engineering, 1-4.*

Angelakopoulos, G. and A. Mihiotis (2011) "E-banking: Challenges and opportunities in the Greek banking sector", *Electronic Commerce Research*, (11)3, pp. 297-319.

Bickford, J. et al. (2010) "Rootkits on smartphones: Attacks, implications and opportunities", In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications* (pp. 49-54).

Chan, Y. E. and B. H. Reich (2007) "IT alignment: what have we learned?", *Journal of Information Technology*, (22)4, pp. 297-315.

Daud, M. I. (2010) "Secure software development model: A guide for secure software lifecycle", *In Proceedings of the international MultiConference of Engineers and Computer Scientists* (Vol. 1, pp. 17-19).

De Jonge, M. (2009) "Developing product lines with third-party components", *Electronic Notes in Theoretical Computer Science*, (238)5, pp. 63-80.

Dhaliwal, J. et al. (2011) "Alignment within the software development unit: Assessing structural and relational dimensions between developers and testers", *The Journal of Strategic Information Systems*, (20)4, pp. 323-342.

Ghobadi, S. and L. Mathiassen (2015) 'Perceived barriers to effective knowledge sharing in agile software teams", *Information Systems Journal.*

Glaser, B. G., Strauss, A. L. (1967) *The discovery of grounded theory: Strategies for qualitative research.* Chicago. Aldine Pub. Co.

Haddox, J. M. et al. (2009) "Method for understanding and testing third-party software components".

Haley, C. B., R. C., Laney, and B. Nuseibeh (2004) "Deriving security requirements from crosscutting threat descriptions", *In Proceedings of the 3rd International Conference on Aspect-Oriented Software Development* (pp. 112-121).

Harknett, R. J. and J. A. Stever (2011) "The new policy world of cybersecurity", *Public Administration Review*, (71)3, pp. 455-460.

Highsmith, J. and A. Cockburn (2001) "Agile software development: The business of innovation", *Computer*, (34)9, pp. 120-127.

Lacey, D. (2011) *Managing the Human Factor in Information Security: How to win over staff and influence business managers*, John Wiley & Sons.

Liu, J. Y. et al. (2011) “Relationships among interpersonal conflict, requirements uncertainty, and software project performance”, *International Journal of Project Management*, (29)5, pp. 547-556.
Luftman, J. and R. Kempaiah (2007) “An update on business-IT alignment: “A line” has been drawn”, *MIS Quarterly Executive*, (6)3, pp. 165-177.
Matavire, R. and I. Brown (2013) “Profiling grounded theory approaches in information systems research”, *European Journal of Information Systems*, (22)1, pp. 119-129.
McCallin, A. M. (2003) “Designing a grounded theory study: Some practicalities”, *Nursing in Critical Care*, (8)5, pp. 203-208.
McGraw, G. (2006) *Software security: building security in* (Vol. 1). Addison-Wesley Professional.
Mouratidis, H., Giorgini, P., and G. Manson (2005) “When security meets software engineering: A case of modelling secure information systems”, *Information Systems*, (30)8, pp. 609-629.
Oueslati, H., Rahman, M. M. and B.L. Othmane (2015) “Literature review of the challenges of developing secure software using the agile approach” In Availability, Reliability and Security (ARES), 2015 10th International Conference on (pp. 540-547).
Pickard, A. (2012) *Research methods in information*. Facet publishing.
Pikkarainen, M. et al. (2008) “The impact of agile practices on communication in software development”, *Empirical Software Engineering*, (13)3, pp. 303-337.
Preston, D. and E. Karahanna (2009) “How to develop a shared vision: The key to IS strategic alignment”, *MIS Quarterly Executive*, (8)1, pp. 1-8.
Rindell, K., S. Hyrynsalmi, and V. Leppäne (2015) “A comparison of security assurance support of agile software development methods”, *In Proceedings of the 16th International Conference on Computer Systems and Technologies* (pp. 61-68).
Schadler, T., & McCarthy, J. C. (2012) “Mobile is the new face of engagement”.
Siponen, M. (2001) “Five dimensions of information security awareness”, *Computers and Society*, (31)2, pp. 24-29.
Strong, D. and O. Volkoff (2010) “Understanding organization-enterprise system fit: A path to theorizing the information technology artifact”, *MIS Quarterly*, (34)4, pp. 731-756.
Zhang, X. et al. (2014) “Sources of conflict between developers and testers in software development”, *Information & Management*, (51)1, pp. 13-26.
Zhu, R. (2015) “An initial study of customer Internet banking security awareness and behaviour in China”, *Pacific Asia Conference on Information Systems*, Paper 87, http://aisel.aisnet.org/pacis2015/87/.