Asia Pacific Journal of Information Systems
Vol. 26 No. 2 (JUNE 2016), 231-246

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
http://dx.doi.org/10.14329/apjis.2016.26.2.231

Communal Antecedents in the Adoption of Secure Coding Methodologies

Sung Kun Kim[a,*], Ji Young Kim[b]

[a] *Professor, Department of Business Administration at Chung-Ang University, Korea*
[b] *Research Associate, Business School of Chung-Ang University, Korea*

ABSTRACT

Technology acceptance model has demonstrated that technology adoption behavior can be explained by two user belief constructs: perceived usefulness and perceived ease of use. A number of studies have explored how these beliefs develop by utilizing primarily individual-level antecedents. However, because innovation and new techniques bear a direct relation to social concerns, non-individual antecedents may be necessary. Therefore, in this study, social and organizational supports are used to understand how software developers foster beliefs regarding secure coding practices.

We compiled data from 83 software developers to evaluate the technology acceptance model. Our findings show that these collective antecedents can effectively explain user belief constructs and the intention to adopt secure coding methodologies. These findings imply that society and organizations offering more concrete support programs will experience smoother deployment of security-enhancing measures.

Keywords: Secure Coding, Technology Adoption, Social Support, Organizational Support, Communal Antecedents

Ⅰ. Introduction

Security must be a key feature of information systems. Especially, software systems must be able to withstand ever-present threats from cyber. For instance, when source code contains vulnerabilities such as buffer overflows or SQL injections, the systems are more likely to be exploited. Therefore, software developers must write code that eliminates such vulnerabilities.

Secure coding is a set of pre-deployment practices for eliminating software vulnerabilities (Jang and Choi, 2014; Seacord, 2006). These practices, however, are not easy (Graff and van Wyk, 2003; Jones and Rastogi, 2014; Whittaker, 2003). They are comprised of multifaceted objectives, which include software developers' security awareness (Taylor and Kaza, 2011) and a set of analysis techniques or tools to

*Corresponding Author. E-mail: skim@cau.ac.kr Tel: 8228205133 be utilized (Chess and West, 2007; Emanuelsson, and Nilsson, 2008). Specifically, these practices must be combined into an integrated and comprehensive system development methodology (Gregoire et al., 2007).

An adoption of a system development methodology by software developers represents a significant change in their practices (Pfleeger, 1999). Because such a change is often burdensome to developers, they tend to hesitate to adopt and use. As Riemenschneider et al. (2002) emphasized that "organizations attempting to deploy a methodology tend to face much resistance from individual developers".

As a result, only a few methodologies have been adopted by software developers, even though a wide selection of newly proposed methodologies have been available in the information systems (IS) community (Tan, 2006; Recker, 2010a). When deploying a new methodology, organizations may encounter an unnecessary waste of expenditure and effort (Roberts et al., 1998).

These problems have contributed to a call in the IS community for further research. Accordingly, a few researchers have addressed the adoption of methodologies or modeling methods. Their main research objective was to determine factors leading to the successful adoption of methodologies. Specifically, Hardgrave et al. (2003) and Riemenschneider et al. (2002) studied software developers' adoption of methodologies while the adoption of process modeling methods was investigated by Recker (2010a,b) and Tan (2006). The conceptual basis employed in their studies are strongly related to theories of motivated human behavior of tool adoption, including technology acceptance model (TAM) (Davis, 1989), with an assumption that "the tool acceptance domain would be applicable to the methodology acceptance context" (Riemenschneider et al., 2002).

