computers

MDPI

Article

An Empirical Study on Security Knowledge Sharing and Learning in Open Source Software Communities

Shao-Fang Wen

Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 2815 Gjovik, Norway; shao-fang.wen@ntnu.no

Received: 19 July 2018; Accepted: 28 September 2018; Published: 1 October 2018

Abstract: Open source software (OSS) security has been the focus of the security community and practitioners over the past decades. However, the number of new vulnerabilities keeps increasing in today's OSS systems. With today's increasingly important and complex OSS, lacking software security knowledge to handle security vulnerabilities in OSS development will result in more breaches that are serious in the future. Learning software security is a difficult and challenging task since the domain is quite context specific and the real project situation is necessary to apply the security concepts within the specific system. Many OSS proponents believe that the OSS community offers significant learning opportunities from its best practices. However, studies that specifically explore security knowledge sharing and learning in OSS communities are scarce. This research is intended to fill this gap by empirically investigating factors that affect knowledge sharing and learning about software security and the relationship among them. A conceptual model is proposed that helps to conceptualize the linkage between socio-technical practices and software security learning processes in OSS communities. A questionnaire and statistical analytical techniques were employed to test hypothesized relationships in the model to gain a better understanding of this research topic.

Keywords: open source software; software security; knowledge sharing; open source software community

1. Introduction

Open source software (OSS) is based on the principle that computer programs should be shared freely among users, giving them the possibility of introducing improvements and modifications. OSS is at the core of today's information technology (IT) infrastructure and information systems; about 80% of companies run their operations on OSS [1] and 96% of applications utilize OSS as the software components [2]. OSS is developed collectively by an online community of practices (CoPs) with a strong relationship between the social and technical interactions in a decentralized and knowledge-intensive process [3,4]. Groups of volunteers participate in the communities that are essential for OSS project development. They collaborate and integrate expertise to solve particular programming problems, as well as to deliver and maintain the software that is produced by the OSS community [5–7].

OSS security has been the focus of the security community and practitioners over recent decades. Many studies have been conducted by both researchers and practitioners on the mechanisms of building security in OSS development [8]. However, the number of new vulnerabilities keeps increasing in today's OSS systems. The Blackduck 2017 Open Source Security and Risk Analysis report announced that 3623 new OSS vulnerabilities occurred in 2016—almost 10 per day on average and a 10% increase from 2015 [2]. These vulnerabilities open some of the most critical OSS projects to potential exploitation such as Heartbleed and Logjam (in OpenSSL); Quadrooter (in Android); Glibc Vulnerability (in Linux servers and web frameworks); NetUSB (in Linux kernel), and many others [9,10].

With today's increasingly important and complex OSS, lacking software security knowledge to handle security vulnerabilities in OSS development will result in breaches that are more serious in the future.

Building secure applications is a complex and demanding task that developers often face. Knowledge of software security is more than simply having a checklist or reminders of things [11]. It is about understanding the potential security risks that are induced by the software, and how to manage them [12]. Comparing with proprietary software development in enterprises, which usually involves formal training and practices about secure software development, OSS development relies mainly on the ability of participants themselves to acquire, refine, and use new aspects of security knowledge to fulfill the needs of their work in the community. Much of an OSS community's security knowledge lies within its documents, discussions, decisions, processes, and the awareness by members of other members' expertise. Both finding and learning the security requirements and practices of the project become key challenges that are highly dependent on the knowledge resources the community provides. Many OSS proponents believe that the OSS community offers significant learning opportunities from its best-practices [13,14], which are different from the education of traditional models [15,16]. However, studies that specifically explore security knowledge sharing and learning in OSS communities are scarce.

As there is still a dearth of empirical research into security knowledge learning in the context of OSS development, this study intends to fill this gap by empirically investigating factors that affect knowledge sharing and learning about software security and the relationships among them. The purpose is twofold. Firstly, we are interested in obtaining a deeper understanding of how factors complement each other in shaping security knowledge sharing and learning behavior. Secondly, we suggest a conceptual framework that includes both social (security culture) and technical (security expertise coordination) constructs to investigate how OSS communities can shape this behavior. We attempt to fulfill this purpose by utilizing a questionnaire survey and statistical analytical techniques on OSS project participants. The data analysis result is the main contribution of the paper. This is presented as a preliminary research model, which includes a set of socio-technical constructs that could potentially describe security knowledge sharing mechanisms and learning processes in OSS communities.

The rest of this paper is structured as follows. Section 2 describes the theoretical background of the research. The conceptual framework defining the constructs and hypothesized relationships are depicted in Section 3. The research method is explained in Section 4. In Section 5, we present the result of data analysis. Section 6 provides a discussion based on the result. We describe the conclusion and limitation of this study in Sections 7 and 8 respectively.

2. Theoretical Background

2.1. Knowledge Sharing

Christensen [17] defines knowledge sharing as a process that exploits existing knowledge by identifying, transferring, and applying it to solve tasks better, faster, and cheaper. It is 'the process of transferring knowledge from a person to another in an organization' [18]. Knowledge sharing is a deliberate act that makes knowledge reusable by other people through knowledge transfer [19]. It is about "how people share and use what they know" [20] and requires the active engagement of individuals in a process of interaction and learning [21]. As Nonaka [22] points out, the knowledge is created and expanded through social interaction between people and their creative activities [22]. Through knowledge sharing individuals could exchange tacit or explicit knowledge, hence, together create new knowledge [23].

Terminologies such as 'knowledge distribution' and 'knowledge transfer' are also used for referring to knowledge sharing and bring paronomasia; e.g., Haas and Hansen [24], Christensen [17], Cabrera et al. [25], Wasko and Faraj [26], and Inkpen and Tsang [27]. Although these definitions and discussions of knowledge sharing vary in different perspectives, they do deliver a similar core concept, which is using existing knowledge within the organization to solve problems, generating new learning, and empowering the organization for innovation.

2.2. Knowledge Sharing and Learning in OSS Communities

The purpose of the OSS community is essentially knowledge sharing and collaboration [28,29]. An OSS community has been considered as a social leaning CoP [30–32], which aims to establish a structure where tacit and explicit knowledge is shared and exchanged among various members within a given domain to create a collective value useful to everyone [33,34]. Developers build the software by relying on extensive peer production and through the skillful use of the software and communication tools available on the Internet [35]. They share and acquire knowledge associated with their profession. Furthermore, OSS communities have been a source of learning for participants since their creation [36], which offer 24 h, 7 days a week, 365 days support with up to date content and learning materials, and all of this provided by volunteers at no charge. Therefore, an open source community is more than about software development, but also provides a rich field to explore the process of software knowledge creation, accumulation, and dissemination [30].

Knowledge sharing and learning in open source communities have been broadly studied in the literature. Sowe et al. have introduced a knowledge-sharing model to develop an understanding of the dynamics of collaboration and how knowledge sharing is distributed over OSS development teams [37,38]. Chen, Xiaogang et al. adopted the perspective of the transactive memory system (TMS) to empirically examine the possible team cognitive mechanisms that facilitate knowledge sharing in OSS communities [39]. Their study showed that communication quality positively influences knowledge sharing and technical performance of the team. Iskoujina and Roberts investigated the factors that motivate participants to share their knowledge in OSS communities and concluded that the quality of management influences the extent to which the motivations of members actually result in knowledge sharing [40]. Chen, Xiaohong analyzed key factors affecting knowledge sharing in OSS projects, which include participative motivation, social network, and organizational culture [41,42].

