A Cross-role Analysis on Security Efforts and Constraints of Software Development Projects

Fumihiro Kanei, Ayako Akiyama Hasegawa, Eitaro Shioji, Mitsuaki Akiyama

NTT, Tokyo, Japan

Abstract

In this study, we quantitatively analyzed security efforts and constraints of software development projects through an online survey of of software development professionals from the US (N=307). We revealed how certain characteristics of a development project, such as the project's contractual relationships, influence security efforts and constraints. In addition, by comparing the survey results of two groups (developers and managers), we revealed how the gap in their security efforts and constraints influences software security. We believe the results provide insights toward designing usable measures to assist security-related decision-making in software development and conducting an appropriate survey targeting software development professionals.

1 Introduction

In the last five years, researchers have focused on the human aspects of software development to understand the causes of software vulnerabilities and have presented technical and organizational approaches to address these causes [2–4, 6, 10].

Although researchers have intensively researched software developers, they have not sufficiently investigated how the characteristics of software products and the form of software development affect software security. Several studies have examined the causes of conflicts between security and other explicit requirements and discussed the lack of resources (e.g., time, personnel, and budget) for security practices [4, 16]. However, with a focus on multi-person projects for software development, no studies to date have quantitatively investigated how other vital characteristics of the project-based software development (i.e., contract types, development methods, and users of the software) affect developers' attitudes towards software security.

Recent security research on software development has focused on various types of developers (e.g., computer science students, freelancers, and company developers) in order to improve the ecological validity of results [9], but the roles of people involved in project development have not been sufficiently researched. Specifically, there has been insufficient quantitative analysis in terms of the specific security behavior and awareness of managers and how the gaps between software development professionals[1] (i.e., managers and developers) affect the secure software developments.

Against this background, we attempt to answer the following research questions (RQs) in this work.

- **RQ1.** How do software development characteristics (e.g., the project's contractual relationship) impact developers' security behavior and awareness?
- **RQ2.** Are there any gaps between developers and managers regarding security behavior and awareness? If so, how do they impact the security of products?

To answer these RQs, we conducted an online survey of 307 professionals in the US who were engaged in software development projects. Our questionnaire is composed of questions about (1) characteristics of development projects and products and (2) participants' security behavior and awareness. We conducted analyses on the basis of the correlations among the answers. Also, we surveyed two different groups (developers and managers) and analyzed both groups' tendencies and differences.

This paper makes the following contributions:

- We identified that the absence of decision-making authority and difficulty in security-related decision-making are strong obstacles that prevent secure software development.
- We identified that the perceptions of developers and managers regarding security of development project tend to differ. This finding suggests that, when designing surveys targeting software development professionals, one must first consider the characteristics of developers and managers and select appropriate participants who suit the purpose or the content of a survey.

[1] In this work, we use the terms "developers" and "managers" when explicitly distinguishing between the two roles, and "software development professionals" when otherwise.

2 Related work

Most relevant to our study, Assal and Chiasson conducted an online survey to explore the interplay between developers and software security processes [4], specifically focusing on (1) strategies developers use to deal with security, (2) developers' motivations and deterrents towards software security, and (3) the influence of the development methodology, company size, or adoption of Test-Driven Development (TDD) for software security. Our study differs from their work in the following aspects: the newly focused factors in software development, such as user scope of developed software, contractual relationships, as well as development methodology and company size; and a comparative analysis of developers vs. managers who have decision-making authority to introduce security efforts for secure development.

3 Methodology

Survey design: Our survey questions are categorized into the three parts: (1) participant demographics, (2) development characteristics, and (3) security behavior and awareness. In designing the questionnaire, we made iterations of reviews to minimize the cognitive load of participants. Specifically, we first designed a prototype of our questionnaire for (1) participant demography and (3) security behavior and awareness on the basis of the questionnaires presented in previous studies [4, 14]. Also, on the basis of our interviews with five experts with rich software development experience, we designed questions for (2) development characteristics. After that, the prototype questionnaire was thoroughly reviewed by the authors as well as by four development experts, and questions that were semantically similar to other questions or difficult to understand were removed or revised. Finally, we conducted a pilot survey with two development management experts to check whether the questionnaire was sufficient and was an appropriate length. Note that, as same as in previous study [4], we told participants that all questions were optional and the survey would be conducted anonymously.

Questions on development characteristics: We asked questions about three development-related characteristics that may impact software security: (1) whether the software being developed by participants was for use by the general public or limited to specific users; (2) what the contractual relationship of the participants' project was: in-house development (the product is developed for the participant's company) or contracted development (the product is developed for another company); and (3) which of the following development methods was adopted in the participants' project: Waterfall, Agile, or a hybrid of the two (e.g., Spiral). A similar survey was conducted by Assal and Chiasson [4], but they did not qualitatively study user scope, or contractual relationship, which are new perspectives added in our study.