Since discovering the utility of TAM, the IS community has been researching factors that influence belief constructs. The identified antecedents have been primarily related to individual characteristics such as experience (Lederer et al., 2000) and self-efficacy (Venkatesh and Davis, 2000; Riemenschneider et al., 2003), and to the nature of technologies such as quality (Lederer et al., 2000) and relative advantages (Venkatesh et al., 2003). This seems natural because these studies were primarily designed to address individual decisions about tool acceptance. In terms of methodologies or methods relating to measures in the community or society, collective antecedents should be additionally considered (Zhao et al., 2010). Society or enterprise-level endeavors can serve as sound examples. Nowadays, secure coding has become a society-level or communal endeavor for more secure IT environments, not a choice by any single software developer. Our study aims to identify communal antecedents for explaining TAM belief constructs with the intention of guiding software developers to accept secure coding methodologies.

Ⅱ. Background and Research Models

2.1. Models Explaining Technology Acceptance

Why users adopt and employ a certain technology is a major research question in the IS community. As expected, the community has proposed models to explain individual adoption and use of technology.

One of the earliest and most fundamental models is the theory of reasoned action (TRA) developed by Fishbein and Ajzen (1975). Drawn from the field of social psychology, the model asserts that a person's behavior can be determined by his or her behavioral intention to perform; such an intention can be explained by the person's attitude and subjective norm toward the behavior. Ajzen developed the theory of planned behavior (TPB) by adding the construct of perceived behavioral control, which is defined as "the perceived ease or difficulty of performing the behavior" (Ajzen, 1991).

The most renowned model is the TAM (Davis, 1989). Grounded in the TRA, the TAM takes a restrictive approach. That is, the adoption intention can be explained by two specific belief constructs: perceived usefulness (PU) and perceived ease of use (PEOU). Furthermore PEOU is expected to have a positive association with PU.

Because of its simplicity and strong theoretical foundation, the TAM has been applied to a wide range of technologies and users. Subsequently, extensions or revisions to the TAM have been made in subsequent studies. TAM2 (Venkatesh and Davis, 2000) and the unified theory of acceptance and use of technology (UTAUT) (Venkatesh et al., 2003) are good examples. To compare these models, King and He (2006) determined through a meta-analysis using 88 published TAM studies that the original TAM is a powerful and robust predictive model.

An initial adoption of technology does not guarantee its continuous use. After a technology or an information system has been accepted and made available to users for performing their work activities, they may discontinue using it or apply it in a way that deviates from the original usage mode. In this regard, Bhattacherjee (2001), by differentiating initial and post-adoption behaviors, developed the IT continuous use (ICU) model. According to this model, the intent to continue using a technology is determined by both perceived performance and post-adoption satisfaction, which are influenced by initial pre-usage expectations.

2.2. Intention to Adopt Systems Development Methodologies

A software development method, which is composed of a set of formalized processes, representation constructs, and guidelines/techniques, is designed to improve software development practices. In the IS community, the acceptance and adherence to such methodologies by software developers is an essential innovation. Because the adoption of a method represents a greater change in work practice, developers are inclined to resist adoption and use of the new method. That is the reason why, despite an abundance of methodologies proposed for the IS community, only a few methodologies have been widely accepted (Recker, 2010a; Tan, 2006).

Research into this issue has been undertaken by a few scholars. Their major goal has been to determine why and how software developers accept and continue to use software development methodologies. Among pioneering studies, the work of Hardgrave et al. (2003) combined the TAM and the DOI to produce a research model in which an intention of methodology adoption can be predicted using belief-related constructs. The empirical results showed that perceived usefulness, social pressure, compatibility, and organizational mandates influence methodology adoption intention, while complexity does not.

Tan (2006) and Recker (2010b) studied the adoption of process modeling methods. In Tan's study, the TPB was used as the primary theoretical foundation with antecedent factors, such as modeling method characteristics and institutional factors. The result showed that the TPB effectively explains software developers' intentions to continue using a modeling method. Furthermore, two institutional factors, training and social influence, were found to have a significant impact on attitude, subjective norm, and perceived behavioral control (Tan, 2006). Moreover, Recker (2010b) used a combined model of TAM and ICU to explain developers' continuous use of process modeling methods. As their antecedent constructs, individual difference factors were employed, including modeling experience, modeler background, and grammar familiarity. The empirical results indicated that the combined model is valid for explaining developers' intentions to continue using a process modeling method and that the individual difference factors influence user belief constructs.