Au et al. explored open-source debugging as a form of organizational learning [43], which heavily relies on adaptive learning [44] to overcome the complexity of software. Singh and Holt provided insights on how the OSS community uses the forums for learning and solving problems. They explored the motivations for joining OSS communities [36], the learning that occurs in the communities, and the challenges to learning. Hardi had a case study using Google Chrome project [45] to affirm that situated learning [46] is present among open source developers at an earlier time of a project. Hemetsberger and Reinhardt examined how knowledge sharing and learning processes develop at the interface of technology and communal structures of an OSS community [4,32]. They suggested that knowledge is shared and learned in OSS communities through the establishment of processes and technologies that enable virtual re-experience for the learners at various levels. They viewed learning in OSS communities as experiential learning whereas learning is a process whereby learning is created through the transformation of experiences as developed by Kolb [47].

3. Conceptual Framework

The conceptual framework is developed based on the author's prior ethnographic study on three OSS communities [48]. The study applied a socio-technical systems perspective [49], which systematically and holistically took into account the social context as well as technological aspects. The observation result was analyzed and categorized with social (culture and organization structure) and technical (method and machine) aspects. Figure 1 depicts the conceptual and theoretical structure that includes four constructs, namely: security culture, expertise coordination, security knowledge sharing, and software security learning. The background of the conceptual framework is described below.

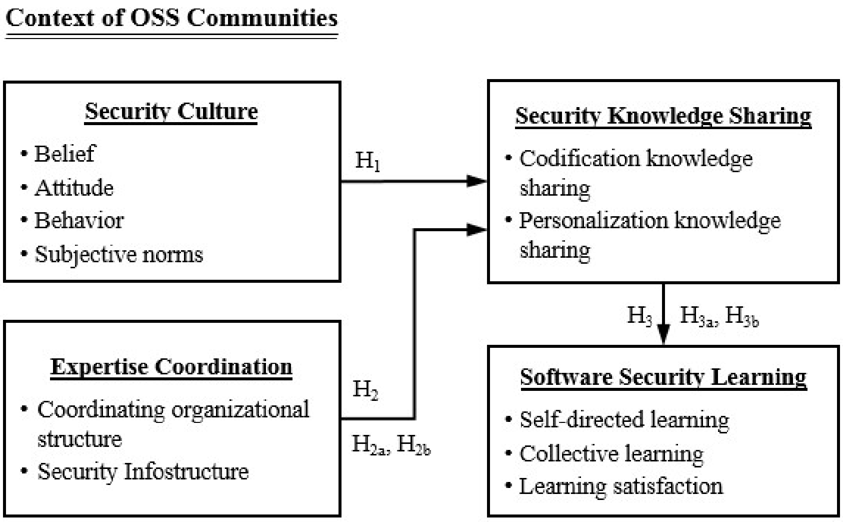

Figure 1. The conceptual framework.

3.1. Security Culture and Security Knowledge Sharing

Security culture is recognized in the security community and scientific literature as one of the most important foundations of organizational security. In short, security culture is the way our minds are programmed to create different patterns of thinking, feeling, and actions for providing the security process [50]. Security culture is based on the interaction of people with information assets and the security behavior they exhibit within the context of the organizational culture in the organization [51]. Security culture involves identifying security-related ideas, beliefs, and values of the group, which shape and guide security-related behaviors [52]. Martins and Eloff define information security culture as the perceptions, attitudes, and assumptions that are accepted and encouraged by employees in an organization in relation to information security [53]. Ngo et al. suggest that security culture is the accepted behavior and actions of employees and the organization as a whole, as well as how things are done in relation to information security [54]. Therefore, the four main aspects of security culture formed in this study are:

Belief: An acceptance or a firmly held opinion that security is of value to the community.
Attitude: A feeling or emotion toward various activities that pertain to the security of the software product produced by the community.
Behavior: Actual or intended activities and risk-taking actions in secure software developments.
Subjective norms: A combination of perceived expectations from relevant individuals or groups along with intentions to comply with security-related expectations.

Organizational culture has been shown to influence the success of knowledge management practices [55–58]. Culture shapes what a group defines as relevant knowledge, and this directly affects the knowledge a unit focuses on [57]. In the context of information security, security culture decides how much security knowledge is disseminated within the community and what knowledge learners can learn. The security culture backgrounds either at organizational or individual levels impact on the amount of security knowledge transferred within the community, further affecting the participants' learning processes. Thus, the research hypothesis is as follows:

Hypothesis 1. *Security culture is positively associated with security knowledge sharing.*

3.2. Expertise Coordination and Security Knowledge Sharing

Expertise coordination is the process of knowledge integration and the outcome of exchanging and combining knowledge through interactions among team members [59,60]. Expertise coordination is believed to serve as an important component of software development. According to the findings of empirical studies in the literature, expertise coordination strongly influences project performance, team effectiveness, and team efficiency in software development projects [61–64]. This has a bearing on both physical and virtual development teams [65–67]. For complex non-routine intellectual tasks, expertise coordination (the management of knowledge and skill of dependencies) is necessary so that the software team can recognize where expertise is located, needed, and accessed [61]. A great challenge of security expertise coordination is to combine explicit and tacit knowledge in all management and security expert decisions, and to get knowledge moved from individuals within the whole organization between different actors, and from tacit domain to explicit domain and also vice versa [68]. In this study, expertise coordination is manifested through the two following strategies: coordinating organizational structure and security infostructure.

3.2.1. Coordinating Organizational Structure

The organizational structure supports the assignment of both technical and human resources to the tasks that must be done and provide mechanisms for their coordination [69]. It also establishes and enables strategic- and operational decision-making, monitoring of performance, and operating mechanisms that transfer directives on what is expected of organizational members and how the directives should be followed [69]. The organizational challenges faced by OSS projects are significant because the project must deal not only with problems faced by any software development process, but also with the complexity of coordinating efforts of a geographically distributed base of volunteers working on the software [70]. OSS projects usually utilize security experts to define security requirements and best practices, help perform code reviews, and provides the necessary education for the software development staff [71]. The coordinating organizational structure serves as a subject matter expert to ensure that security-related issues receive necessary attention in the community. Through this structural mechanism, the security knowledge is able to gain valuable insights from the organization to facilitate strategic decision making [72].

3.2.2. Security Infostructure

The term infostructure is commonly used to describe the infrastructure of information that is used in multiple disciplines. As indicated by Tilton, an infostructure is the layout of information in a manner such that it can be navigated—it is what is created any time an amount of information is organized in a useful fashion [73]. In the knowledge sharing process, infostructure serves as a role to provide rules, which govern the exchange between the actors on the network providing a set of cognitive resources (metaphors, common language) whereby people make sense of events on the network [74]. In the context of OSS development, developers contribute from around the world, meet face-to-face infrequently if at all, and coordinate their activity primarily by means of digital channels on the internet [75,76]. A proper infostructure can help learners identify the location of the security information, knowing where an answer to a problem is located, and acquiring as much knowledge as possible [77].

