Summer 6-15-2016

THE IMPACT OF SECURITY BY DESIGN ON THE SUCCESS OF OPEN SOURCE SOFTWARE

Golriz Chehrazi
Fraunhofer Institute for Secure Information Technology, chehrazi@sit.fraunhofer.de

Irina Heimbach
Technische Universität Darmstadt, heimbach@emarkets.tu-darmstadt.de

Oliver Hinz
Technische Universität Darmstadt, hinz@emarkets.tu-darmstadt.de

Follow this and additional works at: http://aisel.aisnet.org/ecis2016_rp

Recommended Citation
Chehrazi, Golriz; Heimbach, Irina; and Hinz, Oliver, "THE IMPACT OF SECURITY BY DESIGN ON THE SUCCESS OF OPEN SOURCE SOFTWARE" (2016). Research Papers. 179.
http://aisel.aisnet.org/ecis2016_rp/179

This material is brought to you by the ECIS 2016 Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in Research Papers by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
THE IMPACT OF SECURITY BY DESIGN ON THE SUCCESS OF OPEN SOURCE SOFTWARE

Research

Chehrazi, Golriz, Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany, chehrazi@sit.fraunhofer.de

Heimbach, Irina, Technische Universität Darmstadt, Darmstadt, Germany, heimbach@emarkets.tu-darmstadt.de

Hinz, Oliver, Technische Universität Darmstadt, Darmstadt, Germany, hinz@emarkets.tu-darmstadt.de

Abstract

The Security by Design paradigm – a systematic awareness for and integration of security during the whole lifecycle of a software product – is claimed to be beneficial. Studies emphasize the economic and social benefit of early security consideration during software development. Unfortunately, this claim is little supported by empirical studies. The objective of this study is to examine the relations between perceived effort invested in IT security in distinct phases of software development and deployment and their impact on project success. We address the following question: Does early effort invested in security contribute to the success of software? We conduct an online survey among project leaders of Open Source Software projects and combine these data with objective, secondary data. As dependent variable we examine the perceived success of the software projects as well as the number of downloads as an objective measure for software success at three different points in time. We find that considering IT security in the early stages of development is positively related to the project’s success whereas late consideration is negatively related.

Keywords: Secure Software Development, Survey, Open Source Software.

1 Introduction

Because of the steady growth of mobile and cloud computing, the widespread use of Internet-enabled applications and systems interacting over the Internet, secure IT systems and applications become increasingly important for private users, companies, and the society (Brown et al. 2009; Evers 2006; Waidner et al. 2014). The consciousness for secure IT systems and software (SW) has increased by several incidents of security breaches in recent years e.g., through cyber-attacks at Sony (Bilton and Stelter 2011), Twitter (Jones 2013), and LinkedIn (Silveira 2012). Such security incidents entail increased costs for vulnerability fixing and distribution, data security improvements, and incident investigations and can ultimately lead to users’ and customers’ churn (Nofer et al. 2014; Suby 2013).

The Security by Design paradigm reflects a systematic awareness for and integration of security issues as an important SW quality criteria, during the whole lifecycle of a SW and implies the design of SW’s security right from the start of the SW development (Aberdeen-Group 2010; Forrester 2011; Viega and McGraw 2011). Advocates of the Security by Design paradigm and related studies emphasize long-term economic and social benefits of early investments into security such as cost reduction in development and maintenance phases by avoiding considerably increased late fixing effort, by the reduced likelihood of future security incident costs, by creation of competitive advantages, next to increased customer trust.

Unfortunately, the economic value of Security by Design effort has not been investigated in comprehensive empirical studies yet. Available studies present rather anecdotal evidences or are conceptual papers published by security-related or large SW companies, consulting firms or governmental institutions (Aberdeen-Group 2010; Cavoukian and Chanliau 2013). These studies often lack scientific rigor and a clear description of research processes (e.g., measurement of dependent and independent variables), which impedes the replication and validation of the results (Verendel 2009). The studies addressing the measurement of IT Security effort, relations and dependencies, and its impact on benefits are rarely comparable or generalizable (Anderson et al. 2008). As suggested by Schryen and Rich (2010) and Brecht and Nowey (2013), more empirical work is necessary to analyze benefits promised by Security by Design paradigm.

Our objective is to address this gap and to shed light on whether and how early investments into IT security impact the SW success. The underlying assumptions here are that an increasing awareness of users and developers for security contributes to the perceived and observed security level of a SW to be one selection criteria in favor of their SW’s choice and that the scope of effort taken is reflected in the SW products resulting security level. In this study we examine how effort invested into IT security in distinct phases of software development and deployment is related to project success. For this purpose we conduct an online survey among project leaders on SourceForge.net, one popular Open Source Software (OSS) collaborative platform. As dependent variable we examine the perceived success of the software projects as well as the number of downloads as an objective success measure for software at three different points in time.