In addition, secure system development has been studied. Woon and Kankanhalli (2007) investigated factors determining the adoption of the secure system development practice. Their study did not address a particular method; it covered a whole practice of secure system development, including requirements analysis, design, coding, testing and maintenance. Currently the most pressing concern in the security domain is secure coding, which is clearly distinct from the efforts in other stages of secure software development (Chess and West, 2007; Jones and Rastogi, 2004). Kumar et al. (2007) emphasized that, no matter how well requirements and design phases are performed, mistakes in coding can still occur because "code is developed by humans and humans are imperfect." Moreover, writing secure code is quite difficult (Whittaker, 2003; Taylor and Kaza, 2011). Many organizations are facing difficulties in successfully deploying secure coding methodologies.

Ⅲ. Research Model and Hypotheses

3.1. Research Model

Secure coding is the practice of writing source code that cannot be exploited for performing illegal operations by cyber attackers. An application that is not coded in a secure way may certainly include several vulnerabilities that can be exploited by attackers. There is a fundamental difference between approaches taken by developers and those used by adversaries. While developers implement functional requirements so that the application performs intended tasks, attackers are "more interested in how an application can be manipulated to perform tasks other than intended" (OWASP, 2010).

Despite the importance of secure coding, almost all secure coding studies have focused on technical issues. A major aim of these studies has been to identify exploitable code and to instruct on technically correcting it (Seacord, 2006; Halfond and Orso, 2005). For instance, Klein (2011) described how he discovered vulnerabilities in actual software systems ('bug hunting' in his term) and suggested how these vulnerabilities could be remedied. However, we believe that behavioral issues are equally important, including how and why developers perceive, accept, or continue to use secure coding methodologies. Whittaker (2003) emphasized that we can "hope to write secure applications" only if we come to recognize the presence of exploitable vulnerabilities. This implies that whether or not secure coding practices are accepted largely depends on developer behavior.

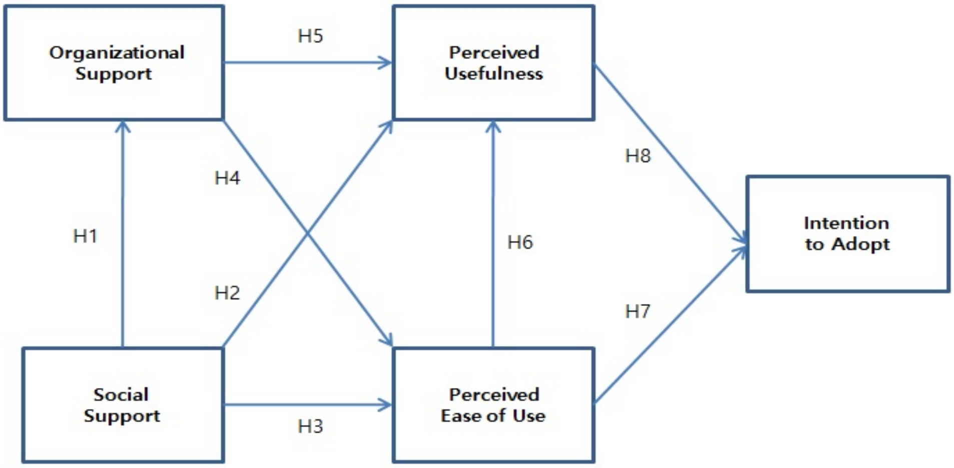
As to whether the initial adoption model or continuous usage model is applied, we have decided that the former is more appropriate because secure coding has only recently come to prominence in Korea. Furthermore, the mere simplistic application of a secure coding method does not guarantee that there is no security flaws in the developed software. Graff and van Wyk (2003) stressed that psychological factors heavily work against secure software in practice. Therefore, the TAM is the fundamental theoretical framework for the problem we aim to address, and we have selected two belief constructs, PU and PEOU, as predictors of method adoption intention.