Based on the above discussion, the research hypotheses are given as follows:

Hypothesis 2. *Expertise coordination is positively associated with security knowledge sharing.*

Hypothesis 2a. *Coordinating organizational structure has a positive effect on security knowledge sharing.*

Hypothesis 2b. *Infostructure has a positive effect on security knowledge sharing.*

3.3. Security Knowledge Sharing and Software Security Learning

Learning may be the most strategically valuable dynamic capability [78]. Learning is the process by which knowledge comes into being and is enhanced over time, and is therefore intimately associated with knowledge sharing process [79]. Learning experts argue that online knowledge sharing can be regarded as an important form of collective learning [80]. In OSS projects, the fundamental functionality for security knowledge sharing is to capture security experts' knowledge in the project repository that other project participants can access and learn about software security. Knowledge sharing can be facilitated by the project-based organization by using codification or personalization mechanisms [81,82].

3.3.1. Codification Security Knowledge Sharing

The codification knowledge sharing mechanism captures individual or group-held knowledge and makes it the wider property of the organization [81], which facilitates a setting for participants to exercise self-directed learning. The basic functionality for this knowledge sharing mechanism is to capture security experts' knowledge in the project repository that other project participants can access and learn about software security, which provides a setting for participants to exercise self-directed learning. Moreover, the internet resources have the advantage to provide the community with an information infrastructure for sharing codification materials of software development in the form of hypertext, video, and a software artifact content indexes or directories. These codification materials (documentation, wiki, release notices, security advisories, source code, etc.) provides the participants with a shortcut for obtaining an overview of the system or for understanding the code that provides a particular feature. At the very least, it includes instructions on how to get started and details of where to find more information.

3.3.2. Personalization Security Knowledge Sharing

Personalization knowledge sharing provides communications in another form, as it is concerned with the use of people as a mechanism for sharing knowledge [83]. Personalization as a knowledge sharing mechanism has the inherent flexibility of transmitting tacit knowledge, and allowing for discussions and sharing interpretations that may lead to the development of new knowledge [81]. OSS communities adopt various forms of technologies, such as mailing list, forum, and Internet Relay Chat (IRC) to support knowledge sharing via personalization mechanisms. These technologies provide useful means of storage and acquisition for the communities' experiential knowledge, given that individuals have a general preference for obtaining information from other people, rather than from documents [84]. Although in OSS development, a programmer may write a complete program independently from other programmers, the software code will be still examined by other software engineers. Coding review also represents a form of personalization knowledge sharing mechanism in which knowledge is created collectively in a distributed work process. Peer review process emphasizes the importance of collecting learning and shared dialogue [30]. During code review, questions, answers, and discussion about the coding issues are communicated back and forth between the community and the members. Developers have the opportunities to reflect their code, to take corrective actions and build concrete experiences in the code review process.

Based on the above discussion, the following hypotheses were made:

Hypothesis 3. *Security knowledge sharing is positively associated with software security learning.*

Hypothesis 3a. *Codification knowledge sharing has a positive effect on software security learning.*

Hypothesis 3b. *Personalization knowledge sharing has a positive effect on software security learning.*

4. Methodology

This research adopted a quantitative approach to a survey research method to investigate the relationships among security culture, expertise coordination, security knowledge sharing, and software security learning. A self-administered Web-based survey was used to collect individual-level perception data from participants in OSS projects. The use of an OSS participant survey was deemed appropriate to test the hypotheses outlined in the previous section.

4.1. Instruments

The survey instrument used in this study was the outcome of an iterative process of checking and refinement. The constructs and items used to operationalize the research were developed following the generally accepted guidelines of reliability and validity or multiple-item measures [85]. After synthesizing the results of the literature review, a questionnaire was developed based on the structure of the research framework. Some survey questions were inspired by existing studies, while others were created specifically to suit the research context of our study. For the measurement instrument of key variables, each item was measured on a five-point Likert scale (Cf. Appendix A). The primary references for the constructs and items used in this study are summarized in Table 1.

Table 1. Measurement instrument for key variables in the questionnaire.

Construct		Item	Reference
Security culture	Belief	Security value, cognition	[51–54,86]
	Attitude	Risk-taking, responsibility	
	Behavior	Secure coding, compliance	
	Subjective Norms	Peer influence, expectation	
Expertise coordination	Coordinating structure	Security expert, assistance	[61,74,77,87]
	Security Infostructure	security website, navigation, taxonomy	
Security knowledge sharing	Codification knowledge sharing	Documentation, multimedia,	[22,81,88,89]
	Personalization knowledge sharing	Experience, collaboration	
Software security learning	Self-directed learning	Exploration, search	[90–92]
	Collective learning	Feedback, problem-solving	
	Learning satisfaction	Enjoyment, simplicity	

4.2. Data Collection

Samples for the empirical study were randomly collected from participants in OSS development projects, available on GitHub. GitHub is an online database of open source software projects. Users and potential contributors can access information about the projects and download current versions of the software being developed. As of April 2017, GitHub reports having almost 20 million users and 57 million repositories [93], making it the largest host of source code in the world [94]. The anonymous questionnaires were sent via e-mail to a list of OSS participants at the beginning of August 2017. Data collection period lasted 3 months and 402 questionnaires were completed. Among them, 324 were valid; and another 78 respondents were discarded due to the reason that they did not participate in any open source community. Table 2 shows demographic information about the sample, which includes gender, age, and the seniority in the community and product categories of the projects.

Table 2. Demographic characteristics of the respondents (n = 324).

Item	Category	Frequency	Percentage
Gender	Male	289	89.2%
	Female	23	7.1%
	Prefer not to say	12	3.7%
Age	<20	13	4.0%
	20–30	147	45.4%
	31–40	116	35.8%
	41–50	35	10.8%
	>50	7	2.2%
	Prefer not to say	6	1.9%
Seniority in the community	<3 months	13	4.0%
	3–6 months	17	5.2%
	7 months–1 years	47	14.5%
	2–3 years	89	27.5%
	>3 years	158	48.8%
Product Category	Healthcare, Health Tech	12	3.7%
	Science, Geospatial, Astronomy	9	2.8%
	Retail & E-Commerce	7	2.2%
	Big Data, AI, BI, Machine Learning	22	6.8%
	Enterprise Software	11	3.4%
	Mobile Apps	19	5.9%
	Gaming, Entertainment, Media	13	4.0%
	Financial Services	15	4.6%
	Development Framework	35	10.8%
	Internet, email, browser, content management	43	13.3%
	Database, file system	30	9.3%
	Security, firewall, anti-virus, encryption	27	8.3%
	Operating system	21	6.5%
	Education, knowledge management, eLearning	19	5.9%
	Internet of things	28	8.6%
	Others	13	4.0%

4.3. Reliability and Validity Analysis

Validating constructs is important before any further analysis is conducted. To this end, reliability and validity tests were carried out following the sequence and approach that was taken by Straub [95]. Table 3 outlines the results of the reliability and validity tests performed on the survey items. Convergent validity, the degree to which multiple attempts to measure the same concept are in agreement, was evaluated by examining the factor loading within each construct, composite reliability, and variance extracted [96,97]. We used confirmatory factor analysis (CFA) with AMOS to examine the convergent validity of each construct. The factor loadings range from 0.493 to 0.872, and these are greater than the recommended level of 0.35, which is based on 250 samples and a 0.05 significance level [97]. All composite reliabilities and variance-extracted measures of constructs exceed the recommended level of 0.8 and 0.5 each. Reliability of a scale (factor or construct) is to examine its internal consistency by calculating Cronbach's alpha. This method indicates the extent to which items within a scale are homogenous or correlated [98,99]. It is also reflective of the consistency between different items on a scale, in measuring the same attribute. The resulting alpha values ranged from 0.827 to 0.907, which were above the acceptable threshold (0.70) suggested by Nunnally [85]. From the analyses mentioned above, it was found that the survey items on each construct met the requirements for reliability and validity.