We focus in our research on OSS because of 1) the general increase of the popularity of OSS and 2) its embedding in commercially developed SW products (Arora et al. 2004; Forrester 2011; Fortify’s Security Research Group 2008). Thus, OSS’s security level influences many commercial products’ security (Baize 2012). For example, a security flaw in the widely used cryptographic software library OpenSSL disclosed in 2014 forced many companies to invest large amounts of resources to reduce the impact of this vulnerability on their services (Grimes 2014). However, the procedure presented in this paper can also be applied to proprietary SW development projects.

Our work is of high practical importance. As the benefits have not been shown empirically, it is hard to motivate and convince companies and developers to invest time, money, and manpower into security from the very beginning of the SW development. The lack of empirical evidence might then hinder the adoption of Security by Design practices, although it might be worthwhile in the long run. Further, our empirical investigation contributes to the body of knowledge by reflecting the current practice of efforts in security in comparison with general development efforts over time of Open Source projects.

The remainder of this paper is structured as follows. The following section presents background and related work. In the conceptual framework section we develop our hypotheses. Further, we describe our data collection process, provide summary statistics of the sample characteristics and present the estimation results. The paper concludes with a summary of the main findings, followed by a discussion of its limitations and directions for future research.

2 Related Work

Although advocates of Security by Design promote long-term economic benefits, there are only few empirical studies that address this issue. Most of the studies are white papers and technical reports e.g., (Aberdeen-Group 2010; Cigital 2003; IBM 2013), or single case studies that investigate benefits of the implementation of particular IT security mechanisms e.g., (Baca et al. 2008, 2013).

Based on expert interviews of SW developers, Tassey (2002) develops general cost factors to reflect the effort needed to fix bugs for the six distinct SW lifecycle phases: “Requirements Engineering and Design”, “Coding”, “Integration and component Testing”, “Beta tests”, and “Post-Release”. A study by...
Cigital (2003) based on a single case study shows average cost savings by early vulnerability fixing of over £1.5 million. Unfortunately, this study does not describe how the data were retrieved and how the figure was calculated. Stecklein et al. (2004) provide an overview of studies ascertaining fixing cost factors and conduct one case study to determine bug fixing costs for the four initial development phases: Requirements Engineering, Design, Coding and Testing, and Deployment. These results highlight cost factor increases of 29 up to 1500 units to fix requirement errors during deployment. Studies by Neubauer et al. (2005) and Neubauer and Hartl (Neubauer and Hartl 2009) focus on calculation of security return on investments and decision support for IT security investment strategies. The study by Baca et al. (2008) examines a company’s return on investments by using a static code analysis tool based on customer reports. With regard to data collection, most studies apply surveys (Cigital 2003, 2011; CIO 2007; IBM 2013), expert interviews (Baca et al. 2008; Tassey 2002), observational data (Reinke and Saiedian 2003; Schryen and Rich 2010), and a combination of different methods (Baca et al. 2013; Schryen 2011). Empirical work on the measurement of IT security effort on the SW project success is challenging due to the lack of consistent and validated metrics (Fenz 2010; Verendel 2009). Previous research uses two different types of variables to quantify IT security 1) vulnerability lifecycle-based metrics: e.g., vulnerability density and discovery rate (Alhazmi et al. 2007), number of, mean time between and time to provide patches to published vulnerabilities (Schryen 2011), number of security bugs found during project’s lifecycle using issue tracker data (Sartori et al. 2011), and severity of published vulnerabilities (Reinke and Saiedian 2003) and 2) market-based metrics: e.g., the financial impact of information security using breach-disclosure information and changes in share value (Ko and Dorantes 2006; Telang and Wattal 2007; Zafar et al. 2011).

The success of a SW project is attributed not only to quality related factors such as security (Crowston et al. 2006; Lee et al. 2009). With regard to Open Source software, the number of users, subscribers and developers positively impact OSS success (Sen et al. 2012). Schweik et al. (2009; 2012) find in a longitudinal large scale study combining survey and observational data that the high number of active developers and users positively impact ongoing collaboration in OSS projects. The results regarding the impact of complexity on project success are diverse (Schweik and English 2012).

To check whether a SW’s perceived security level is an impacting significant influence factor on project success, developers’ security considerations need to be observable by users. Indeed, several ways exist for users to investigate the extent to which security issues have been considered in a SW development project. They can investigate vulnerability announcements at the projects website, search for security issues in SW related forums or in issue trackers, look for security relevant update information in releases, and for information provided in documentations (Groven et al. 2010). When security is perceived as an important quality criteria, users’ decision in favor of a SW may be reflected by its download. In fact, the cumulative number of downloads is, next to others, one metric that is widely used to measure OSS project’s popularity as a measure of market success and the level of consumers interest in the SW, and its growth (Crowston et al. 2006; Midha and Palvia 2012; Schweik et al. 2009; Sen et al. 2012; Subramaniam et al. 2009; Emanuel et al. 2010; Robinson and Vlas 2015).

