A Methodology for Experimental Evaluation of a Software Assistant for the Development of Safe and Economically Viable Software

Alina Khayretdinova[1] and Michael Kubach[2]
[1]University of Stuttgart, Institute for Human Factors and Technology Management IAT, Nobelstrasse 12, Stuttgart, Germany
[2]Fraunhofer IAO, Fraunhofer Institute for Industrial Engineering IAO, Nobelstrasse 12, Stuttgart, Germany

Keywords: Assistant Tool, Experimental Evaluation, Viable Security, Privacy by Design, Usable Security, Socio-economic Security, Software Assistant, Software Development.

Abstract: Very often software developers of IT security solutions tend to focus on subjects of privacy and security of the product neglecting other important aspects of the development such as socio-economics and usability that are crucial for the success of the product on the modern market. To address this problem, project CUES developed a software assistant that has an interdisciplinary approach. The assistant guides the developers of IT security solutions through an entire software development process by aiding to identify present problems and suggesting effective solutions from the fields of (a) Usability and User Experience, (b) socio-economics, (c) IT security, and other disciplines. In this paper, we propose a method to evaluate the assistant in the conditions that are closest to reality: the assessment of the software assistant is carried out through two case studies where at each two student teams have a task to develop a security related software that will also be attractive for users and the market. One of the student teams in each case study was supported by the assistant, whereas the second teams were not. The teams supported by the assistant performed better.

1 INTRODUCTION

The software development process of IT security solutions is a complicated multidimensional task that does not rely on a strong basis of security and privacy aspects only but is also dependant on other factors such as e.g. stakeholder requirements, business model, market needs, usability, etc. The consideration of different aspects as security, usability and socioeconomic is crucial for the success of the software product on the market (Koçak, et al., 2015). However, it is a common problem that developers tend to focus on the former aspects while neglecting the latter. According to (Grabowski, 2015) and (Hengsberger, 2018), some of the reasons why innovative products fail on the market and do not deliver any meaningful financial return are the following:

- Poor user experience
- Bad pricing policy
- Lack of market orientation (wrong or small target market)
- No clear understanding of the target audience needs

As can be seen, these are mostly usability and socio-economic factors that actually suffer during the development of products. Moreover, according to (Zibuschka and Roßnagel, 2011a), (Zibuschka and Roßnagel, 2011b), (Greenwald, et al., 2004), software solutions that are successful on the market are usually those that are easy to use and meet the user needs. In case of development of secure solutions for software, the above-mentioned aspects are also applicable and developers tend to overlook or diminish their meaning.

To solve the problems mentioned above, a complex holistic approach should be applied during the development process, in order to produce software solutions that are secure, user-friendly and economically successful. Unfortunately, to knowledge of the authors there is currently no such an extensive approach available that helps to deal with the problem on all the mentioned layers (Hofer and Sellung, 2016). The project CUES addresses this problem by creating a software assistant that is an

Khayretdinova, A. and Kubach, M.
A Methodology for Experimental Evaluation of a Software Assistant for the Development of Safe and Economically Viable Software.
DOI: 10.5220/0008069102340241
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 234-241
ISBN: 978-989-758-386-5

integrated guiding tool for developers of IT solutions (Ruff and Horch, 2018). The overall goal of the assistant is to support developers, who are typically already experts in the IT security field, build on their foundation with assistance to integrate other disciplines (usability and socio-economics) in order to establish a secure but also market friendly software (Hofer and Sellung, 2016). The assistant guides the developers through the whole software development process and on each phase, it presents a specifically defined set of questions to identify the status of the process and possible problems of the current phase. Having defined the problems encountered by the developers, the assistant presents solutions based on the expert knowledge that will help at a particular state of the development process. This knowledge was collected from experts in the fields of IT security, usability and socio-economics through several workshops and is stored in the assistant in the form of questions, problem identifiers, and solutions from three abovementioned fields.

This paper describes the methodology and lessons learned from the experimental evaluation of a software assistant. Two methods were used for the assessment and one of them, an experimental evaluation through a case study is presented in this paper in detail. More information on the previous development stages of the assistant has already been published and is thus out of scope of this paper. It can be found in the following papers: (Hofer and Sellung, 2016) present the selection process of the methods and standards from the fields of usability, socioeconomics, and IT-security used by the assistant; the description and method of construction of the semantic data model that structures the knowledge base of the wizard can be found in (Horch, et al., 2017); and (Ruff and Horch, 2018) provides information on the overall functionality of the software assistant.