Table 3. The convergent validity and reliability test results.

Construct		Item	Convergent Validity (Factor Loading [1])	Reliability (Cronbach's α)
Security culture	Belief	Security value Cognition	0.727 0.651	0.873
	Attitude	Risk-taking Responsibility	0.736 0.814	
	Behavior	Secure coding Compliance	0.801 0.735	
	Subjective norms	Peer influence Expectation	0.781 0.665	
Expertise coordination	Coordinating structure	Security expert Assistance	0.674 0.523	0.827
	Security infostructure	Security website Navigation	0.818 0.798	
Security knowledge sharing	Codification knowledge sharing	Documentation Multimedia	0.746 0.812	0.907
	Personalization knowledge sharing	Experience Collaboration	0.728 0.727	
Software security learning	Self-directed learning	Exploration Search	0.831 0.736	0.883
	Collective learning	Feedback Problem-solving	0.753 0.851	
	Learning satisfaction	Enjoyment Simplicity	0.493 0.627	

[1] Factor loadings are from confirmatory factor analysis.

5. Analysis and Result

Statistic software SPSS 24.0 for Windows was used to analyze the data. Pearson's correlation analysis and multiple regression analysis were to analyze security culture, expertise coordination, security knowledge sharing, and software security learning.

5.1. Relationship between Security Culture and Security Knowledge Sharing

This study adopted Pearson's correlation analysis to determine the correlation between security culture and security knowledge sharing. Table 4 shows that the correlation coefficient between security culture and security knowledge sharing is 0.671, a highly positive correlation. The correlation coefficients of each of the security culture factors—belief, attitude, behavior, and subjective norms are 0.591, 0.628, 0.427, and 0.584 respectively. Regarding correlation among all security culture factors, the results show a strong correlation among them that reaches a significant level ($p < 0.01$). Thus, security culture has a significant positive correlation with security knowledge sharing. Hence, Hypothesis 1 is proven.

5.2. Relationship between Expertise Coordination and Security Knowledge Sharing

Table 5 indicates that expertise coordination has a significant positive correlation with security knowledge sharing in which Pearson correlation is 0.400 and $p < 0.01$. The correlation coefficients of expertise coordination factors-coordinating organizational structure and infostructure are 0.628 and 0.584 respectively. The results showed a strong correlation among all expertise coordination factors that reached a significant level ($p < 0.01$). Consequently, the research result favored Hypothesis 2, the stronger coordinating organizational structure and security infostructure, the higher the security knowledge sharing degree. Hence, H2a and H2b are also proven valid.

Table 4. The correlation analysis for security culture and security knowledge sharing.

		Security Knowledge Sharing
Security culture	Pearson correlation Sig. (2-tailed)	0.671 ** 0.000
Belief	Pearson correlation Sig. (2-tailed)	0.591 ** 0.000
Attitude	Pearson correlation Sig. (2-tailed)	0.628 ** 0.000
Behavior	Pearson correlation Sig. (2-tailed)	0.427 ** 0.000
Subjective norms	Pearson correlation Sig. (2-tailed)	0.584 ** 0.000

** Correlation is significant at the 0.01 level (2-tailed).

Table 5. The correlation analysis for expertise coordination and security knowledge sharing.

		Security Knowledge Sharing
Expertise coordination	Pearson correlation Sig. (2-tailed)	0.400 ** 0.000
Coordinating organizational structure	Pearson correlation Sig. (2-tailed)	0.376 ** 0.000
Security infostructure	Pearson correlation Sig. (2-tailed)	0.370 ** 0.000

** Correlation is significant at the 0.01 level (2-tailed).

Since expertise coordination has a significant correlation with security knowledge sharing, this study used multiple-regression analysis to understand the linear relationship between a group of forecast variable and a valid variable. The multiple-regression analysis used in this research is shown in Table 6. As indicated in the table, B value, Beta, and *t*-value have positive values. The prediction equation is based on the unstandardized coefficients, as follows: $y_1 = 2.418 + 0.151x_3 + 0.217x_4$ (where x_3 is coordinating organizational structure and x_4 is security infostructure). All variables show a positive relationship. Looking at the *p*-value for each variable, the predictor variables of coordinating organizational structure and security infostructure not statistically significant because of both of their *p*-value greater than 0.05. In this model, the two factors do not provide a significant impact on security knowledge sharing. Thus, given the above relationship, Hypotheses 2a and 2b are partially supported.

Table 6. The multiple-regression analysis for expertise coordination on security knowledge sharing.

Model 1	**Unstandardized Coefficients**		**Standardized Coefficients**			**Collinearity Statistics**	
	B	Std. Error	Beta	t	Sig.	Tolerance	VIF
(Constant)	2.418	0.217		9.878	000		
Coordinating organizational structure	0.151	0.085	0.128	1.768	0.078	0.414	2.416
Security infostructure	0.217	0.086	0.217	2.514	0.013	0.446	2.244

Dependent Variable: Security knowledge sharing.

5.3. Relationship between Security Knowledge Sharing and Software Security Learning

Table 7 indicates that security knowledge sharing has a significant positive correlation with software security learning in which the Pearson correlation is 0.578 and $p < 0.01$. The correlation coefficients of security knowledge sharing factors–codification knowledge sharing and personalization knowledge sharing are 0.491 and 0.455 respectively. The results showed a strong correlation among all security knowledge sharing factors that reached a significant level ($p < 0.01$). Thus, security knowledge sharing had a significant positive correlation with software security learning. Consequently, the research result favored Hypothesis 2, the stronger codification and personalization knowledge sharing about software security, the higher the security learning level. Hence, H3a and H3b are also proven valid.

Table 7. The correlation analysis for security knowledge sharing and software security learning.

		Software Security Learning
Security knowledge sharing	Pearson correlation Sig. (2-tailed)	0.578 ** 0.000
Codification knowledge sharing	Pearson correlation Sig. (2-tailed)	0.491 ** 0.000
Personalization knowledge sharing	Pearson correlation Sig. (2-tailed)	0.455 ** 0.000

** Correlation is significant at the 0.01 level (2-tailed).

Table 8 shows the result of the multiple regression analysis. As indicated in the table, B value, Beta, and t-value have positive values. The prediction equation is based on the unstandardized coefficients, as follows: $y_2 = 0.652 + 0.362x_5 + 0.216x_6$ (where x_5 is security knowledge sharing and x_6 is software security learning). All variables show a positive relationship. Looking at the p-value for each variable, we can see that the predictor variables of codification knowledge sharing and personalization knowledge sharing are significant because both of their p-value are smaller than 0.05. This indicates that the regression model fits the data or there is a significant relationship between predictor variables (Codification knowledge sharing and Personalization knowledge sharing) and dependent variables (Software security learning). It also appears multicollinearity is not a concern because the VIF scores are both less than three. It shows a positive sign which indicates a positive linear relationship and the result is statistically significant. Thus, given the above relationship, Hypotheses 3a and 3b are proven valid.

Table 8. The multiple-regression analysis for security knowledge sharing on software security learning.