Questions on security behavior and awareness: The questionnaire asked 32 questions (R1, E1–E15, A1–A15, and C1–11) about the survey participants' security behavior and awareness in their projects. Question R1 asked about the percentage of resources directed towards security out of the overall resources in a project. Questions E1–E15, A1–A5, and C1–C11 asked about security efforts practiced, security awareness, and factors hindering security, respectively. These questions asked participants to rate, on a 5-point Likert scale (Strongly agree to Strongly disagree), how much they agreed with a statement. For participants who were not aware of what security efforts were in place, E1–E15 includes the option of "Not sure". Also, to cover the cases not listed, we added a question with a free-format answer.

Recruitment: Our survey focuses on software development in teams, so we target managers in addition to developers who work on software development. We conducted the survey with participants recruited through a paid service offered by a survey company [8], which has a diverse participant pool, in August 2020. Participants were first asked following screening questions and then filtered and grouped on the basis of the results: (1) whether participants were working on software development in a team of multiple people, and (2) whether their role on the team was developer (with development tasks such as implementation, testing, and reviewing) or manager (with management tasks such as scheduling and resource management). All participants who completed the survey were paid US$10 worth of monetary reward. This amount is well above the federal minimum wage calculated on the basis of the average survey completion time.

Data quality: To ensure sufficient data quality, we excluded low-quality responses on the basis of the following filter rules: responses that failed to pass a simple attention check, responses with contradictions, responses that finished in less than 5 minutes, and responses that included a meaningless answer in an open-ended question, i.e., answers which seem to be filled mechanically without reading questions.

Ethics: This study follows the research ethics principles stated in the Menlo Report, and the survey questions and procedures were approved by our Institutional Review Board. Participants were informed in advance about the content of the survey and participated at their own will. Collected personal data was handled in compliance with the personal information protection laws of the participant's country.

4 Results

Our survey covered a total of 307 participants (162 developers and 149 managers). The average survey completion time with valid answers was 22.4 mins (Md=10.9).

Factor analysis We performed an exploratory factor analysis (EFA) on the results of the security-related questions to reduce the number of variables used in the analysis by grouping the results of each question. EFA was conducted using

principal axis factoring and promax rotation. The questions with factor loadings of 0.4 or higher for the common factors were grouped together, and the others were excluded from the analysis as recommended by Fabrigar et al. [5]. EFA was performed for questions about security efforts (E1-E15) and security constraints (C1-C11). The questions about security awareness (A1-A5) were excluded from the EFA because the Kaiser-Meyer-Olkin (KMO) measure [7] was less than 0.5 (KMO=0.49), and the results were inappropriate for EFA.

We grouped 12 of the 15 questions on security efforts into three factors. *Plan/Design* describes the status of security efforts during upstream processes such as planning and designing in development process. *Implementation* describes the status of security efforts during implementation phase in development process. *Vulnerability assessment* describes the status of security efforts for vulnerability assessments. Three questions did not conform to any factor.

The 11 questions about security constraints were grouped into four factors. *Lack of resources* describes how a lack of resources hinders security in development projects. *Unconcerned about security* describes security constraints caused by unconcern about security in development projects. *No authority/Conservative* describes the difficulty of changing the current development process and how lack of decision-making authority interferes with security. *Difficulty of introducing security* describes the difficulty of introducing new security measures into the development project.

When performing statistical tests, we used variables for each factor by averaging the answer of questions belonging to the same factor. Note that the average scores were calculated using numerical scores, $+2$ (strongly agree) to -2 (strongly disagree), assigned to each option of a Likert scale.

RQ1. Development characteristics and security: We investigated how characteristics of development, i.e., project and developed product, impact the security of developed software. Specifically, we divided the answers into groups in accordance with the results of answers to questions on development characteristics and compared each group to see if they had different tendencies regarding their answers to security-related questions. In this analysis, we used a Mann-Whitney U test to investigate whether there was a significant difference between two groups. The significance level used for the tests was 0.05. Due to the limitation of space, we describe the results of the comparison for contractual relationships, where the most important findings were made.