The organization of the paper is as follows. Section 2 presents the software assistant and introduces its structure and core model. In Section 3, the evaluation methods of the assistant are described and the conclusion is given in Section 4.

2 THE CUES ASSISTANT

The CUES assistant is a tool that contributes to improving the software development process. It guides the developers of IT security solutions through the whole process following a comprehensive approach that includes such aspects as IT security, usability and socio-economics. By including more disciplines in the development process, the assistant makes the whole development process more comprehensive and inclusive (Hofer and Sellung, 2016). This way it helps to address a technical bias that often leads to drawbacks or blind spots that could have been avoided had the development process included more disciplines.

The assistant comprises a semantic database (Horch, et al., 2017) built on expert knowledge on common problems and challenges that software developers may encounter in the development process as well as adequate and comprehensive solutions to tackle them. Moreover, in order to identify potential problems, the assistant provides specific questions and related information by letting the users of the assistant fill out a questionnaire.

The CUES assistant allows for two cases of use. In the case where the software developers are already aware of the problems they face in the current development process and have identified them, they can directly search for solutions. In cases where the developers do not know whether they might encounter a problem or cannot define the type of issue they encounter, the developers enter the following meta-data for the project:

- Project name
- Short description of the project
- Incorporated phases (e.g. test, development, evaluation, etc.)
- Start date of the project and each phase
- End date of the project and each phase
- Type of software to be developed (e.g. web application, mobile app, etc.)
- Budget of the project
- Number of software developers and security/usability/economics experts.

Further, the assistant asks them a set of specific questions, which will help identify present or possible problems and will then offer the most adequate solutions in the form of methods, best practices or heuristics.

The expert knowledge stored in the assistant includes different types of information such as, for example, current processes for software development and embedded phases, common problems of software development projects, questions that help identify these problems, and relevant solutions for possible problems (Ruff and Horch, 2018). The knowledge base was acquired through numerous workshops on three topics (usability, socioeconomics and IT-security) involving experts from the respective domains – industry practitioners as well as researchers. During these workshops, teams of experts suggested and discussed methods that should be included in the software assistant as well as shared their experience on the common problems during the development process.

Moreover, given the project data mentioned above, the CUES assistant can offer suitable forms to create and build a project.

The assistant offers the following main functions:

- *Browsing function* as an entry point to the assistant that provides an overview of the solutions.
- *Guiding function* as core function of the assistant that guides the users through the development process, helps to identify present and possible future problems, and presents adequate solutions.
- *Editing function* as a tool for experts in order to add, edit and delete problems of software projects, questions to identify the problems as well as solutions (Ruff and Horch, 2018).

All of the functions were assessed during the development process of the CUES assistant to receive the immediate feedback and be able to integrate it before the end of the project. In this paper, we describe only the evaluation of the browsing and guiding functions, paying special attention to the former one. More information on the full functionality of the software assistant can be found in (Ruff and Horch, 2018).

As mentioned above, the *browsing function* serves as an entry point to the assistant where users can get an overview of the available solutions (methods, best practices, etc.) of the assistant. To filter the solutions to problems the developers may be encountering, the following features can be used:

1. The discipline covered by the solution (IT security, socio-economics, usability);
2. The level of knowledge required for its application (e.g. beginner, expert);
3. The effort for its application (e.g. high, low);
4. The phase of the project it may support (e.g. evaluation, implementation);
5. The type of solution (e.g. method, heuristic or design pattern) (Ruff and Horch, 2018).

The overview of a solution includes the following information: name of the solution, project phase for its application, required level of knowledge, application effort, type and discipline (IT security, usability, socio-economics) of the solution, motivation for its application, short description of the solution, further links, references and user rating.

In the next section, the evaluation of the assistant's browsing feature is presented in detail.

3 EVALUATION OF THE CUES ASSISTANT

During its later development stage, the assistant was evaluated with the help of two different methods (see Table 1: Evaluation methods). Through the first method, the *browsing function* of the assistant was tested and evaluated with an experiment involving student teams that had to create and carry out the development of a software product concept. During the experiment, the content (methods, best practices, etc.) of the software was validated. The second method tested the *guiding function* of the software assistant and relied on the opinion of experts (software developers) to whom the assistant was presented at a workshop. There, the experts could experience the full functionality of the assistant by directly using it. During the workshop, the experts gave feedback on the functionality, content and architecture of the software assistant. Moreover, a separate round of interviews took place where the experts were presented with both the CUES assistant and the use cases and later evaluated the software assistant. In this paper, we address the methodology of the experimental evaluation of the assistant's browsing function only, the details as well as the outcomes of which will be described in the next sessions.