Antecedents employed by most TAM-related studies have been individual-level constructs and technology-characteristic constructs. Experience (Lederer et al., 2000), self-efficacy (Venkatesh and Davis, 2000; Riemenschneider et al., 2003) or the nature of technologies (quality (Lederer et al., 2000), and relative advantages (Venkatesh et al., 2003) are good examples.

However, both individual and collective dimensions are important to understand human behavior (Ali, 1988; Rogers, 1995). Two types of collective effort or support exist: society and organization. Leavy (1983) and Viswesvaran et al. (1998) described the role of social support in the process of work stress. Further, antecedents of organizational support have been used in many areas, including the adoption of technology (Karahanna and Straub, 1999; Venkatesh and Davis, 2000).

Today, software systems are essential components of the social infrastructure. For many individuals to function in daily life, software systems, such as internet banking, e-government, and e-logistics, should be stable and secure. For optimal security assurance, society must play an active role. For instance, society is required to provide a variety of support, including continuous monitoring of security incidents and making regular public announcements of most urgent software vulnerabilities. The Open Web Application Security Project (OWASP) and the System Administration, Networking, and Security (SANS) Institute are examples. We define social support as the extent to which professional or public entities provide support for society members to accomplish certain objectives.

When organizations deploy innovations that are uncomfortable to members, they should take steps to offer support for the implementation. In other words, for organizational innovation to be successfully deployed, organizational support is required. In a study on the adoption of collaborative commerce by organizations, Zhao (2010) identified organizational support as antecedents that influence belief constructs. To implement secure coding practices,



<Figure 1> Research Model

organizations should provide secure coding tools, such as static analysis (Chest and West, 2007) and the expertise and support of secure coding specialists. Therefore, we argue that 'organizational support' should be selected as an antecedent. Organizational support is defined as the extent to which an organization provides support for its members to accomplish certain objectives.

3.2. Research Hypotheses

Based on the research model shown in <Figure 1>, a number of research hypotheses can be derived. First, organizations must follow common social values because they are part of a superordinate social system (Terreberry, 1968). Dowling and Pfeffer (1975) explained that in order for organizations to be legitimate, their practices should align with the goals of society. Secure coding is a method that societies can employ to make their social information technology (IT) infrastructures more secure and safe. Society can offer various support or programs, including awareness-raising seminars or training programs. These kinds of social efforts may induce organizations to perform some activities that foster secure coding. The purchasing of secure coding tools and making them available to system developers is one example. In this regard, the greater the social support (SS), the greater the organizational support (OS).

H1: The extent of social support will positively affect the extent of organizational support.

Social influence or pressure has been recognized as a key determinant in technology acceptance (Anandarajan et al., 2002; Taylor and Todd, 1995). Venkatesh and Davis (2000) indicated that there are three social forces that influence individual technology adoption behavior: subjective norm, voluntariness, and (projected self-) image. They also empirically showed the impact of these social forces on perceived usefulness.

Social support can be understood as a process that enables these social forces to naturally occur. A key component of social support is an awareness of social issues and countermeasures. One can see only as much as one knows. Regarding secure coding, professional communities often hold seminars and training programs to demonstrate cases of security breaches caused by software vulnerabilities, as well as techniques to eliminate these vulnerabilities in source code. Once the importance of secure coding and its implementation is recognized, the usefulness of secure coding practices will be perceived to a greater extent.

H2: Social support will positively affect perceived usefulness.

Compeau and Higgins (1991) indicated that support programs can increase user self-efficacy. That is, people who are educated and trained through social support programs may recognize that much of the discomfort and difficulties experienced under innovation can be eliminated. Through secure coding training courses, developers can learn why a particular coding practice is prone to be exploited and how to revise the problem code. Such training, we contend, will lead to the perception that secure coding is not necessarily incomprehensible to ordinary developers, and that they can manage it without great difficulty once proper coding guidelines and related tools are employed

H3: Social support will positively affect perceived ease of use.