Despite the ongoing debate whether OSS is more secure than proprietary SW (Brown 2002; Hoepman and Jacobs 2007; Raymond 1999), observational, objective data based empirical studies show no significant difference between OS and proprietary SW’s security (Anderson 2002; Schryen 2011). Besides, there is no different treatment of secure design for OSS projects observable. Guiding OSS literature includes basic principles of computer security, such as the treatment of security vulnerabilities, the importance of prioritizing their fixing and how to maintain and properly publish security releases (Fogel 2005). Efforts in and treatment of security depend on the policies of the particular development communities (Schryen and Rich 2010), competences, priorities and knowledge of the project leader combined with conducted project management practices and treatment of quality control (Crowston et al. 2012).
3 Conceptual Framework

As we are interested in the effect of IT security efforts on the early stages of a SW lifecycle, we distinguish between the initial development phases (Dvlpmt) and the deployment phases (Dplymt). The initial development includes the four following phases: Planning and Requirements Engineering (Re), Design (De), Coding (Cod), and Testing (Test) and associated activities as depicted in Figure 1 and Table 1. According to the Security by Design paradigm, some important security influencing factors such as system requirements, design concepts, basic SW architecture and technology are determined during initial development phase (McGraw 2006). Therefore, security should already be considered in the initial development phases building a system which is secure by default. Thus, we hypothesize:

H1: The higher the relative effort into security and related activities in the initial development phase, the higher the Open Source Software’s project success

Often security effort increase during deployment phases when the software is widely used. Start of deployment phase is defined as the point in time when the first SW version is released that is assumed to be stable and well tested. It is comparable with the point in time of a SW roll-out that the respondent – the project leader - would assess to be well usable or production ready. According to the advocates of Security by Design such approaches might negatively influence the project success. The late consideration of security might lead to higher efforts to enable security features or to fix vulnerabilities. Security shortcomings might not only be seen as a threat but also as a signal of low quality and consequently lead to fewer users. The second hypothesis is thus:

H2: The higher the relative effort into security in the deployment phase, the lower the Open Source Software’s project success.

Additionally, we include controls for SW complexity, the number of testers and developers, project lead development skills as well as whether the OSS project receives support from businesses. Following Midha and Palvia (2012), Schweik and English (2012) and Sen et al. (2012), we expect that a large developer and user base are positively related to project success. These factors are measured by the average number of developers and testers in our study. The self-reported programming skill rating contributes to project success in the initiation stage of OSS projects (Schweik and English 2012). Thus, we hypothesize that the skill level of the project leader has a positive impact on project success, because it reflects the project leads’ competences. Prior work suggests a positive relation of business involvement on OSS project success (Steward et al. 2006), the impact of the software complexity is unclear (Schweik and English 2012). Hence, we also include these self-reported factors in our model.

Figure 2 depicts the conceptual model. First, we use the perceived, self-reported project success assessment of the project leader to analyze if any relation between this variable and efforts in security can be observed. Our long-term attempt is to objectivize indicators to enhance comparability, enable automatic collection of more datasets and thus, achieve validity improvements. Therefore, the number of downloads is additionally used as operationalization for the project success. Users selection decision in favor of a SW can be expressed by a higher usage rate and thus by the cumulative number of downloads. The
control variables are displayed in dashed rectangles. Project success is used as dependent variable. The relationships between the variables are displayed by the signs on the arrows. A positive relationship is depicted by the sign “+”, a negative relationship by the sign “−” and an unclear relationship by “?”.

Figure 2. Research model

4 Research Methodology

4.1 Survey Design and Sample Selection

To investigate the effect of Security by Design on the project success we administered an online survey. We ask project leaders to rate on the 7-point Likert scale (Likert 1932) the general effort, effort invested into IT security during the different development phases and corresponding activities, the success of the OSS project being characterized by qualitatively or functionally good SW, a big number of users and interested organizations, and several other questions. To avoid misunderstandings, we provided definitions and descriptions for all terms related to SW development and IT security (see Table 1) and the general SW lifecycle (Figure 1) in the questionnaire ensuring a similar understanding of terms among the participants. To enable comparability, we distinguish the development process in different phases.

Until start of deployment, the process is separated into idealized phases that are associated with activities. In the questionnaire, we acknowledge that particularly in OSS development a sharp distinction between initial development and deployment is difficult to make, due to flexibility, fluctuating number of persons involved and the "release early and often"- paradigm, but point out the importance of the distinction made due to our assumption that important security influencing factors are determined during initial development phase. We ask to estimate the average general effort of the main project team invested for typical activities assigned to these phases until start of deployment, irrespective of the development process conducted. Exemplary activities are provided to enhance common understanding (see Table 1). Table 2 shows exemplarily survey questions with focus on those related to general and IT security effort. Further, we asked the participants for information related to their OSS project (for example, name, starting date, deployment start, end of project if existent, SW category, business support, average number of developers and testers involved in the project and the source code’s complexity) and for information related to the project leader (for example, education level and development skills).