We divided answers into two groups on the basis of contractual relationships: in-house development (N=64, 21%) and contracted development (N=242, 79%). The results of a comparison between the two groups regarding security constraints are shown in Figure 1. We observed significant difference in *No authority/Conservative* (U=8843.5, p<.05, r=0.13). This suggests that software development professionals in projects located in the lower part of a contractual hierarchy are less likely to be able to make security efforts

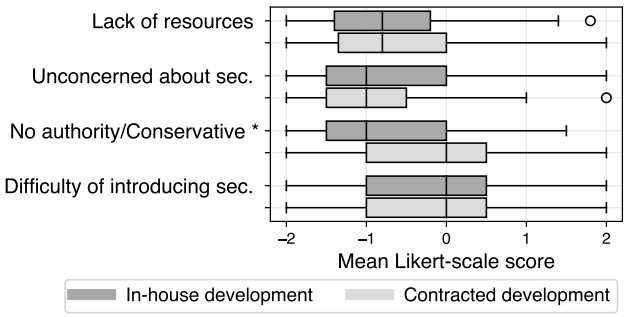

Figure 1: Comparison of security constraints between contractual relationships *w/sig diff (p<.05)

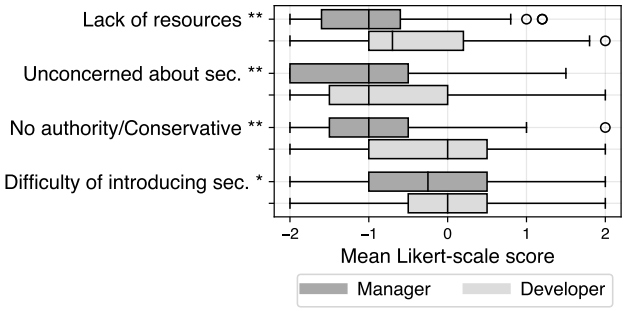

Figure 2: Comparison of security constraints between developers and managers *w/sig diff (p<.05), **w/sig diff (p<.01)

at their own discretion, possibly due to requests or priorities made from their contractor. This finding is supported by some free-format answers that describe security constraints experienced by participants who selected contracted development, as they stated that some constraints are caused by prioritization of non-security-related requests given by their clients (product owners). An example was "*Sometimes our client doesn't demand for security features.*"

RQ2. Difference between developers and managers: By comparing the results between developers and managers, we investigated whether there are any gaps between their security behavior and awareness. We used the same procedure as in the analysis of RQ1 to test for significant differences.

From the answers for security constraints, we confirmed that C8 (decision-making is difficult), C9 (no cost-effective measures), and C11 (cannot change dev. process) were strong security constraints for both developers and managers. Focusing on developers, constraints related to decision-making (C8, C10) were in the top three security constraints . Figure 2 shows the results of comparison between developers and managers regarding security constraints. Since significant differences were observed between developers and managers for all factors on security constraints, developers

tend to consider security-related constraints more than managers do. Also, we observed that there was a particularly large difference in scores for the *No authority/Conservative*. These results suggest that, even when developers feel the need for security, they do not have authority over managing development resources or deciding the development process or priorities. This suggests that that developers cannot make the security efforts they want to because of their position and that they strongly feel constrained about it.

We also analyzed the numbers of answers for which "Not sure" was selected for E1–E15 to see how well the managers and developers understood the security efforts in the project. Throughout E1–E15, developers tended to answer "Not sure" more than managers, and more than 5% of developers selected "Not sure" for in E7 (tools for secure coding) and E12 (outsourcing sec. assessment). This result suggests that developers are less likely than managers to understand the project's overall security efforts.

5 Implications

Supporting security-related decision-makings in software development. According to the results in Sections 4, the developers tend not to know the overall security measures of the project or have the decision-making authority to implement them. Therefore, when considering security technologies for software development, it is important to approach managers who have decision-making authority. Moreover, since the difficulty of decision-making tends to hinder secure software development, managers will need to be assisted in the decision-making process. We consider that the information needed to make a decision about security measures contains both "how well the security measure covers the threats" and "how much the measure costs to implement." If these details can be notified to managers accurately and effectively, they may be able to reduce the difficulties in security-related decision-making.

The results in Section 4 also indicate that managers tend to feel less constrained about security than developers. This discrepancy in participants within the development team may hinder the smooth implementation of security measures. If a manager does not correctly recognize the importance of security measures or the actual impediment of adopting them, proper security measures may not be taken. In fact, one developer wrote the following open-ended response regarding security awareness: "*Getting the client, managers and development team to be on the same page has always been a difficult task.*" To resolve this discrepancy in developers' and managers' awareness, interventions can be conducted to share the security issues that developers are concerned about with managers.

In supporting decision-making, it is necessary to consider not only the roles of people involved in development project but also the stakeholders. From the results in Section 4, we found that it is important to approach the organization that has the decision-making authority to implement the security measures (i.e., prime contractor organization), because subcontractors carry out the development projects in accordance with pre-determined deadlines, budgets, and functional requirements but tend not to have the discretion to implement the security measures.