Model 2	Unstandardized Coefficients		Standardized Coefficients			Collinearity Statistics	
	B	Std. Error	Beta	t	Sig.	Tolerance	VIF
(Constant)	0.652	0.257		2.539	0.012		
Codification knowledge sharing	0.362	0.056	0.361	6.46	0.000	0.823	1.215
Personalization knowledge sharing	0.216	0.069	0.196	3.139	0.002	0.661	1.514

Dependent Variable: Software security learning.

6. Discussion

In this study, the research hypotheses are proposed with a conceptual framework, which was validated through conducting empirical examinations including survey question design, questionnaire data collection, validity and reliability testing, and correlation and linear regression analysis among 22 items in 324 valid questionnaires. The testing results of the research hypotheses are summarized in Table 9.

Table 9. Testing results of research hypotheses.

Hypothesis	Result
H1. *Security culture is positively associated with security knowledge sharing.*	Supported
H2. *Expertise coordination is positively associated with security knowledge sharing.*	Supported
H2a. *Coordinating organizational structure has a positive effect on security knowledge sharing.*	Partially supported
H2b. *Infostructure has a positive effect on security knowledge sharing.*	Partially supported
H3. *Security knowledge sharing is positively associated with software security learning.*	Supported
H3a. *Codification knowledge sharing has a positive effect on software security learning.*	Supported
H3b. *Personalization knowledge sharing has a positive effect on software security learning.*	Supported

According to the result of the Pearson's correlation analysis (Table 4), there is a significant positive relation between security culture and security knowledge sharing. This means that if an OSS project truly holds the value that software security is important, then particular security knowledge sharing behaviors and actions can be expected. The more perceived normative support for security culture in their community means that participants are more likely to perform exemplary secure behaviors and avoid risk. As the security culture would certainly influence the operation activities of security knowledge sharing and further impact on the effectiveness of software security learning, the community should regard security culture as an important factor for supporting and guiding security practices.

Regarding the relation between expertise coordination and security knowledge sharing, this study finds that security expertise coordination is associated with the degree of security knowledge sharing. According to the Pearson's correlation analysis (Table 5), there is a significant positive relation between security expertise coordination and security knowledge sharing. Moreover, when the factors of expertise coordination are more significant, they meaningfully affect security knowledge sharing, as evidenced by the significant variance explained by the regression analysis (Table 6). This implies that if the factors of expertise coordination—coordinating organizational structure and security infostructure are more efficient and effective—they can significantly enhance security knowledge sharing. Although the two factors do not have a significant correlation with security knowledge sharing in the regression model, they still have positive coefficients. Achieving a successful software system requires tight coordination among the various efforts involved in the software development cycle [64]. If OSS communities can provide an internal security consulting organization with dedicated responsible people for security activities, and place the security information in a structured and collected manner, it will lead to a knowledge sharing arrangement actually being established.

On the other hand, our regression model also provides strong support for a significant contribution of security knowledge sharing to the software security learning process. The result of the Pearson's correlation analysis (Table 7) shows a significant positive relation between security knowledge sharing and software security learning. Moreover, as evidenced by the significant variance explained by the regression analysis (Table 8), while codification and personalization knowledge sharing are more significant, software security learning is significantly and positively affected. In the context of OSS communities, codification can be a good mechanism to store large amounts of security knowledge on the project website and to create an organizational memory for all participants. The method of personalization knowledge sharing reflects security experts' experience (via the forum, mailing list, code review etc.) which collectively produces knowledge that can be spread further to the individuals or the whole team. The two knowledge-sharing mechanisms create a digital pipeline or an intelligent link for knowledge building that appears to support the software security learning process. As the community provides opportunities for its members to share security knowledge or experiences with others, which increases the amount of knowledge sharing, it should stimulate the software security learning.

7. Conclusions

This empirical study focuses on investigating the organizational practices and behaviors that affect knowledge sharing and learning about software security in OSS communities, and the relationships among them. OSS has become a critical component and a key competency of information and communication technology (ICT) ecosystems. While the number of found vulnerabilities in OSS is increasing, it is noteworthy that effective learning about security knowledge in the context of OSS development has not gained much attention. Thus, it is necessary to examine how the security knowledge is transferred and acquired by OSS participants.

As Scacchi points out, the meaning of open source in the socio-technical context is broader than its technical definition and includes communities of programming practice, organizational culture and structure, and technical practices [100]. This can be viewed as a necessary condition within a learning framework as both social and technical aspects are of equal importance. This research proposes a model that helps conceptualize the linkage between such socio-technical practices and software security learning process in OSS communities. We gathered empirical evidence from 324 questionnaires and quantitatively analyzed data to test the hypothesized relationships in the model.

The statistical analysis shows that both security culture and the coordination of expertise can positively influence and contribute to security knowledge sharing at a certain level in OSS communities. Security culture provides a strong indication of a participant's disposition to act. It is important because unless the community believes that security is valuable to the software product, participants are unlikely to work securely and exchange their experiences in the field of software security. Indeed, every member involves in OSS development should be concerned with software security, but it is inefficient to demand each participant taking care of all security aspects. Hence, in order to enhance security knowledge sharing, a community should cultivate a culture that engages dialogue and interest among participants in order to promote the value of software security to their products and raise awareness. If OSS communities can nurture a security culture, it will be easy for them to create an environment where developers and users are willing to share and talk about software security, providing the opportunity to draw lessons from each other's experiences.

On the other hand, as OSS and its communities continue to grow in size and complexity, security expertise coordination within the community plays a larger role in security governance. While security information is provided with an adequate coordinating structure and infostructure support in the community, the implementation of security knowledge sharing throughout the community can be instilled in its culture. This study also concludes that the learning process (self-directed and collective learning) of software security and learning satisfaction are definitely influenced by security knowledge sharing. It indicates that the successful sharing of security knowledge in the OSS community, either through codification or personalization mechanisms, will enable software security learning to flow through an entire community.

People join the OSS community at different ages and have different backgrounds, capacities, and resources, as well as different objectives. They come from many disciplines which might lack formal, college-level software security training, and therefore do not see any economic incentive for squeezing security thinking into their work to produce secure codes. On the other hand, learning software security is a difficult and challenging task as the domain is rather context-specific, and the real project situation is necessary to apply the security concepts within the specific system. It is suggested that OSS communities must establish beliefs and norms, as well as roles and knowledge facilities for secure software developments; i.e., to offer environments and opportunities for security knowledge sharing and the development of software security knowledge for participants as well on the horizontal level between the experienced (but ever-learning) community members.

Ultimately, the contributions of this research supply researchers with a conceptual framework for software security knowledge sharing and learning in the OSS community in a thorough manner, providing a context in which to operate. The study also provides other researchers a firm basis to develop new security learning approaches for OSS communities, addressing many of the identified limitations.

8. Limitations

Several limitations of this research should be noted. Despite a rigorous examination of the trustworthiness of the collected data, this study might have some method bias. First, the samples were chosen opportunistically from GitHub projects, and the number of responses obtained from the survey was rather small compared with today' enormous OSS projects and field workers. Second, even though there are other known human factors that facilitate security knowledge sharing behaviors in organizations as Safa and Von Solms suggested, this study did not consider factors such as motivation or intention in OSS communities [101]. Thus, there is a need for further research efforts focused on accumulating more evidence that is empirical and data to break through the limitations. These efforts should improve the generalizability of this study to the entire OSS development phenomenon by considering a larger number of responses covering a range of diverse OSS projects. In addition, special attention should be geared toward finding the human factors, which affect independent variables such as reputation, self-efficacy, and promotion.