In many studies organizational support has been used to explain technology adoption behavior as an antecedent (Mahmood et al., 2001; Zhao et al., 2010) and a moderator (Lee et al., 2005). Lewis et al. (2003) demonstrated the influence of organizational factors, such as management's commitment to new technology, on belief constructs. The impact of organizational support on both perceived ease of use and perceived usefulness was empirically confirmed in Anandarajan et al. (2002). Accordingly, to successfully deploy secure coding practice, organizations must provide various support programs such as retaining secure coding specialists and purchasing code analysis tools. Developers who have experienced these organizational support programs should experience increased PEOU and PU.

H4: Organizational support will positively affect perceived ease of use.

H5: Organizational support will positively affect perceived usefulness.

The remaining hypotheses were drawn from the original TAM (Davis, 1989). The two constructs, PU and PEOU, have been fundamental determinants of technology adoption behavior. At the same time, ease of use has been asserted to have an influence on usefulness.

H6: Perceived ease of use will positively affect perceived usefulness.

H7: Perceived ease of use will positively affect behavioral intention.

H8: Perceived usefulness will positively affect behavioral intention.

Ⅳ. Research Design

Below, we briefly describe the instrumentation, sampling method, and scale validation process.

4.1. Instrumentation

Our instrument was constructed by analyzing and adapting scales from previous studies and constructing new scales when necessary. Behavioral intention (INT) items were adapted from Hardgrave et al. (2003), using two items. PEOU items were adapted from Lucas and Spitler (1999) and applied to secure coding. The PU measure was newly constructed based on Davis's (1989) scale, which consists of increases in 1) job performance, 2) productivity, 3) effectiveness, 4) efficiency, 5) task manageability, and 6) value on the job. Because the domain of our study is development methodologies, not a particular technology, some of the above six items are less relevant to our study while some are better combined. We therefore used a three-item scale: 1) increased level of security in software systems (similar to Davis's Item 3); 2) decreased development costs, which corresponds to direct outcomes of secure coding (similar to Davis's Items 2 and 4 combined); and 3) decreased maintenance costs, which corresponds to the outcome expected after the developed system is delivered (a new item). Both social support (SS) and organizational support (OS) were constructed using a two-item scale. Each item was measured using a seven-point Likert scale. Appendix A shows the measurement items.

4.2. Data Collection

To evaluate the above hypotheses, a field survey of developers who participated in a secure coding

<Table 1> Summary of Respondent Demographics

Job Title	Respondents (%)	Experience	Respondents (%)
Programmer	56 (70.9)	Less than 3 years	23 (29.1)
System Designer	11 (13.9)	3 - 5 years	13 (16.5)
Tester	8 (10.1)	5 - 7 years	12 (15.2)
Project Manager	4 (5.0)	7 - 10 years	15 (19.0)
		More than 10 years	16 (20.2)
Total	**79 (100.0)**	**Total**	**79 (100.0)**

seminar held by a security-related public agency was conducted. The aim of the seminar was to improve secure coding-related capabilities of developers employed by software development contractors whose main business is the development of government-operating systems. This seminar was free and participation was entirely voluntary. A questionnaire and seminar-related materials were provided to participants at the registration desk. During the initial seminar session, we presented a brief introduction about the survey. Participants were asked to complete the questionnaire during breaks and return it to the desk when leaving.

Out of 138 distributed forms, 83 responses were collected, yielding a response rate of 60.1%. With four incompletely answered questionnaires excluded, this study used 79 samples for the analysis.

As shown in <Table 1>, all survey respondents held a job directly related to system development. Among them, more than 80% were programmers and software testers whose jobs mainly involved program code. Regarding experience, survey respondents were diversely distributed.

