

Experiences by Using AFFINE for Building Collaborative Applications for Online Communities

Mohamed Bourimi and Dogan Kesdogan

Information Systems Institute,
University of Siegen, Germany
bourimi@wiwi.uni-siegen.de, kesdogan@uni-siegen.de

Abstract. Continuous problems and deficits in developing complex and ever-changing (software) systems led to agile methods, e.g. Scrum. Nevertheless, the problem of considering a plethora of different functional as well as nonfunctional requirements (N/FRs) remains unsolved and gains in importance when engineering state-of-the-art software. The current tide of approaches aims at handling every single NFR by an individual process integrated into Scrum, yielding a process complexity which can not be handled properly. Scrum-based AFFINE¹ was designed explicitly to provide an alternative solution to over-complex design- and development-processes and still considering all kinds of NFRs early enough in the process. In this paper, we discuss collected findings by using AFFINE in various projects dealing with the development of software for user-centered online communities towards some evidence of its suitability.

Keywords: Agile Software Process, Nonfunctional Requirements Engineering, Security and Usability, User Experience, Scrum, AFFINE.

1 Motivation

Applications, covering many collaboration measures and social aspects for many important areas of our professional and leisure life activities, are increasingly used in our information society. Technical support for this is mainly provided through different kinds of collaborative applications also known as groupware. Software systems and applications supporting collaboration are considered as socio-technical systems in the Human-Computer Interaction (HCI), IT Security, and Computer Supported Collaborative Work (CSCW) research fields [1–3]². Shneiderman et al. state in [1] that most computer-based tasks will become collaborative because most work environments have social aspects. From a general software engineering (SE) point of view, the socio-component is related to human

¹ Agile Framework For Integrating Nonfunctional requirements Engineering.

² Due to the multidisciplinary nature of our contribution and difficulty to consider related work from each research field, we cite in the following one representative work from each research community for argumentation completeness.

factors (developers, end-users etc.) and their influence on Information Systems and Information Technology (IS/IT) projects is significant from various perspectives (i.e. development and management perspectives). The inherent involvement of the (non-deterministic human) socio-component makes the significance and impact of human factors in the development of collaborative applications more crucial than in other IS/IT projects. The ultimate goal of any IS/IT project is to efficiently reach the following aims:

1. reducing costs by optimizing resource allocation,
2. minimizing product delivery failure risk (increasing so opportunity of success), and
3. reaching end-users/customers satisfaction by ensuring good product quality and User eXperience (UX).

Researchers from various fields recognized that solutions of static nature cannot satisfy changing needs, e.g., requirements emerging from the usage of a software system. With respect to collaborative applications, different agile approaches promise better consideration of changing requirements and of human factors. They at least strongly and constantly early involve end-users and better react on uncertainties in the development process (e.g. difficulties related to requirements elicitation, negotiation, etc.). Thus, various user-centered and participatory design methodologies with different degrees of agility are increasingly adopted today when building sophisticated groupware solutions. Furthermore, various human factors related issues arise due to the adopted agility, e.g., between users' and developers' needs. Such needs remain mostly neglected in our opinion and have to be better considered. Agile development is believed to help in reaching these aims, even evidence is still investigated for different project aspects. SE practitioners agree on the need of evidence supporting this believe and state the rarity of studies confirming it. Lack of evidence is the most-cited criticism against agile development methodologies even they are gaining importance.

In this paper, we report on experiences of agile development for building collaboration software with AFFINE [4] by handling nonfunctional requirements (NFRs) at different levels (i.e. management and development level) and avoiding the complexity within the process thereby. The remainder of this paper is structured as follows: Problem analysis is addressed in the following Section. Section 3 presents AFFINE's design while Section 4 discusses first collected experiences by using it for building collaborative applications for online communities. Section 5 concludes our contribution.