The questionnaire was tested in a pilot study with three OSS project leaders that answered to our invitation request to test the survey (sent to randomly selected participants) and three researchers from distinct fields. Based on the pilot test participants’ input we implemented minor design changes and provided additional definitions. Data were collected by sending a link to our online survey to the project leaders of OSS projects on SourceForge.net. We randomly selected 200 projects from each of the ten software categories provided at SourceForge, e.g., Audio & Video, Business & Enterprise, Communications, Development to cover a broad range of distinct applications. After removing double entries caused by multiple categories assignment, we invited 2,858 project leaders to participate in our survey. We did not inform the participants about the exact research question and only explained that the study

---

1 SourceForge provides access to a large repository of OSS projects including project’s metadata and issue data for researchers to analyse OSS phenomena. We chose SF to be able to extend our analyses with further objective project data.
investigated the impact of IT security issues on OSS projects. The survey was open for 5 weeks. We sent one reminder after 2 weeks. 158 participants finished the survey (response rate 5.53%). As the security effort values for security SW substantially differ from the other SW types, we excluded all security related SW projects (N = 18) from our further analyses. After consistency checks and discarding of incomplete responses the final data set contains N=114 observations.

Table 1. Definition of terms related to the present research

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT security</td>
<td>IT security includes all aspects of SW and network security used to ensure availability, integrity and confidentiality of systems and applications e.g., rights management, elimination of security vulnerabilities (buffer overflows, etc.), authentication, and encryption mechanisms, testing for security vulnerabilities, privacy issues.</td>
</tr>
<tr>
<td>Start of deployment</td>
<td>Start of deployment is defined as the point in time when the first SW version is released that is assumed to be stable and well tested. It is comparable with the point in time of a SW roll-out that you would assess to be well usable or production-ready.</td>
</tr>
<tr>
<td>Exemplary activities in Planning and Requirement Analysis – (RE)</td>
<td>Project planning activities and eliciting, analyzing, documenting, validating and managing SW or system requirements, …</td>
</tr>
<tr>
<td>Exemplary activities in Software and System Design &amp; Specification – (De)</td>
<td>Component and algorithm design and architecture design based on requirements. Description of desired features and operations in detail including screen layouts, process diagrams, etc.</td>
</tr>
<tr>
<td>Exemplary activities in Coding/ Implementation and Module Testing – (Coding)</td>
<td>Acquiring and installing systems environment, creating and testing databases, preparing test case procedures, test files coding, compiling, refining programs, etc.</td>
</tr>
<tr>
<td>Exemplary activities in Integration and System Testing – (Test)</td>
<td>Integration of modules and checks for system errors, bugs and interoperability, etc.</td>
</tr>
<tr>
<td>Activities in Operations, Deployment &amp; Maintenance – (Dplynt)</td>
<td>After start of deployment phase: software changes, corrections and additions of functionalities; maintenance activities.</td>
</tr>
</tbody>
</table>

4.2 Descriptive Analysis

Table 3 provides the sample characteristics of our study. Our sample covers distinct SW categories spanning from desktop applications to mobile applications and libraries (multiple selections possible). The numbers of developers and testers are right-skewed with means of 4.08 and 26.82 (median=1.1 respectively 1.05), respectively. Software’s complexity (mean=4.86) and perceived project success (mean=4.77) are almost normally distributed. The project leaders are on average well skilled in SW development (mean=4.09). The project’s lifecycle measure the number of months from project start at SourceForge until questionnaire date. It spans from one month to 162 months (mean=75).
Table 2. Selected survey questions

<table>
<thead>
<tr>
<th>Factor</th>
<th>Question</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity of source code</td>
<td>How do you approximately rate the complexity of the project’s source code?</td>
<td>7-point Likert scale (1 = very low; 7 = very high)</td>
</tr>
<tr>
<td>Business Support</td>
<td>Has the project been supported by a business?</td>
<td>Binary (Yes = 1; No = 0)</td>
</tr>
<tr>
<td></td>
<td>(Def.: Business support or sponsorships refers to active interests of business organizations in the project. Indicators might be companies who pay their employees to participate and/or donate directly.)</td>
<td></td>
</tr>
<tr>
<td>General effort distribution</td>
<td>How big was the general development effort invested into the project in the following SW lifecycle phases (Re, De, Coding, Test, first third, middle and last third of Deployment)?</td>
<td>7-point Likert scale (1 = no effort; 7 = very high effort)</td>
</tr>
<tr>
<td>IT security effort distribution</td>
<td>How big was the effort invested into IT security implementation and integration in the following SW lifecycle phases (Re, De, Coding, Test, first third, middle and last third of Deployment)?</td>
<td>7-point Likert scale (1 = no effort; 7 = very high effort)</td>
</tr>
<tr>
<td>Project success</td>
<td>How do you rate the success of your project?</td>
<td>7-point Likert scale (1 = very small; 7 = very high)</td>
</tr>
</tbody>
</table>