Funding: This research received no external funding.

Conflicts of Interest: The author declares that they have no conflict of interest.

Ethical Approval: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards Informed consent was obtained from all individual participants included in the study.

Appendix A

The items that were used in the questionnaire are presented as follows:

Item	Question
Value	Software security is of value to the community.
Cognition	I am confident that the project can govern the security quality of the software product.
Responsibility	Software security is an important part of my work in the project(s).
Risk-taking	When I do my work, I assume that the software might be misused actively to reveal bugs, and that bugs could be exploited maliciously.
Secure coding	I always make the software components behave in a predictable manner despite unexpected inputs or user actions.
Compliance	I always adhere to the security guideline.
Peer influence	Members of the community help each other solve security issues.
Expectation	I am encouraged to work securely by members in the community.
Security expert	There is a security team (or at least one member) in the community, who provides documentation about software security (e.g., secure coding practices, vulnerability information, etc.)
Assistance	There is a security team (or at least one member) in the community, who provides assistance for participants in resolving security issues.
Security site	There is a dedicated internet website related to software security in the community.
Navigation	The security information is available in a structured and collected manner in the community.
Documentation	The community saves and renews security information in the project website.

Item	Question
Multimedia	The community transfers security knowledge through words, pictures, or video.
Experience	Members are willing to share their experience and knowledge about software security.
Collaboration	Members help each other solve security problems.
Exploration	I learn software security by exploring the project repository (source code, documentation, wiki, etc.)
Search	I browse mailing list, forum, blog or other information channels of the community to learn about software security.
Feedback	I learn coding errors or security vulnerabilities by receiving comments from code reviews (pull request).
Problem solving	I learn software security through discussions in the community.
Enjoyment	I enjoy learning software security in the community.
Simplicity	It is easy for me to find the security information in the community.

References

1. NorthBridge, B. Future of Open Source Survey, Electronic Document. 2016. Available online: https://www.slideshare.net/blackducksoftware/2016-future-of-open-source-survey-results (accessed on 11 June 2018).
2. BlackDuck Software. Open Source Security and Risk Analysis. 2017. Available online: https://www.blackducksoftware.com/open-source-security-risk-analysis-2017 (accessed on 11 June 2018).
3. Kogut, B.; Metiu, A. Open-source software development and distributed innovation. *Oxf. Rev. Econ. Policy* **2001**, *17*, 248–264. [CrossRef]
4. Hemetsberger, A.; Reinhardt, C. Learning and knowledge-building in open-source communities: A social-experiential approach. *Manag. Learn.* **2006**, *37*, 187–214. [CrossRef]
5. Scacchi, W.; Feller, J.; Fitzgerald, B.; Hissam, S.; Lakhani, K. Understanding free/open source software development processes. *Softw. Process Improv. Pract.* **2006**, *11*, 95–105. [CrossRef]
6. Feller, J.; Fitzgerald, B. *Understanding Open Source Software Development*; Addison-Wesley: London, UK, 2002.
7. Feller, J.; Finnegan, P.; Kelly, D.; MacNamara, M. Developing open source software: A community-based analysis of research. In *Social Inclusion: Societal and Organizational Implications for Information Systems*; Springer: Boston, MA, USA, 2006; pp. 261–278.
8. Wen, S.-F. Software Security in Open Source Development: A Systematic Literature Review. In Proceedings of the 21st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 6–10 November 2017.
9. Pittenger, M. Know your open source code. *Netw. Secur.* **2016**, *2016*, 11–15. [CrossRef]
10. Levy, J. Top Open Source Security Vulnerabilities, WhiteSource Blog. Available online: https://www.whitesourcesoftware.com/whitesource-blog/open-source-security-vulnerability/ (accessed on 22 February 2017).
11. Barnum, S.; McGraw, G. Knowledge for software security. *IEEE Secur. Priv.* **2005**, *3*, 74–78. [CrossRef]
12. McGraw, G. *Software Security: Building Security in*; Addison-Wesley Professional: Boston, MA, USA, 2006; Volume 1.
13. Hippel, E.V.; Krogh, G.V. Open source software and the private-collective innovation model: Issues for organization science. *Organ. Sci.* **2003**, *14*, 209–223. [CrossRef]
14. Lakhani, K.R.; Von Hippel, E. How open source software works:free user-to-user assistance. *Res. Policy* **2003**, *32*, 923–943. [CrossRef]
15. Cerone, A.; Sowe, S.K. Using free/libre open source software projects as e-learning tools. *Electron. Commun. EASST* **2010**, *33*. [CrossRef]
16. Fernandes, S.; Martinho, M.H.; Cerone, A.; Barbosa, L.S. Integrating Formal and Informal Learning through A FLOSS-Based Innovative Approach. In Proceedings of the International Conference on Collaboration and Technology, Wellington, New Zealand, 30 October–1 November 2013.
17. Holdt Christensen, P. Knowledge sharing: Moving away from the obsession with best practices. *J. Knowl. Manag.* **2007**, *11*, 36–47. [CrossRef]