Table 1: Evaluation methods.

Method 1	Method 2
• Student projects: *1. Smart Office Device Manager (SODAM)* *2. Identity and access management on the shop floor for industry 4.0* • Two groups per project: *1. Group 1 is supported by the assistant* *2. Group 2 is not supported by the assistant* • Supervision and comparison of the outcome • Feedback of the developers	• Expert feedback: *1. <u>Interviews of experts:</u> the assistant is presented to the experts and explained through different use-cases.* *2. <u>Workshop:</u> Workshop with experts where the assistant is presented to the experts and tested by the experts.*

3.1 Student Projects and Evaluation Matrix

The browsing function of the assistant was evaluated through an experimental case study. (Kitchenham, 1996) gives a very good overview of various evaluation methods of software tools, among which is case studies. She claims that the advantage of this method is the fact that it can be incorporated into the normal software development activities – the characteristic which is crucial for the CUES assistant as it is supposed to be present at different stages of the software development process. Moreover, the product can be considered "scaled-up" to life size, if it is tested on real projects (Kitchenham, 1996).

Two student projects were set up: one of the projects was carried out in the area of "Internet of Things" whereas the second involved the topic of the "industry 4.0" (Internet of Things in manufacturing environments). Each project involved two teams of students, which in their turn consisted of two students each.

To be able to observe the application of the CUES assistant in conditions that are closest to reality and evaluate its effectiveness in comparison, after some time, one student group of each project was granted access to the software assistant as a support during the development of their concept whereas the second group did not. The aim of this method was to find out whether the team supported by the assistant would improve their concept after receiving access to it and whether it would show better overall results compared to the team working without support.

The student teams did not know that they were in fact testing the CUES assistant. They thought they were developing Internet of Things / Industry 4.0 solutions for the team. Throughout the project time, teams had to regularly present the progress regarding the development of their solution and the underlying reasoning why specific decisions were taken to the developers of the assistant in regular status meetings, an interim presentation and a final presentation. Both projects were realized over the timeframe of five months (May to September), the exact plan of the meetings as well as can be seen on Fig. 1 and 4.

After each presentation, the two project reviewers (developers of the assistant) independently evaluated each team on multiple criteria and scored them from 1 as the lowest to 4 as the highest score:

1,0 – 1,4: requirements not met
1,5 – 2,4: requirements met to a minor degree
2,5 – 3,4: requirements met to a satisfactory degree
3,5 – 4,0: requirements met to the highest degree

Scores were combined and in cases of differences in scoring of 2 or more a brief discussion to clarify and agree on a common score followed.

The set of criteria covered the three main focus areas of CUES: IT-security, usability, and socio-economic factors and additionally the approach taken for software development. Thus, the criteria used in the evaluation were the following:

- Approach
  - Structured approach
  - Consideration of existing and related applications
  - Interdisciplinary
- Security aspects
  - Security-orientation
  - Consideration of confidentiality aspects
  - Consideration of integrity aspects
  - Consideration of availability aspects
  - Consideration of accountability aspects
- Socio-economic aspects
  - Product-orientation
  - Consideration of cost-use-aspects
  - Consideration of different criteria for a potential market success
  - Consideration of possibilities to create a product innovation
- Usability aspects
  - User-orientation
  - Consideration of usability standards
  - Consideration of the user experience

The tasks received by the student teams as well as the description of the process and the results of the evaluation are described in the following sections.

3.1.1 Student Projects in the Area "Internet of Things"

As mentioned earlier, the student projects were carried out in the form of an experimental case study. The background for this project was to develop the concept of a Smart Office Device Access Manager (SODAM) that regulates the access rights to the smart-office objects produced by the fictional company SOS AG. The case study presented to the students at the kick-off meeting is the following:

*SOS AG (**S**mart **O**ffice **S**ystems) is planning the production and transfer of a new product line of*

professional smart-office devices to the market: SODAM-Smart. The company has already invested significantly into the development of both hard- and software as they expect a lot from the future market and hope to resist the international competition.
The SODAM-Smart product line covers a variety of smart devices that can be used in a smart office such as projectors, coffee machines, or robot vacuum cleaners. Being connected to the assistant and thus to each other, the smart devices proactively support the everyday office life.