Suggestion about design of user study for software development professionals. Two points found in this study should be considered in future research on software developers: (i) developers tend not to fully understand the implementation of security measures in their projects, and (ii) developers often do not have the authority to make decisions about implementing security measures. Much usable security research that has investigated people involved in software development has pointed out the ecological validity, which is the extent to which research findings generalize to real-world settings [1, 9, 15]. To improve the ecological validity, two factors have been discussed: participant demographics [1, 9] and experimental context [1, 15]. While computer science students, freelancers, and company developers have been investigated in previous research for the former factor, we shed new light on managers in this study. Although much research has investigated people involved in software development in the context of usable security, to the best of our knowledge, few studies [11–13] have distinguished between managers and developers as different participant demographics or analyzed differences in manager's and developer's awareness and behaviors. When designing surveys targeting software development professionals, one must first consider the characteristics of developers and managers and recruit appropriate participants who suit the purpose and content of a survey.

6 Conclusion and Future work

In this study, we conducted an online survey of people involved in software development projects. Our analysis of survey results revealed that characteristics of development or software development professionals' positions impact the security of software. Among those characteristics, the lack of security-related decision-making authority or the difficulty in making decisions strongly impacts software security; therefore, when considering security measures for software development professionals in the future, it is important to approach the people or organizations with decision-making authority. In addition, since there is a gap in security awareness between developers and managers, one must appropriately design a survey considering this fact when conducting a survey study targeting software development professionals.

References

[1] Y. Acar, S. Fahl, and M. L. Mazurek. You are not your developer, either: A research agenda for usable security and privacy research beyond end users. In *Proceedings of the 2016 IEEE Cybersecurity Development*, SecDev '16, pages 3–8. IEEE, 2016.

[2] Y. Acar, C. Stransky, D. Wermke, C. Weir, M. L. Mazurek, and S. Fahl. Developers Need Support, Too: A Survey of Security Advice for Software Developers. In *Proceedings of the 2017 IEEE Cybersecurity Development*, SecDev '17. IEEE, 2017.

[3] H. Assal and S. Chiasson. Security in the Software Development Lifecycle. In *Proceedings of the 14th Symposium on Usable Privacy and Security*, SOUPS '18. USENIX Association, 2018.

[4] H. Assal and S. Chiasson. "Think Secure from the Beginning": A Survey with Software Developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19. ACM, 2019.

[5] L. R. Fabrigar, D. T. Wegener, R. C. MacCallum, and E. J. Strahan. Evaluating the use of exploratory factor analysis in psychological research. *Psychological methods*, 4(3):272, 1999.

[6] P. L. Gorski, L. L. Iacono, D. Wermke, C. Stransky, S. Moeller, Y. Acar, and S. Fahl. Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic API Misuse. In *Proceedings of the 14th Symposium on Usable Privacy and Security*, SOUPS '18. USENIX Association, 2018.

[7] B. D. Hill. *Sequential Kaiser-meyer-olkin Procedure as an Alternative for Determining the Number of Factors in Common-factor Analysis: a Monte Carlo Simulation*. PhD thesis, Oklahoma State University, 2011.

[8] Macromill Group. https://group.macromill.com/, 2020.

[9] A. Naiakshina, A. Danilova, E. Gerlitz, and M. Smith. On Conducting Security Developer Studies with CS Students: Examining a Password-Storage Study with CS Students, Freelancers, and Company Developers. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13. ACM, 2020.

[10] D. S. Oliveira, T. Lin, M. S. Rahman, R. Akefirad, D. Ellis, E. Perez, R. Bobhate, L. A. DeLong, J. Cappos, Y. Brun, and N. C. Ebner. API Blindspots: Why Experienced Developers Write Vulnerable Code. In *Proceedings of the 14th Symposium on Usable Privacy and Security*, SOUPS '18. USENIX Association, 2018.

[11] H. Palombo, A. Z. Tabari, D. Lende, J. Ligatti, and X. Ou. An Ethnographic Understanding of Software (In)Security and a Co-Creation Model to Improve Secure Software Development. In *Proceedings of the 16th Symposium on Usable Privacy and Security*, SOUPS '20. USENIX Association, 2020.

[12] A. Poller, L. Kocksch, S. Türpe, F. A. Epp, and K. Kinder-Kurlanda. Can Security Become a Routine? A Study of Organizational Change in an Agile Software Development Group. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17. ACM, 2017.

[13] T. W. Thomas, M. Tabassum, B. Chu, and H. Lipford. Security During Application Development: An Application Security Expert Perspective. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 262. ACM, 2018.

[14] D. Votipka, D. Abrokwa, and M. L. Mazurek. Building and Validating a Scale for Secure Software Development Self-Efficacy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20. ACM, 2020.

[15] D. Votipka, K. R. Fulton, J. Parker, M. Hou, M. L. Mazurek, and H. Michael. Understanding security mistakes developers make: Qualitative analysis from Build It, Break It, Fix It. In *Proceedings of the 29th Conference on USENIX Security Symposium*, SEC '20. USENIX Association, 2020.

[16] J. Xie, H. R. Lipford, and B. Chu. Why do programmers make security errors? In *Proceedings of the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing*, VL/HCC '11. IEEE, 2011.