2 Problem Analysis and Statement(s)

Recently, agile method(ologie)s such as Extreme Programming (XP) and Scrum are becoming popular in industrial and academic fields. They are used in order to better match changing requirements and human factors³ in the development of

³ Human resources constitute an average of 70% of SE projects costs. "Project management issues (costs, time, schedule) are often considered as non-functional requirements as well", however, at the project organizational/management level [5].

groupware (e.g., [6]). Scrum as an agile framework [7] is experiencing a wide acceptance nowadays [8]. It provides explicit support for addressing human factors related issues in its framework, as shown in recent studies, over the influence of human factors on IS/IT projects [9, 8]. However, practitioners of agile methodologies stress, that adequate support for NFRs is not provided (e.g., it is not easy to consider NFRs in user stories). With respect to Scrum, Ambler states in [10] that *"Scrum's product backlog concept works well for simple functional requirements, but as I described in 'Beyond Functional Requirements on Agile Projects' (www.ddj.com/architect/210601918), it comes up short for nonfunctional requirements and architectural constraints."* [SIC]. In this respect, the identified gap of properly considering NFRs in agile methods is the main reason preventing the adoption of agility in the security (requirements) engineering area for instance. Indeed, a systematic literature review shows explicit reserve [11]. This is originating from various factors i.e.: (i) the nature of security oriented research targets to be formal as possible in order to assess traceability of requirements, their completeness etc., and (ii) security requirements engineering methods were designed and mostly used with classical software life cycle processes (e.g. waterfall or V model cf. Fig. 1) with expected slots for assurance and risk analysis techniques etc., thus being not easily portable or even suitable to agile methods [12] (at least without further research [13]). However, recently one also can notice an emerging need for more investigation with respect to the suitability of agility for security, especially to spare costs while ensuring earlier consideration of security requirements. Figure 2 depicts a suggestion made by some practitioners⁴ from the industrial field on how to extend an agile process (Fig. 2-a) to consider security best practices (Fig. 2-b).

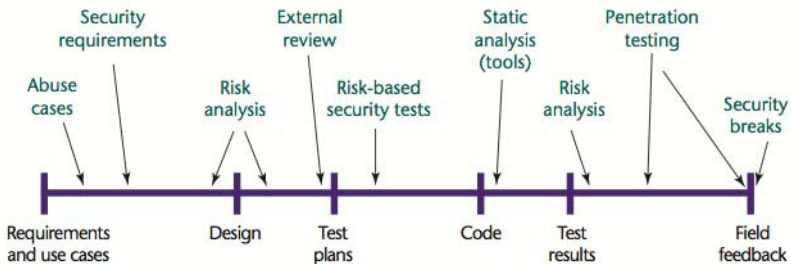


Fig. 1. Software security best practices within a sequential process (from [12])

In contrast to usability, such reserve against agility cannot be noticed in the HCI community. In fact, usability engineers and UX experts tend to ask for agility since it supports earlier involvement and tests while developing the

⁴ Agile and Secure: can we do both?

<http://jazoon.com/portals/0/Content/ArchivWebsite/jazoon.com/jazoon09/download/presentations/7102.pdf> (Last access: March 2013).

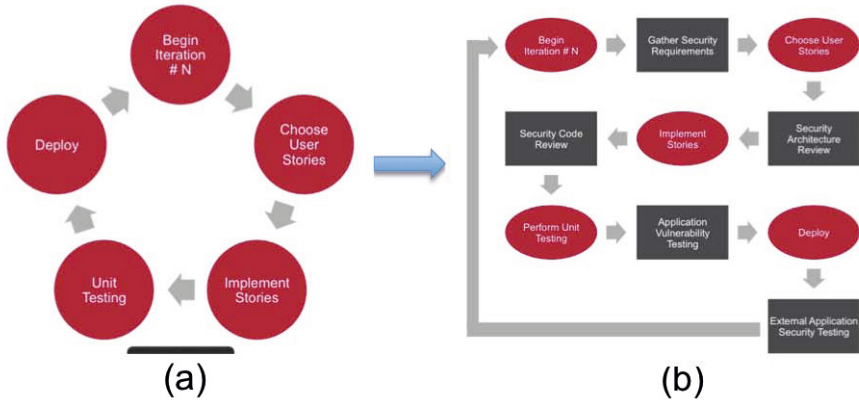


Fig. 2. Suggested extension of an agile process to integrate security practices (by Jason Li and Jerry Hoff)

intended product. However, one can state that most of software and requirements engineering method(ologie)s and development processes followed in each research community are going their own way for dealing with their NFR of importance.