Furthermore, according to the background story, SOS AG hired two competing teams of software developers to design and develop a detailed concept of a digital assistant SODAM. As a result, the teams had to come up with the best solution and sell it to the company. The teams were to work and present their interim and final results separately to the SOS AG managers, who in the framework of the experiment were the reviewers.

The exact project plan can be seen in Fig. 1.

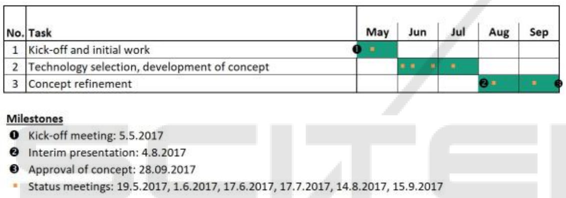

Figure 1: Time schedule of the "Internet of Things" student projects.

The student teams had to update the reviewers at six status meetings and give one interim presentation as well as present their results at the final meeting. Both student teams provided qualitative results at the end of the project. As already mentioned, after each presentation (status meetings, interim and final presentations) two reviewers separately gave scores from 1 to 4 for the work of each team to assess later the impact of the CUES assistant on the quality of the project development process and results.

The results of both student teams (Team 1 and Team 2) can be found on Fig. 2 and 3 respectively. In the middle of the project, Team 2 received access to the assistant as support whereas Team 1 had to work without assistant until the end of the project.

As can be understood from the picture, the team that did not use the assistant (Team 1) had a weaker start than Team 2, especially in the aspect of Usability. Nevertheless, they improved their performance on most of the aspects already on the second meeting. On the third status meeting they showed much higher results on the aspect of usability and kept the level until the end of the project. Another weak point of the team that did not have the support of the assistant was the socio-economic aspect, but they managed to find better solutions to bring it up to the level of other aspects by the end of the project.

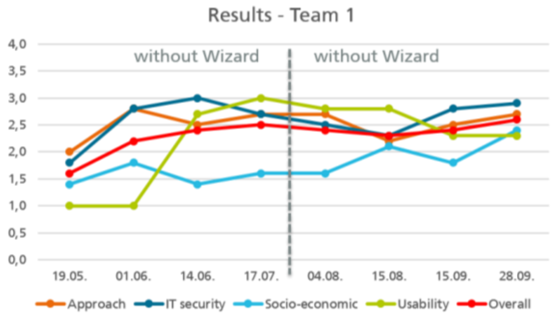

Figure 2: Results of Student Team 1 in the area of "Internet of Things".

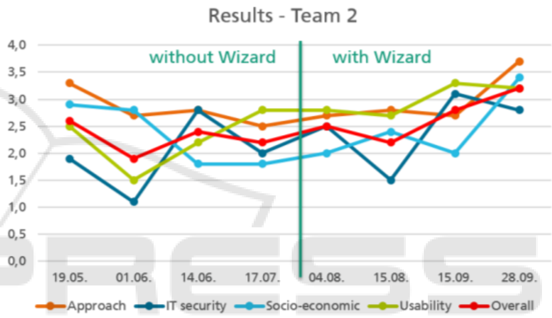

Figure 3: Results of Student Team 2 in the area of "Internet of Things".

On the other hand, the student team that had the assistance of the CUES assistant (Team 2) started with a very good score, with the exception of the IT security aspect, but showed lower results at the second status meeting. From the third presentation, the work of Team 2 had improved and they stayed on this level until the end of the project. Overall, at every meeting, Team 2 showed solid results that stayed approximately at the same level starting from the third meeting. After the assistant was introduced, the performance of the team gradually improved, accelerating at the third meeting after the introduction – probably the team first had to get used to the assistant.

It can be clearly seen that by the final meeting (approval of the concept) both teams showed better performance at almost all of the aspects, especially in their approach and involving the socio-economic features in the development of the concept.

Nevertheless, Team 2 showed better results in the end of the project, scoring between 2,8 and 3,4 for most of the aspects and even 3,7 for the approach they used in building the concept. On the other hand, Team 1 received lower grades from the reviewers – between 2,3 and 2,9 for every aspect. The overall performance of Team 2 is also significantly higher than that of Team 1 – 3,2 against 2,6 scores.

3.1.2 Student Projects in the Area "Industry 4.0"

Additionally, the browsing function of the CUES assistant as well as its impact on the product development process were assessed through experimental student projects in the area of Industry 4.0. The experiment followed the same method by creating a story about a fictional company that hires two teams of software developers to build a concept that fits the company's needs and requirements. As one the previous experiment has already been described in detail, this section covers the second experiment only briefly, paying more attention to the results.