Figure 3 shows the distribution of the general development effort in the SW lifecycle phases in absolute terms. The percentage shares of the effort for each phase are displayed in the lower part. The diagram highlights that RE and Design phases are neglected by 7% (N=8 projects) and 5.3% (N=6) of OSS projects, respectively. Most effort was made during the coding phase followed by testing effort. Less than 1% (N=1) did not test its project code, more than 34% only invested low to very low effort (N=39), 21.1% moderate effort (N=24) and almost 44% invested quite high to very high (N=50) amounts into testing. There is an increase in projects with no, low and very low effort during the deployment phases possibly reflecting abandoned projects that were not successful or projects that stopped development and only maintained the current project status. Beginning of deployment is on average marked by very high efforts. A continuous decrease of general development effort can be observed in the course of deployment. This movement seems reasonable, because the general assumption is that, after a well-tested SW product with originally planned functionalities is released, major changes, corrections and new feature implementations will take place through user feedback in the beginning phases of deployment. The more time elapses the more the effort will shift to maintenance activities and less corrections and additions will take place on average.

Figure 4 shows effort in IT security in the OSS projects lifecycle. Security has not been considered in requirements engineering in almost 45% (N=51) of the projects, more than 37% (N=43) have not put any effort in security during the design phase. Astonishingly, more than 25% (N=29) of the projects have not taken security into account during the coding phase. Moreover, around 33% (N=38) neglected security also during the testing phase. Overall, most of the effort in security in initial development was done during coding (39.5% moderate to very high effort, N=45), followed by moderate to very high efforts during testing (29.8%, N=34) and design (28.1%, N=32). Almost one third of the projects (30.7%, N=34) did not invest in IT security at the beginning phase of deployment, around 40% low to moderate effort. Still, 9.6% projects invested quite high to very high effort in IT Security (N=11) steadily increasing to 14.9% (N=17) during the end phase.
### Table 3. Sample characteristics

<table>
<thead>
<tr>
<th>Variables</th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
<th>Median</th>
<th>Percent of sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop application /Stand alone</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>46.49%</td>
</tr>
<tr>
<td>Libraries</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>28.07%</td>
</tr>
<tr>
<td>Server centric applications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>23.68%</td>
</tr>
<tr>
<td>Research &amp; scientific applications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>21.05%</td>
</tr>
<tr>
<td>Desktop client</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>14.04%</td>
</tr>
<tr>
<td>Web applications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>09.65%</td>
</tr>
<tr>
<td>System development</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>08.77%</td>
</tr>
<tr>
<td>Mobile applications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>05.26%</td>
</tr>
<tr>
<td>Embedded software</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>05.26%</td>
</tr>
<tr>
<td>Network applications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>01.80%</td>
</tr>
<tr>
<td># Developers (average)</td>
<td>4.08</td>
<td>12.37</td>
<td>0.1</td>
<td>125</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td># Testers (average)</td>
<td>26.82</td>
<td>188.04</td>
<td>0</td>
<td>2000</td>
<td>1.05</td>
<td></td>
</tr>
<tr>
<td>Complexity of SW</td>
<td>4.86</td>
<td>1.30</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Business support</td>
<td>.21</td>
<td></td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development skills of project lead</td>
<td>4.09</td>
<td>.9</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Project success</td>
<td>4.77</td>
<td>1.53</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Project’s lifecycle (months)</td>
<td>75.03</td>
<td>49.32</td>
<td>1</td>
<td>162</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td># Downloads at questionnaire date</td>
<td>604,384.3</td>
<td>1,713.25</td>
<td>32</td>
<td>13,300,000</td>
<td>22,625.5</td>
<td></td>
</tr>
<tr>
<td># Downloads after 6 months</td>
<td>666,082.5</td>
<td>1,865.31</td>
<td>89</td>
<td>14,603,500</td>
<td>31,228.5</td>
<td></td>
</tr>
<tr>
<td># Downloads after 12 months</td>
<td>713,011.1</td>
<td>1,975.83</td>
<td>101</td>
<td>15,500,000</td>
<td>31,881.5</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 3. General development effort in distinct phases of the software lifecycle
To sum up, we observe an opposite movement of security effort compared to the general development during deployment. Security efforts during deployment slightly fluctuate but remain constant in average. A slight decrease can be observed towards the end deployment phase. For almost 15% of the projects security effort steadily increase from deployment start to the time of the survey. This might reflect successful projects with increasing users and developers that demand for security.