For instance, Lee et al. presents an integrated approach known as eXtreme Scenario-based Design (XSBD) towards agile UX. Figure 3 (a) depicts the "curse of complexity" when trying to extend the basic Scrum scheme (lower part) with

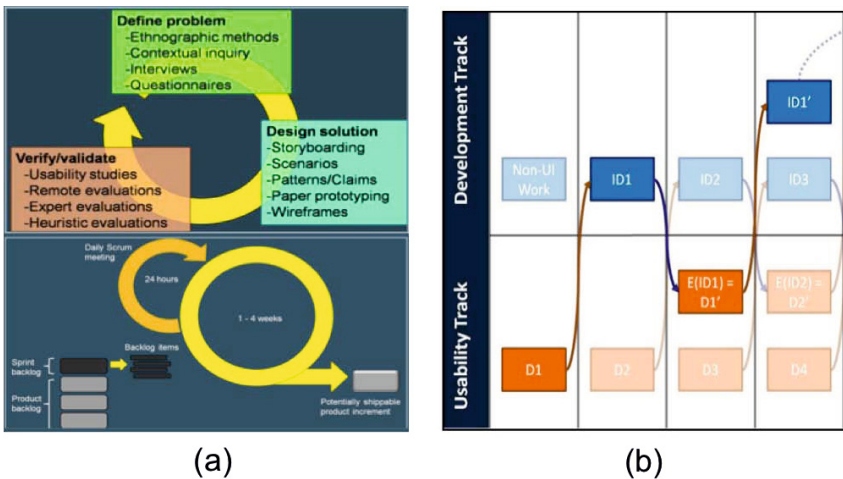


Fig. 3. Usability best practices integrated into agile Scrum (from Lee’s Tutorial Materials at CHI’12 and [14])

different sub-processes for the plethora of different methods addressing NFR (upper part) in analysis, design and implementation phases for complex systems. Based on further experimentation and comparative analysis they also currently try to address XSBDD usage for distributed teams by considering time factors in the process [14] (cf. Fig. 3-b).

We argue that, while the first tide of ignoring NFRs mostly took place at the level of addressing them as *"add-ons, often postponed, not considered as system-wide properties"*⁵ etc., the current tide is doing it at the level of studying their consideration and integration in agile method(ologie)s and processes (and mostly contemplated separately from other (N)FRs as shown above for security and usability people). From both presented extensions of agile processes above (in each example), it is obvious that the consideration of all (N)FRs is a crucial task and one can ask him/herself the following questions:

- How could a consideration/integration of all NFRs look like in agile methods? What will be the resulting scheme of the steps to be followed in the resulting method(ology) or process?
- How much best practices from each *"NFR" Engineering* field should be adopted and explicitly addressed? Who will decide that (by considering that in the same research community different directions exist, mostly not harmonizing with each other)? Which effect has the nature of the product to be developed on such decisions (e.g. critical safety products will surely follow established and well-proven processes)?
- Will the resulting method(ology), process etc. then still be applicable and by whom (in terms of qualifications)? How much will such an adoption cost?

3 AFFINE's Design and Its Suitability for Our Purposes

The result of our research for answering related issues to the previously listed questions is reflected in AFFINE [4]. AFFINE's main targets consist of *simultaneously* addressing previously cited deficits by:

1. conceptually considering NFRs early in the development process,
2. explicitly balancing end-users' with developers' needs, and
3. proposing a reference architecture providing support for NFRs in order to overcome conceptual lack of guidance and support for efficiently fulfilling NFRs in terms of a software architecture in general.

The nature of our (collaborative) scenarios and prototypes/products to be developed, implies earlier consideration of privacy and (multilateral) security along with other competing NFRs, such as usability and social/group awareness throughout the whole software life cycle process. In our case this nature demands an agile way of development. We do this exemplarily in AFFINE by extending Scrum to enforce their earlier consideration at development as well as management level. Choosing Scrum is not just based on mentioned arguments above

⁵ As stated by Santen for Security in [15] and which remains true for all NFRs.

such as selected practitioners' experiences and empirical studies (e.g. [16, 9, 8]), but also on own positive experiences in other projects. Constitutive requirements for AFFINE were gathered based on a detailed analysis of existing work from various research fields (i.e. HCI, CSCW, Security, and SE) as well as based on experiences from various projects on designing and implementing groupware systems or applications needing privacy and (multilateral) security consideration in general.