According to the experimental setup, Swiss company "Swiss RMG Electronics" develops and produces electronic components for racing cars that fit individual needs of their clients. They are in the process of digitalizing most of the processes and heading in the direction of Industry 4.0 in their production. As part of this plan, "Swiss RMG Electronics" wants to make the processes inside the company more secure by developing an authentication solution that fits all of their requirements. Therefore, the task for two student teams in this case was to come up with a concept of an authentication for the company that will be secure, easy to use, and will therefore help the company to be competitive on the market by supporting the internal processes and ensuring no mistakes in the work with their clients.

The detailed project plan can be seen below on Fig. 4.

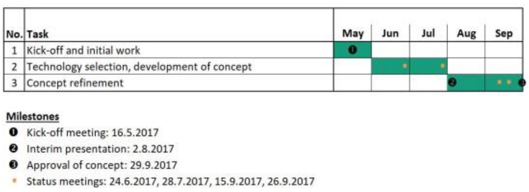

Figure 4: Time schedule of the "Industry 4.0" student projects.

During the project time, the student teams showed their results at four status meetings, an interim meeting and the final meeting to two reviewers (they were not the same reviewers as of the first experiment described in section 3.1.1). At every meeting, the student teams received new impulses from the reviewers to develop their ideas individually.

The performance of the teams was assessed at every meeting according to the quality of the content they provided. The criteria for the evaluation are described in section 3.1 and included four aspects as the first experiment: methodological approach, IT security, socio-economics, and usability. The results of both student teams (Team 1 and Team 2) can be found on Fig. 5 and 6 respectively. Differing from the Internet of Things case, Team 2 was granted access to the assistant already at the second status meeting. Team 1 had to work without support.

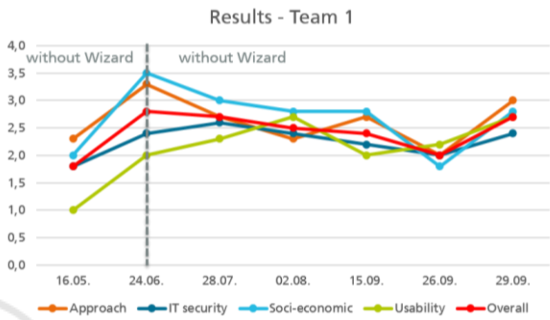

Figure 5: Results of Student Team 1 in the area of "Industry 4.0".

Team 1 started on a good note showing their best results already on the second meeting without using the CUES assistant. Nevertheless, their performance dropped after the second meeting and continued receiving lower scores until the final meeting. For Team 1, the aspect of usability seemed to be the weakest point.

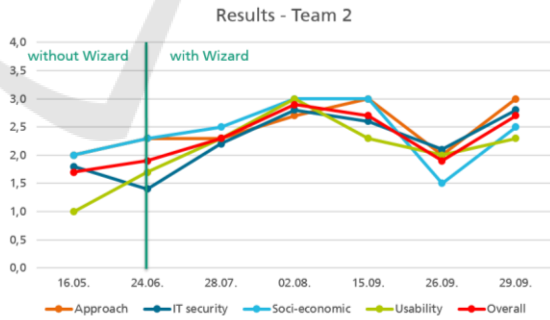

Figure 6: Results of Student Team 2 in the area of "Industry 4.0".

Team 2 on the other hand showed rather average results on the first two meetings but after they received the access to the assistant, their performance started to improve steadily. In comparison with Team 1, Team 2 started showing higher results after the CUES assistant was introduced. Interesting to note, both teams had lower results on the aspect of socio-economics closer to the end of the project. Overall, it can be clearly seen that having the advantage of the software assistant helped Team 2 to show improved results after the second meeting, even if the end results are not very different.

4 CONCLUSION

In this paper, we presented methods used for the evaluation of the CUES assistant, a software assistant that covers disciplines such as IT security, usability and socio-economics and helps its users design and develop software solutions that are strong in all of these aspects. The CUES assistant uses a holistic and easy to use approach supporting the developers in all phases of software development process. We applied different methods for the evaluation of the assistant's functionality, describing the experimental assessment of the browsing function in more detail.