18. Park, H.S.; Im, B.-C. A Study on the Knowledge Sharing Behavior of Local Public Servants in Korea: Structural Equation Analysis. In Proceedings of the Joint Conference of the Tenth International Conference on Advances in Managerment, Seoul, Korean, July 2003; Korean Association for Public Administration, & Korea Institute of Public Administration; pp. 50–63. Available online: http://www.dbpia.co.kr/Journal/ArticleDetail/NODE06711930 (accessed on 30 September 2018).
19. Lee, C.K.; Al-Hawamdeh, S. Factors impacting knowledge sharing. *J. Inf. Knowl. Manag.* **2002**, *1*, 49–56. [CrossRef]
20. Koulopoulos, T.M.; Frappaolo, C. *Smart Things to Know about Knowledge Management*; Capstone Press: Dover, NH, USA, 1999.
21. Roberts, J. From know-how to show-how? Questioning the role of information and communication technologies in knowledge transfer. *Technol. Anal. Strateg. Manag.* **2000**, *12*, 429–443. [CrossRef]
22. Nonaka, I. *The Knowledge-Creating Company*; Harvard Business Review Press: Boston, MA, USA, 2008.
23. Van den Hooff, B.; Elving, W.; Meeuwsen, J.M.; Dumoulin, C. Knowledge sharing in knowledge communities. In *Communities and Technologies*; Springer: Dordrecht, The Netherlands, 2003.
24. Haas, M.R.; Hansen, M.T. Different knowledge, different benefits: Toward a productivity perspective on knowledge sharing in organizations. *Strateg. Manag. J.* **2007**, *28*, 1133–1153. [CrossRef]
25. Cabrera, A.; Collins, W.C.; Salgado, J.F. Determinants of individual engagement in knowledge sharing. *Int. J. Hum. Resour. Manag.* **2006**, *17*, 245–264. [CrossRef]
26. Wasko, M.M.; Faraj, S. Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. *MIS Q.* **2005**, 35–57. [CrossRef]
27. Inkpen, A.C.; Tsang, E.W. Social capital, networks, and knowledge transfer. *Acad. Manag. Rev.* **2005**, *30*, 146–165. [CrossRef]
28. Shen, X. Developing Country Perspectives on Software: Intellectual Property and Open Source. *Stand. Res. Inf. Technol. New Perspect.* **2007**, *3*, 21–43. [CrossRef]
29. Lee Endres, M.; Endres, S.P.; Chowdhury, S.K.; Alam, I. Tacit knowledge sharing, self-efficacy theory, and application to the Open Source community. *J. Knowl. Manag.* **2007**, *11*, 92–103. [CrossRef]
30. Sowe, S.K.; Karoulis, A.; Stamelos, I. A constructivist view of knowledge management in open source virtual communities. In *Managing Learning in Virtual Settings: The Role of Context*; Figueiredo, D.A., Paula, A., Eds.; Idea Group Inc.: Hershey, PA, USA, 2005; pp. 290–308.
31. Mirbel, I. OFLOSSC, an Ontology for Supporting Open Source Development Communities. In Proceedings of the ICEIS, Milan, Italy, 6–10 May 2009.
32. Hemetsberger, A.; Reinhardt, C. Sharing and Creating Knowledge in Open-Source Communities: The Case of KDE. In Proceedings of the Paper for Fifth European Conference on Organizational Knowledge, Learning, and Capabilities, Innsbruck, Austria, 2–3 April 2004.
33. Lave, J.; Wenger, E. *Situated Learning: Legitimate Peripheral Participation*; Cambridge University Press: Cambridge, UK, 1991.
34. Wenger, E. Communities of practice and social learning systems. *Organization* **2000**, *7*, 225–246. [CrossRef]
35. Lanzara, G.F.; Morner, M. The knowledge ecology of open-source software projects. In Proceedings of the 19th EGOS Colloquium, Copenhagen, Denmark, 3 July 2003.
36. Singh, V.; Holt, L. Learning and best practices for learning in open-source software communities. *Comput. Educ.* **2013**, *63*, 98–108. [CrossRef]
37. Sowe, S.K.; Stamelos, I.; Angelis, L. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *J. Syst. Softw.* **2008**, *81*, 431–446. [CrossRef]
38. Sowe, S.K.; Ghosh, R.; Soete, L. Annals of Knowledge Sharing in Distributed Software Development Environments: Experience from Open Source Software Projects. In Proceedings of the 33rd Annual IEEE Software Engineering Workshop, Skövde, Sweden, 18 August 2009.
39. Chen, X.; Li, X.; Clark, J.G.; Dietrich, G.B. Knowledge sharing in open source software project teams: A transactive memory system perspective. *Int. J. Inf. Manag.* **2013**, *33*, 553–563. [CrossRef]
40. Iskoujina, Z.; Roberts, J. Knowledge sharing in open source software communities: Motivations and management. *J. Knowl. Manag.* **2015**, *19*, 791–813. [CrossRef]
41. Chen, X.; Probert, D.; Zhou, Y.; Su, J. Mechanisms of knowledge sharing in open source software projects: A comparison of Chinese and Western practice. *Int. J. Technol. Intell. Plan.* **2016**, *11*, 117–139. [CrossRef]

42. Chen, X.; Zhou, Y.; Probert, D.; Su, J. Managing knowledge sharing in distributed innovation from the perspective of developers: Empirical study of open source software projects in China. *Technol. Anal. Strateg. Manag.* **2017**, *29*, 1–22. [CrossRef]
43. Au, Y.A.; Carpenter, D.; Chen, X.; Clark, J.G. Virtual organizational learning in open source software development projects. *Inf. Manag.* **2009**, *46*, 9–15. [CrossRef]
44. Tyre, M.J.; Von Hippel, E. The situated nature of adaptive learning in organizations. *Organ. Sci.* **1997**, *8*, 71–83. [CrossRef]
45. Hardi, J. Situated Learning among Open Source Software Developers: The Case of Google Chrome Project. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 24 September 2010. Available online: http://urn.kb.se/resolve?urn=urn:nbn:se:bth-5266 (accessed on 11 June 2018).
46. Lave, J. Situating learning in communities of practice. *Perspect. Soc. Shared Cognit.* **1991**, *2*, 63–82.
47. Kolb, D. *Experiential Learning as the Science of Learning and Development*; Prentice Hall: Englewood Cliffs, NJ, USA, 1984.
48. Wen, S.-F. Learning Secure Programming in Open Source Software Communities: A Socio-Technical View. In Proceedings of the 6th International Conference on Information and Education Technology, Osaka, Japan, 6–8 January 2018.
49. Kowalski, S. IT insecurity: A Multi-Discipline Inquiry. Ph.D. Thesis, Department of Computer and System Sciences, University of Stockholm and Royal Institute of Technology, Stockholm, Sweden, 1994.
50. Al Sabbagh, B.; Kowalski, S. Developing Social Metrics for Security Modeling the Security Culture of it Workers Individuals (Case Study). In Proceedings of the 2012 Mosharaka International Conference on Communications, Computers and Applications (MIC-CCA), Istanbul, Turkey, 12–14 October 2012.
51. Da Veiga, A.; Eloff, J.H. A framework and assessment instrument for information security culture. *Comput. Secur.* **2010**, *29*, 196–207. [CrossRef]
52. Ramachandran, S.; Rao, S.V.; Goles, T. Information security cultures of four professions. In Proceedings of the 41st Annual Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 7–10 January 2008.
53. Martins, A.; Elofe, J. Information security culture. In *Security in the Information Society*; Springer: Boston, MA, USA, 2002; pp. 203–214.
54. Ngo, L.; Zhou, W.; Warren, M. Understanding Transition towards Information Security Culture Change. In Proceedings of the 3rd Australian Information Security Management Conference, Perth, Australia, 30 September 2005; pp. 67–73.
55. Duhon, H.J.; Elias, J. Why It's Difficult To Learn Lessons: Insights from Decision Theory and Cognitive Science. In Proceedings of the SPE Annual Technical Conference and Exhibition, Anaheim, CA, USA, 11–14 November 2007.
56. Eskerod, P.; Skriver, H.J.G. Organizational culture restraining in-house knowledge transfer between project managers—A case study. *Project Manag. Inst.* **2007**, *38*, 110–122. [CrossRef]
57. David, W.; Fahey, L. Diagnosing cultural barriers to knowledge management. *Acad. Manag. Exec.* **2000**, *14*, 113–127.
58. Ford, D.P.; Chan, Y.E. Knowledge sharing in a multi-cultural setting: A case study. *Knowl. Manag. Res. Pract.* **2003**, *1*, 11–27. [CrossRef]
59. Okhuysen, G.A. Structuring change: Familiarity and formal interventions in problem-solving groups. *Acad. Manag. J.* **2001**, *44*, 794–808.
60. Reich, B.H.; Gemino, A.; Sauer, C. Modeling the knowledge perspective of IT projects. *Project Manag. J.* **2008**, *39*, S4–S14. [CrossRef]
61. Faraj, S.; Sproull, L. Coordinating expertise in software development teams. *Manag. Sci.* **2000**, *46*, 1554–1568. [CrossRef]
62. Jiang, J.J.; Klein, G.; Chen, H.-G. The effects of user partnering and user non-support on project performance. *J. Assoc. Inf. Syst.* **2006**, *7*, 6. [CrossRef]
63. Tiwana, A.; McLean, E. Knowledge Integration and Individual Expertise Development in E-Business Project Teams: Prom the Pod to the Peas. In Proceedings of the 2002 ACM SIGCPR Conference on Computer Personnel Research, Kristiansand, Norway, 14–16 May 2002.
64. Kraut, R.E.; Streeter, L.A. Coordination in software development. *Commun. ACM* **1995**, *38*, 69–82. [CrossRef]
65. Cummings, J.N.; Espinosa, J.A.; Pickering, C.K. Crossing spatial and temporal boundaries in globally distributed projects: A relational model of coordination delay. *Inf. Syst. Res.* **2009**, *20*, 420–439. [CrossRef]