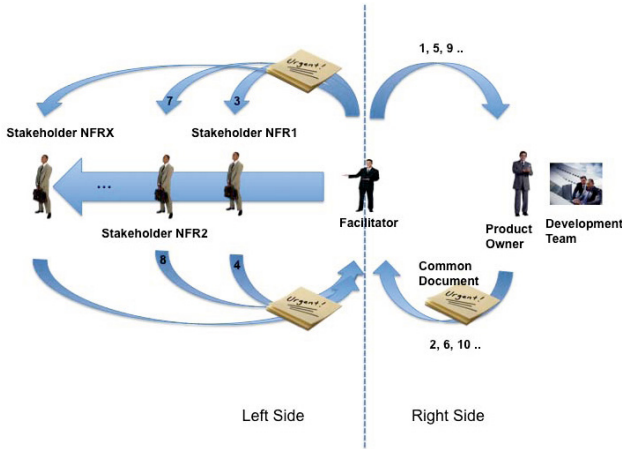


Fig. 4. The big picture of the Scrum-based AFFINE (from [4])

In summary, Scrum was chosen due to its high tailorability, its support for human factors consideration⁶, not only at the level of development but also at the management level, and its increasing adoption in industry as well as positive first-hand management experience with Scrum. Nevertheless, even though Scrum is helping in overcoming many issues in this respect, some of the important phases need to be specified more sharply as this is the case for requirements engineering/gathering in it (cf. [16]). E.g. the usage of a common document⁷ in AFFINE (s. Fig 4) helps in better eliciting requirements and keeping track of changes (e.g. traceability of change requirements, decisions for solving conflicts for a given design among NFRs, etc.). The consideration of NFRs best practices is ensured by explicit involvement of experts in the respective iterations. While in our case, Scrum defines the coarse agile production process, experts still have the chance to integrate their wished practices while developing the product (in the requirements engineering phases, i.e. gathering, elicitation and negotiation).

⁶ In contrast to other agile approaches, Scrum supplies the support for agile project management in general. XP, e.g., primarily focuses on development aspects [7].

⁷ Agile methodologies tend to avoid documentation. However, many extensions in different research fields suggest recently to rethink such practice.

Choosing the necessity as well as degree of accuracy is delegated to them and negotiation as well as conflict solving mechanisms are defined in AFFINE (refer to [4] for how AFFINE is executed, e.g. follow numbering in Fig. 4 for a perfect iteration without breakdowns that could result from conflicts among stakeholders). Furthermore, we suggest a concrete mapping of NFRs by using Service-Oriented Architectures (SOA) and Aspect-Oriented Programming (AOP) techniques (followed since 2006 in our group for this kind of socio-technical software).

4 Experiences by Using AFFINE

We used AFFINE in various projects from 2009 until now to investigate some evidence about its suitability for building collaborative applications. The most important ones are the iAngle⁸ and iFishWatcher⁹ projects which deal with support for online mobile communities. The iAngle project itself emerged as a spin-off from the EU project PICOS¹⁰ that dealt with privacy and identity management in mobile communities. PICOS followed a user centered and scenario based proceeding for eliciting the gathered needs and requirements (such as user stories, interviews and questionnaires) for three different mobile communities (Taxi Drivers Community, Angling Community, and Gamer Community). The Angling Community is built by recreational anglers who explore water bodies and coastal areas, to an extent that is unattainable by scientific projects. They spend enormous time and effort investigating fish communities. The PICOS and iAngle location-based services (LBS) scenarios are of collaborative nature which means, that they presume the interaction of the community members (i.e. entering watercourses and fishing spots, rating those spots, etc.). The prototypes implemented various LBS scenarios such as *"Sharing Fishing Sites"* with different use cases like *"Show Fishing Spots"* and *"Add Fishing Spots"*; as well as *"Localizing Contacts around Me"*. Various functional extensions followed with the time and are reflected in the results listed on the respective websites.