4.3 Modelling and Estimation Approach

To test our hypotheses and to determine whether the Security by Design paradigm is positively related to project success, we calculate the variables which measure relative IT security effort as follows. First, we divide the SW lifecycle into two phases: initial development consisting of RE, De, Coding and Test, and deployment consisting of the three deployment phases (see Table 1 and Table 2). Then, we calculate the average general effort and the average effort invested into IT security for each phase and put both values into relation to account for the relative security effort taken during initial development (SecEffortDvlpmt) and during deployment (SecEffortDplymt). For example, a projects average security effort in development is 2.75 if invested very low security effort in RE and Design, moderate effort in Coding and low effort in Testing. The average general implementation effort is 4 if low effort is invested in RE, quite high effort in Design and Coding and low effort in Testing. The resulting relative security effort during development is consequently .69. For another project that has only invested low effort in security during Coding (average = 1.5) compared to very low development effort during Design, high Coding and low Testing effort (average = 3), the relative security effort in development is lower, 0.5. Further, based on findings of related literature we control for other factors that influence the dependent variable project success (PrSuccess) and include the average number of developers (Developers) and testers (Testers) involved into the respective OSS project, perceived project complexity (Complexity), the skills level in SW development of the project leader (Skills) and the existence of business support during SW development (BusinessSupport) as independent variables. Table 4 describes our model variables (that are not already introduced in Table 2). Our estimation equation looks as follows:

\[ PrSuccess_i = \alpha + \beta_1 \cdot SecEffortDvlpmt_i + \beta_2 \cdot SecEffortDplymt_i + \beta_3 \cdot Complexity_i + \beta_4 \cdot Developers_i + \beta_5 \cdot Testers_i + \beta_6 \cdot Skills_i + \beta_7 \cdot BusinessSupport_i + \epsilon_i , \]

where \( \alpha \) is the constant term and \( \epsilon_i \) is the error term.
To validate the subjective perception of project success by the project leaders, and to deal with a potential common method bias, and to ensure the robustness of our findings, we used objective data as a second source of information. We therefore additionally collected the number of downloads from the projects’ SourceForge download website at the time of the survey, after six, and twelve months as objective market success measure. Further, we normalize the cumulative number of downloads by dividing it by the project’s lifecycle months until the time of data collection to incorporate the various distinct lifecycle phases of the SW projects varying from one to 162 months.

## Results and Discussion

We pre-analysed our data set. The variance inflation factors are well below 4 (Mean = 1.42) indicating that there is no multicollinearity in our data set. We estimate our model using ordinary least squares with robust standard errors to account for heteroscedasticity. Table 5 (Models 3-5) shows our estimation results for the perceived project success as dependent variable. The F-tests show (p < .000) that at least one of the coefficients is not zero. The adjusted R²s range from 10.5% to 22.9% indicating that our models explain quite well the variation of the project success. Enhancement of the model with the control variable business support slightly reduces the model’s goodness of fit, thus Model 4 is the most appropriate. The effect of one scale increase in the relative effort into IT security during initial development increases the project success by 1.114 scales (p < .1, standard error 0.654). The results of the stepwise regression (Models 1-3) show that this effect is robust with regard to model specification. Further, we find that the average number of testers has a significant and slightly positive effect on the perceived project success (p < .1). One additional tester that tests code of other developers increases project success by 0.001 scales. The level of complexity of the project’s source code as rated by the project leader has a highly significant and positive influence (p < .01). One scoring point increase of the SW’s complexity leads to an increase of perceived project success by 0.475 scales. No significant effect of business support on perceived project success can be observed. Also, the results show a negative, but not significant relation between the relative effort in security during deployment and perceived project success.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number downloads (objective success measure)</td>
<td># Downloads</td>
<td>Cumulative number of downloads at the time of the survey</td>
</tr>
<tr>
<td>Number developers</td>
<td># Developers</td>
<td>Average number of main developers involved into the OSS project</td>
</tr>
<tr>
<td>Number testers</td>
<td># Testers</td>
<td>Average number of developers who tested code of other developers</td>
</tr>
<tr>
<td>Relative effort into IT security during initial development phases</td>
<td>SecEffort Dvlpnt</td>
<td>Share of avg. effort into IT security relative to the avg. general development effort during initial development</td>
</tr>
<tr>
<td>Relative effort into IT security during the deployment phases</td>
<td>SecEffort Dplymnt</td>
<td>Share of average effort into IT security relative to the average general development effort during deployment</td>
</tr>
<tr>
<td>Skills</td>
<td>Skills</td>
<td>SW development skills of the project leaders 5-point Likert scale (1 = Not skilled; 5=Very skilled)</td>
</tr>
</tbody>
</table>

Table 4. Model variables

Research that uses stated information can suffer from a common method bias or variance (CMV) (Podsakoff et al. 2003). Previous research proposed several methods to control for common method bias, i.e. a systematic error variance shared among variables measured with the same source (Richardson et al.
To reliably eliminate potential CMV, we estimate our equation replacing the survey-based dependent variable with an objective measure, i.e. the logarithms of the cumulative number of downloads of the given OSS project measured at the time of the survey, after six, and twelve months.