The assistant was evaluated through an experimental case study, which showed that the performance of the software development teams can be improved on different development stages and especially when finalizing the product and delivering it. The results of the assessment suggest that including requirements from other disciplines that are not directly related to its core functionality (IT security) tend to improve the overall quality of the software product making it more secure, usable, and economically successful. Moreover, evaluation through an experimental case study lets us assess the software assistant in conditions that are more or less close to reality and observe the ways in which the assistant was used to gain such first insights before testing it in real production environment. A limitation of our approach is certainly rooted in the number of teams and participants involved which makes it hard to generalize the findings. At the same time, the experimental setup using students as test subjects is less resource-intensive as actual case studies in the real-world environment of a productive company. (Kitchenham, 1996) sees the disadvantage of case studies as an evaluation method for software tools in the fact that there is no guarantee that similar results will be found on other projects, but for this reason we had two more rounds of evaluation using different methods, which gives us more confidence in the results of the conducted case study. After integrating the learnings from the experimental evaluation into the assistant, we therefore carried out the second stage of evaluation (see Table 1: Evaluation methods) with professionals as expert reviewers that participated in the final workshop which took place during the SICHERHEIT 2018 conference[1]. All three functions of the CUES assistant (browsing, guiding and editing) were evaluated at the workshop and the wizard received good acceptance. Apart from that, more usability, IT-security and socio-economics experts were interviewed in order to gather deeper analysis of the wizard. The results of these activities were included into the assistant as well. As this is a short paper that focuses due to the limited space available on the evaluation of the assistant's browsing function only, the details of the next evaluation stages are not presented in detail here.

5 LIMITATIONS

Unfortunately, it was not possible to test and evaluate the assistant in a real working environment during the project time, as it would require more time and effort than defined in the framework of the project. Nevertheless, the browsing function of the CUES assistant is actively used within the large-scale project NGI_Trust[2] funded by the European Union's Horizon 2020 research and innovation programme. As part of its open call for projects[3], successful third party applicants will be using the assistant during the development of their novel trust enhancing solutions for the Next Generation Internet. Specifically for this reason, the content of the wizard was translated into English.

Moreover, it is important to note that the CUES assistant was initially developed to be used by SMEs. It is a lightweight tool which supports smaller projects that have limited resources for the product development and if possible, it should be supported by experts from involved areas. Nonetheless, as a result of our various expert workshops with practitioners, larger German companies were interested in using the assistant for internal educational and development purposes as well.

ACKNOWLEDGEMENTS

We thank and acknowledge the Baden-Württemberg Stiftung for financing the CUES project. For more information, please visit: https://www.bwstiftung.de/.

[1] https://sicherheit2018.in.htwg-konstanz.de/programm/

[2] https://www.ngi.eu/

[3] https://www.ngi.eu/opencalls/ngi_trust-open-call/

REFERENCES

Grabowski, P., 2015. *7 Reasons New Products Fail.* [Online] Available at: https://community.uservoice.com/blog/why-products-fail/ [Accessed 5 October 2018].

Greenwald, S. J., Olthoff, K. G., Raskin, V. and Ruch, W., 2004. The user non-acceptance paradigm: INFOSEC's dirty little secret. *Proceedings of the 2004 workshop on New secuirty paradigms*, pp. 35-42.

Hengsberger, A., 2018. *4 reasons why innovations fail.* [Online] Available at: https://www.lead-innovation.com/english-blog/why-innovations-fail [Accessed 5 Ocober 2018].

Hofer, J. and Sellung, R., 2016. *An interdisciplinary approach to develop secure, usable and economically successful software.* s.l., s.n., pp. pp. 153-158.

Horch, A., Laufs, U. and Sellung, R., 2017. A Semantic Data Model for the Development of Secure and Valuable Software. *Open Identity Summit 2017*.

Kitchenham, B. A., 1996. Evaluating software engineering methods and tool part 1: The evaluation context and evaluation methods.. *ACM SIGSOFT Software Engineering Notes 21.1*, pp. 11-14.

Koçak, S. A., Alptekin, G. I. and Bener, A. B., 2015. Integrating Environmental Sustainability in Software Product Quality. *RE4SuSy@ RE*, pp. 17-24.

Ruff, C. and Horch, A., 2018. *A software assistant for the development of secure, usable and economically meaningful software..* s.l., The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp. 136-142.

Zibuschka, J. and Roßnagel, H., 2011a. A framework for Designing Viable Security Solutions. *Proceedings of the 2011 Workshop on Information Security and Privacy (WISP 2011)*.

Zibuschka, J. and Roßnagel, H., 2011b. A structured approach to the design of viable security systems. *Proceedings of the Information Security Solutions Europe Conference (ISSE)*.