The iAngle project was started at the University of Siegen after PICOS lab and user trials which took place on the 27th/28th November 2009 in Vienna and 12th/13th December 2009 in Kiel. Lab and user trial tests were conducted by members of the Center for Usability Research & Engineering in Vienna and Leibniz Institute of Marine Sciences in Kiel. While the iAngle and iFishWatcher projects strongly followed AFFINE, PICOS played a big role in designing it as well as unintentionally comparing it with classical SE processes. A full description of the results goes beyond the scope of this paper. The interesting point is that stakeholders who observed the development process described it as follows: *"The process methodology followed for the Picos project during development was different within the teams: One team, referred to as the AFFINE-team, followed the AFFINE framework described in this paper. The other three software development teams followed traditional software engineering methods such as the*

⁸ <http://www.uni-siegen.de/fb5/itsec/projekte/iangle/index.html>

⁹ <http://www.ifishwatcher.org/news.php>

¹⁰ <http://www.picos-project.eu>

waterfall model". The AFFINE team was responsible for developing LBS as well as communication functionalities for the Angler Community for the first prototype. This includes also the design of the graphical user interface (UI). A single document containing these scenarios and related use cases was then circulated. The development team involved 3 developers and a product owner as well as a facilitator. This document used UML for further eliciting the requirements and provided first UI prototypes. After this, the document was circulated to the end-users again, then to the privacy and usability stakeholders and so on. A PICOS Platform stakeholder, responsible for integration, was only contacted if missed functionality had to be supported. At a given time, the usability stakeholders introduced a sophisticated click dummy to reach better prototyping. The AFFINE team oriented further UI work to fulfill the click dummy UIs. However, the agile method followed, corresponded not to standard Scrum.

The three projects primarily focused on the evaluation of usability and privacy/security, involved usability experts formulated their observations with respect to AFFINE as follows: *"The expert usability evaluation for the different prototypes was carried out through heuristic evaluations. Usability problems and security aspects were assessed by criteria relevant for usability and guidelines. The reviews from the expert evaluations were communicated via multiple channels such as emails or telephone conferences including small reports. A general observation is that reaction to changes was faster in the AFFINE team than in the other teams in the case of PICOS. Moreover the AFFINE team actively asked for usability feedback during implementation, which made the process even more proactive and faster. Considered from a usability expert perspective in PICOS, the usability feedback for the AFFINE team approach was more focused and delivered in small portions. This approach relieves the usability reviewer from evaluating hundreds of screens and enables them to focus on certain aspects. Besides the expert evaluations, end-users are additionally directly integrated in the project through lab and field tests. The results will be communicated and influence a second phase of development. The review for lab and field tests is not of a quick nature as it is more complex to set up lab tests with real end-users and gain results. Therefore quick usability reviews are more suitable for agile approaches such as AFFINE."*

They also stated that: *"The AFFINE approach allowed the integration of HCI instruments such as usability expert evaluations during the whole development process while considering privacy and security as well. Usability expert evaluations are very suitable to solve the ad-hoc problem solving needs inherent to agile methods. Generally this approach is more successful insofar as, the sooner NFR problems (such as usability and privacy problems) are detected, the less cost sensitive are the changes. This is even more important, as deficits in usability and UX have great impact on privacy issues as well. E.g. it is a well known fact, that good usability and UX are important factors for trust .. the team implementing the AFFINE process achieved better results regarding usability, user experience and privacy on the interface level than teams implementing following a traditional software engineering process. The AFFINE framework requires involving*

NFRs such as usability, user experience or privacy throughout the development process; therefore they have been an integral part from the beginning of the development process. The framework itself incorporates several small cycles that assures all stakeholders to be included. The AFFINE-team reacted instantly to the feedback given by the usability team and used it for further implementation .. In general, design for usability and for privacy have to be an integral part from the beginning of the project, and the agility, flexibility and rapidness of AFFINE meets the aims of reacting to changes very quickly. Especially in projects in the context of NFRs, appropriate agile methods to overcome these challenges are beneficial in direct comparison to traditional approaches.” The iAngle and iFishWatcher projects involved under- and post-graduate students from the University of Siegen and partner institutes (in addition to Germany, from Spain and the Philippines). These students were introduced to AFFINE (and Scrum). Stakeholders in both projects are listed on the websites (i.e. students who participated and performed tasks under supervision of experts for usability, privacy and security as well as SE experts). In the case of iAngle/iFishWatcher, the same PICOS Angling Community was involved. The sprints were very short (5 to 6 days), however, not continuous due to restrictions in academic settings. In general, the evaluation with all experts stated that also this kind of distribution of work was still conform to Scrum and reached at the end very good acceptance.