The estimation results of the objective success measure number of projects downloads are displayed in Table 6. The model results supports our hypothesis that the relative effort invested in security in the initial SW development phases fosters the project’s success. The results of the stepwise regression (Models 1-4) show that this effect is robust with regard to model specification. The F-tests show (p < .001) that at least one of the coefficients is not zero. The adjusted R²'s range from 17.5% to 21.3% indicating that our models explain quite well the variation of the project success in terms of number of downloads (Model 3 and 4). Model 4 shows the best goodness of fit. One scoring point increase in the relative effort spend on security in the initial development stages leads to about 3.13% increase in downloads (p < .01). The effect of the relative effort in security during deployment is negative and significant. One scoring point increase in relative security effort during deployment leads to about 2.35% decrease in downloads (p < .05). Further, we find that the complexity of the source code (p < .01) and the average number of developers (p < .1) is positively related to the project success. One score increase in complexity leads to 0.9% increase in project downloads. One additional developer leads to a slight increase of 0.03% in downloads given the other variables in the model are held constant.

Astonishingly, business support has a negative effect on the number of downloads (p < .05). It reduces the number of downloads by 1.69%. Possible reasons for this negative effect are discussed in the subsequent section. No relation between the skill level of the project leader and project success can be observed. The positive highly significant effect of early security effort in development sustainably affect the success of an OSS project even when the dependent variable is replaced by the number of downloads after six and twelve months (Models 5 and 6). Also, the negative significant effect of relative security effort during deployment remains. The same applies for the other significant variables. The estimation

---

2 The cumulative number of downloads is logarithmically transformed to approximate normal distribution.
results of the normalized number of downloads, displayed in the Models 4a-6a of Table 6, differ slightly in the values of the coefficients and in the adjusted R²s, but are consistent considering the direction of the effects and the significance level of variables.

<table>
<thead>
<tr>
<th></th>
<th>(1) lnN-Downloads</th>
<th>(2) lnN-Downloads</th>
<th>(3) lnN-Downloads</th>
<th>(4) lnN-Downloads</th>
<th>(5) lnN-Downloads after 6 months</th>
<th>(6) lnN-Downloads after 12 months</th>
<th>(4a) lnN-Downloads normed</th>
<th>(5a) lnN-Downloads normed6M</th>
<th>(6a) lnN-Downloads normed12M</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecEffort Dvlpmt</td>
<td>1.787***</td>
<td>3.977***</td>
<td>3.006***</td>
<td>3.132***</td>
<td>3.288***</td>
<td>3.343***</td>
<td>2.711***</td>
<td>2.901***</td>
<td>2.990***</td>
</tr>
<tr>
<td></td>
<td>(0.849)</td>
<td>(1.096)</td>
<td>(1.093)</td>
<td>(1.099)</td>
<td>(1.074)</td>
<td>(1.071)</td>
<td>(0.947)</td>
<td>(0.963)</td>
<td>(0.977)</td>
</tr>
<tr>
<td></td>
<td>(1.211)</td>
<td>(1.174)</td>
<td>(1.151)</td>
<td>(1.163)</td>
<td>(1.154)</td>
<td>(1.043)</td>
<td>(1.070)</td>
<td>(1.086)</td>
<td>(1.086)</td>
</tr>
<tr>
<td># Developers</td>
<td>0.026</td>
<td>0.034</td>
<td>0.0321</td>
<td>0.0308</td>
<td>0.0262</td>
<td>0.0260</td>
<td>0.0254*</td>
<td>o.o012</td>
<td>o.o012</td>
</tr>
<tr>
<td></td>
<td>(0.017)</td>
<td>(0.017)</td>
<td>(0.016)</td>
<td>(0.016)</td>
<td>(0.012)</td>
<td>(0.012)</td>
<td>(0.012)</td>
<td></td>
<td></td>
</tr>
<tr>
<td># Testers</td>
<td>0.0002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>(0.001)</td>
<td>(0.001)</td>
<td>(0.001)</td>
<td>(0.001)</td>
<td>(0.001)</td>
<td>(0.001)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skills</td>
<td>0.0914</td>
<td>0.0567</td>
<td>-0.020</td>
<td>-0.033</td>
<td>-0.109</td>
<td>-0.141</td>
<td>-0.138</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.305)</td>
<td>(0.319)</td>
<td>(0.294)</td>
<td>(0.292)</td>
<td>(0.272)</td>
<td>(0.274)</td>
<td>(0.277)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ComplexityOf SW</td>
<td>0.742***</td>
<td>0.854***</td>
<td>0.791***</td>
<td>0.793***</td>
<td>0.720***</td>
<td>0.681***</td>
<td>0.696***</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.214)</td>
<td>(0.209)</td>
<td>(0.176)</td>
<td>(0.172)</td>
<td>(0.184)</td>
<td>(0.159)</td>
<td>(0.158)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Support</td>
<td>-1.690**</td>
<td>-1.678**</td>
<td>-1.670**</td>
<td>-1.567**</td>
<td>-1.562**</td>
<td>-1.568**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.712)</td>
<td>(0.656)</td>
<td>(0.644)</td>
<td>(0.184)</td>
<td>(0.184)</td>
<td>(0.159)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.637)</td>
<td>(0.708)</td>
<td>(1.397)</td>
<td>(1.411)</td>
<td>(1.289)</td>
<td>(1.275)</td>
<td>(1.233)</td>
<td>(1.211)</td>
<td>(1.220)</td>
</tr>
<tr>
<td>adj. R²</td>
<td>0.021</td>
<td>0.070</td>
<td>0.175</td>
<td>0.213</td>
<td>0.221</td>
<td>0.226</td>
<td>0.200</td>
<td>0.200</td>
<td>0.204</td>
</tr>
<tr>
<td>N</td>
<td>114</td>
<td>114</td>
<td>114</td>
<td>114</td>
<td>114</td>
<td>114</td>
<td>114</td>
<td>114</td>
<td>114</td>
</tr>
</tbody>
</table>