66. Espinosa, J.A.; Slaughter, S.A.; Kraut, R.E.; Herbsleb, J.D. Team knowledge and coordination in geographically distributed software development. *J. Manag. Inf. Syst.* **2007**, *24*, 135–169. [CrossRef]
67. Herbsleb, J.D. Global Software Engineering: The Future of Socio-Technical Coordination. In Proceedings of the 2007 Future of Software Engineering, Minneapolis, MN, USA, 23–25 May 2007.
68. Anttila, J.; Savola, R.; Kajava, J.; Lindfors, J.; Röning, J. Fulfilling the Needs for Information Security Awareness and Learning in Information Society. In Proceedings of the 6th Annual Security Conference, Las Vegas, NV, USA, 11–12 April 2007.
69. Child, J. *Organization: A Guide to Problems and Practice*; Sage: Newcastle upon Tyne, UK, 1984.
70. Nelson, M.; Sen, R.; Subramaniam, C. Understanding open source software: A research classification framework. *Commun. Assoc. Inf. Syst.* **2006**, *17*, 12.
71. Howard, M. Building more secure software with improved development processes. *IEEE Secur. Priv.* **2004**, *2*, 63–65. [CrossRef]
72. Kayworth, T.; Whitten, D. Effective information security requires a balance of social and technology factors. *MIS Q. Execut.* **2012**, *9*. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2058035 (accessed on 28 September 2018).
73. Tilton, J. What is an Infostructure. 1994. Available online: http://www.willamette.edu (accessed on 11 June 2018).
74. Pan, S.L.; Scarbrough, H. Knowledge management in practice: An exploratory case study. *Technol. Anal. Strateg. Manag.* **1999**, *11*, 359–374. [CrossRef]
75. Raymond, E. The cathedral and the bazaar. *Knowl. Technol. Policy* **1999**, *12*, 23–49. [CrossRef]
76. Crowston, K.; Wei, K.; Howison, J.; Wiggins, A. Free/Libre open-source software development: What we know and what we do not know. *ACM Comput. Surv.* **2012**, *44*, 7. [CrossRef]
77. Pohl, J. Intelligent software systems in historical context. In *Decision Support Systems in Agent-Based Intelligent Environments*; IOS Press: Amsterdam, The Netherlands, 2005.
78. Teece, D.J.; Pisano, G.; Shuen, A. Dynamic capabilities and strategic management. *Strateg. Manag. J.* **1997**, *18*, 509–533. [CrossRef]
79. Mckeen, J.D.; Zack, M.H.; Singh, S. Knowledge Management and Organizational Performance: An Exploratory Survey. In Proceedings of the 39th Annual Hawaii International Conference on System Sciences, (HICSS'06), Washington, DC, USA, 4–7 January 2006.
80. Rosenberg, M.J. *Beyond e-Learning: Approaches and Technologies to Enhance Organizational Knowledge, Learning, and Performance*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
81. Boh, W.F. Mechanisms for sharing knowledge in project-based organizations. *Inf. Organ.* **2007**, *17*, 27–58. [CrossRef]
82. Prencipe, A.; Tell, F. Inter-project learning: processes and outcomes of knowledge codification in project-based firms. *Res. Policy* **2001**, *30*, 1373–1394. [CrossRef]
83. Argote, L. *Organizational Learning: Creating, Retaining and Transferring Knowledge*; Springer Science & Business Media: Berlin, Germany, 2012.
84. O'Reilly, C.A. Variations in decision makers' use of information sources: The impact of quality and accessibility of information. *Acad. Manag. J.* **1982**, *25*, 756–771.
85. Numally, J.C. *Psychometric Theory*; McGraw-Hill: New York, NY, USA, 1978.
86. Malcolmson, J. What is Security Culture? Does it Differ in Content from General Organisational Culture? In Proceedings of the 43rd Annual 2009 International Carnahan Conference on Security Technology, Zurich, Switzerland, 5–8 October 2009.
87. Caldwell, B.S. Knowledge sharing and expertise coordination of event response in organizations. *Appl. Ergon.* **2008**, *39*, 427–438. [CrossRef] [PubMed]
88. Ajith Kumar, J.; Ganesh, L. Research on knowledge transfer in organizations: A morphology. *J. Knowl. Manag.* **2009**, *13*, 161–174. [CrossRef]
89. Lee, L.L. Knowledge sharing metrics for large organizations. In *Knowledge Management: Classic and Contemporary Works*; Morey, M.D., Thuraisingham, B., Eds.; MIT Press: Cambridge, MA, 2000; pp. 403–419.
90. Garrison, D.R. Self-directed learning: Toward a comprehensive model. *Adult Educ. Q.* **1997**, *48*, 18–33. [CrossRef]
91. Song, L.; Hill, J.R. A conceptual model for understanding self-directed learning in online environments. *J. Interact. Online Learn.* **2007**, *6*, 27–42.

92. Mittendorff, K.; Geijsel, F.; Hoeve, A.; de Laat, M.; Nieuwenhuis, L. Communities of practice as stimulating forces for collective learning. *J. Workplace Learn.* **2006**, *18*, 298–312. [CrossRef]
93. GitHub. Celebrating Nine Years of GitHub with an Anniversary Sale, Web Page. Available online: https://github.com/blog/2345-celebrating-nine-years-of-github-with-an-anniversary-sale (accessed on 11 June 2018).
94. Gousios, G.; Vasilescu, B.; Serebrenik, A.; Zaidman, A. Lean GHTorrent: GitHub Data on Demand. In Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad, India, 31 May–1 June 2014.
95. Straub, D.W. Validating instruments in MIS research. *MIS Q.* **1989**, *13*, 147–169. [CrossRef]
96. Agarwal, R.; Prasad, J. Are individual differences germane to the acceptance of new information technologies? *Decis. Sci.* **1999**, *30*, 361–391. [CrossRef]
97. Hair, J.F.; Black, W.C.; Babin, B.J.; Anderson, R.E.; Tatham, R.L. *Multivariate Data Analysis*; Prentice Hall: Upper Saddle River, NJ, USA, 1998.
98. Saraph, J.V.; Benson, P.G.; Schroeder, R.G. An instrument for measuring the critical factors of quality management. *Decis. Sci.* **1989**, *20*, 810–829. [CrossRef]
99. Badri, M.A.; Davis, D.; Davis, D. A study of measuring the critical factors of quality management. *Int. J. Qual. Reliab. Manag.* **1995**, *12*, 36–53. [CrossRef]
100. Scacchi, W. Understanding the requirements for developing open source software systems. *IEE Proc. Softw.* **2002**, *149*, 24–39. [CrossRef]
101. Safa, N.S.; Von Solms, R. An information security knowledge sharing model in organizations. *Comput. Hum. Behav.* **2016**, *57*, 442–451. [CrossRef]