5 Conclusions and Future Work

The AFFINE framework incorporates several short cycles which assure that all stakeholders are included and that reaction on needed changes can happen in an agile way. To be accurate, the core assumption of this contribution is based on the following: (i) AFFINE is suitable for the development of collaborative applications which could profit from agility due to their complex nature, (ii) AFFINE provides an empirical framework that is powerful enough to handle the problems of early and adequately (according to experts) addressing NFRs without making the process of production complex, and (iii) experts' involvement helps in meeting (i) and (ii) in a multilaterally and qualitatively acceptable manner for all stakeholders within the project. Without (iii) the reader might imagine which process will emerge, if stakeholders are simultaneously extending a process to meet their best practices within their community of interests. Currently we are in the process of analyzing collected data for AFFINE (also in other projects, e.g. the EU funded di.me project) and preparing the results for more accurate scientific dissemination in respective communities (e.g. Empirical SE, Security and Usability Requirements Engineering conferences etc.). Further, we introduced the AFFINE method to practitioners in workshops. A first resonance showed the simplicity of understanding and performing AFFINE-based exercises (e.g. in form of simulation or in different projects works). In summary, first experiences promise great suitability of AFFINE for future work of multidisciplinary nature (HCI and IT Security/Privacy communities in this contribution). Further efforts will focus on the questions listed above and that still need answers and evidence.

Acknowledgments. Thanks are due to Thomas Barth, Eva Ganglbauer, Joerg M. Haake, and Bernd Ueberschaer (in alphabetical order) as well as all people who contributed with design or development activities related to AFFINE.

References

1. Shneiderman, B., Plaisant, C.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 4th edn. Pearson Addison Wesley (2005)
2. Cranor, L., Garfinkel, S.: *Security and Usability*. O'Reilly Media, Inc. (2005)
3. Gross, T., Koch, M.: *Computer-Supported Cooperative Workspace*. Oldenburg (2007)
4. Bourimi, M., Barth, T., Haake, J.M., Ueberschär, B., Kesdogan, D.: AFFINE for enforcing earlier consideration of nFRs and human factors when building socio-technical systems following agile methodologies. In: Bernhaupt, R., Forbrig, P., Gulliksen, J., Lárusdóttir, M. (eds.) *HCSE 2010*. LNCS, vol. 6409, pp. 182–189. Springer, Heidelberg (2010)
5. Cremers, A.B., Alda, S.: *Organizational requirements engineering* (ch. 9) (2010), http://www.iai.uni-bonn.de/III/lehre/vorlesungen/SWT/RE05/slides/09_Nonfunctional%20Requirements.pdf (last accessed, February 2013)
6. Schümmer, T.: *A Pattern Approach for End-User Centered Groupware Development*. Schriften zu Kooperations- und Mediensystemen - Band 3. JOSEF EUL VERLAG GmbH, Lohmar - Köln (August 2005)
7. Schwaber, K., Beedle, M.: *Agile Software Development with Scrum*, 1st edn. Prentice Hall PTR, Upper Saddle River (2001)
8. França, A.C.C., da Silva, F.Q.B., de Sousa Mariz, L.M.R.: An empirical study on the relationship between the use of agile practices and the success of scrum projects. In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2010*, pp. 37:1–37:4. ACM, New York (2010)
9. Li, J., Moe, N.B., Dybå, T.: Transition from a plan-driven process to scrum: a longitudinal case study on software quality. In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2010*, pp. 13:1–13:10. ACM, New York (2010)
10. Ambler, S.: *Complex requirements on an agile project*. Online (2008)
11. Salini, P., Kanmani, S.: Survey and analysis on security requirements engineering. *Comput. Electr. Eng.* 38(6), 1785–1797 (2012)
12. McGraw, G.: *Software security*. *IEEE Security and Privacy* 2, 80–83 (2004)
13. Beznosov, K., Kruchten, P.: Towards agile security assurance. In: *Proceedings of the 2004 Workshop on New Security Paradigms, NSPW 2004*, pp. 47–54. ACM, New York (2004)
14. Lee, J.C., Judge, T.K., McCrickard, D.S.: Evaluating extreme scenario-based design in a distributed agile team. In: *CHI 2011 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2011*, pp. 863–877. ACM (2011)
15. Santen, T.: *Security Engineering: Requirements Analysis, Specification, and Implementation*. Habilitation, Fakultät Elektrotechnik und Informatik, Technische Universität Berlin (2006)
16. Ambler, S.: Beyond functional requirements on agile projects. *Dr. Dobb's Journal* 33(10), 64–66+ (2008)