Standard errors in parentheses: * p < .1, ** p < .05, *** p < .01

Table 6: Estimation results number of downloads/ normalized downloads per month life length of project

Figure 5. Resulting model based on perceived and objective project success data

Figure 5 presents the resulting model with the observed and significant effects on project success (perceived success or number of downloads). We can thus not reject our hypotheses H1. The positive effect is depicted by the sign “+”. The negative effect of late security effort depicted by the sign “-” on the arrow is only significant for the number of downloads. Reasons for the lack of significance based on perceived success estimations might be that high effort into security during deployment might also be a sign of a living growing project with increasing users that demand for security.

The negative impact of business support on the number of downloads is remarkable. Reasons might be that if companies use OSS at most one version is downloaded or is directly provided by the project to
the business’ administrator. Employees would not independently download the SW. Therefore, the number of downloads will not rise evenly with the number of OSS business users.

To conclude, the results presented in this section support the idea of Security by Design and suggest that if OSS projects take security seriously from the very beginning of the development process and invest early in it, this will have a positive effect on the project success in terms of perceived success as well as the number of downloads.

6 Conclusions, Limitations, and Further Work

This study investigates how early relative effort into IT security impact the success of OSS projects compared to late effort during deployment. For this purpose we conduct an online survey among SourceForge.net project leaders to evaluate general development effort together with IT security effort during distinct phases of the projects’ lifecycle and their relation to project success.

The estimation results show positive significant effects of relative security efforts during initial development phase based on the project Lead’s perceived success. These results are supported by examining the number of downloads as an objective popularity and success metric measured at three distinct points of time in the course of the project’s lifecycle. The effect of security effort during deployment phases after a stable and well tested software version has been released is negative, but only significant based on the number of downloads. Thus, our findings support claims of Security by Design advocates that promote economic benefits of early security investments. Besides, regarding our control variables, the source code complexity is positively related to project success next to the average number of developers and testers. The existence of business support during SW development is negatively related to the number of downloads which might be caused by different software access models or fewer single user downloads via the project’s website.

Our study results suggest that developers may want to reconsider the priority they give to IT security and consider it being worth to invest into it from the very beginning to increase the number of users that consider security as important quality criteria for their software selection. The results concerning the second research question are diverse. Based on perceived project success, no significant effects of relative security effort during deployment can be observed. Yet, there is a significant and negative effect of late security effort on the number of downloads. Thus, the higher the relative effort into security in the deployment phase is, the lower is the number of downloads.

Our research approach has some limitations that provide avenues for further research. We used an online questionnaire to collect projects leaders’ subjective project data. Retrospective gathered results can be biased by recall problems, i.e. the participants might forget information, which causes uncertainties in their answers. We tried to overcome this shortcoming and even strengthen our results by including the number of downloads at distinct points of time as an objective measure. This metric is only one indicator of OSS project success, which includes drawbacks (Hahn and Zhang 2005; Rossi et al. 2010), additional OSS success metrics such as the project’s rank or the degree of activity may be included in future analyses to further strengthen the results.

Furthermore, the sharp distinction between initial development and SW deployment might be difficult to make in OSS projects. To enhance understanding and information validity, reasons and assumptions were clearly explained in the questionnaire. Additionally, we received confirmation for our procedure by conducting a preceding pilot study of OSS project leaders via interviews and telephone interviews.

Another limitation might be a bias towards particularly security aware developers in our study. Subjects who followed our survey invitation might have more interest in the topic than the average OSS project leader. This might be also reflected in the response rate of 5.53% and the relatively small sample size. Nevertheless, the results are based on a randomly selected sample of 114 projects covering distinct SW categories and our significant and with objective data validated robust results clearly point into one direction.
References


Hahn, Jungpil and Zang, Chen (2005). "An exploratory study of open source projects from a project management perspective." MIS Research Workshop, Purdue University, West Lafayette, IN.


Schreiner, G. and Rich, E. (2010). “Increasing software security through open source or closed source development? Empirics suggest that we have asked the wrong question.” In: *43rd Hawaii International Conference on System Sciences (HICSS)*. 1-10.