Ⅴ. Data Analysis and Results

5.1. Reliability and Validity

For validity analysis, factor analysis with principal components analysis and varimax rotation was used. At the initial factor analysis run, the item (PU3) was determined to be a category itself. We decided to remove the maintenance-related item because the respondents were developers who were contracted to develop a system, not to perform maintenance work after deployment. <Table 2> shows the final run factor loading of values of the remaining items along with Cronbach's .α.

To assess item reliability, we used the square factor loading. The remaining ten items all met the criteria with SFL ≥ 0.50. All of their Cronbach's .α. values also met the criteria .α. ≥ 0.70, as shown in <Table 3>. Composite reliability, ρ, ranged from 0.80 to 0.95, and was found to meet the criteria ρ ≥ 0.70. To test convergent validity the average variance extracted (AVE) was used. The AVE values of all five constructs ranged from 0.66 to 0.91, satisfying the criteria AVE ≥ 0.50.

5.2. Structural Equation Modeling

Our research model was tested by structural equation modeling using AMOS with maximum-likelihood estimation. In structural equation modeling, model fit is indicated by various indices. As indicated in <Table 3>, all model fit indices met the recommended criteria.

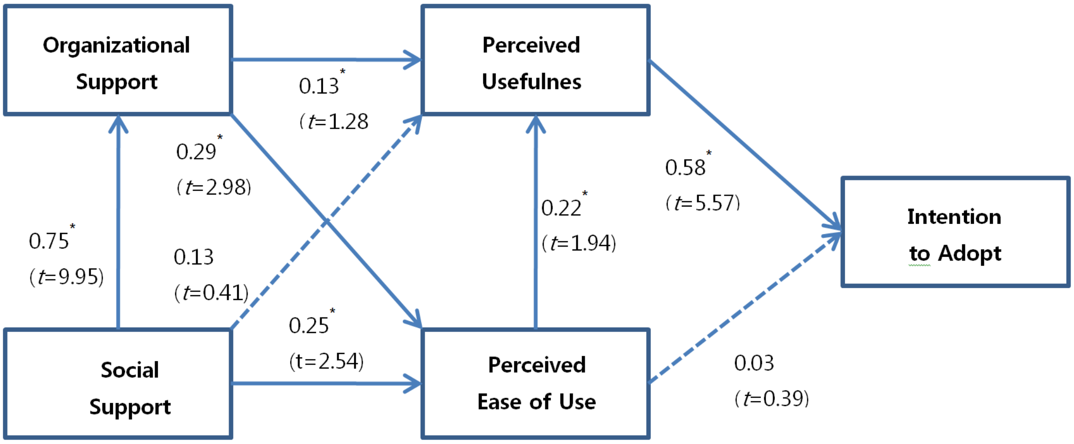
An analysis of causal relationships in the research model was then conducted. <Figure 2> shows results

<Table 2> Validity of Constructs

	Component					Cronbach's a
	1	2	3	4	5	
INT1	.930	.117	.103	.234	.114	0.954
INT2	.925	.070	.080	.290	.054	
OS1	.032	.843	.262	.198	.239	0.809
OS2	.188	.776	.275	.102	.302	
PEOU1	.082	.246	.859	.207	.176	0.843
PEOU2	.118	.282	.832	.119	.244	
PU1	.231	.163	.153	.841	.204	0.798
PU2	.352	.120	.167	.821	.010	
SS1	.109	.304	.242	.139	.887	0.915
SS2	.097	.548	.324	.117	.692	

<Table 3> Reported Values of Model Fit

Model Fit Index	Recommended Value	Reported Value
X^2	$p \geq 0.05$	$P = 0.826$
X^2/df	≤ 3.00	0.191
GFI	≥ 0.90	0.998
AGFI	≥ 0.80	0.985
NFI	≥ 0.90	0.998
CFI	≥ 0.90	1.0
RMR	≤ 0.09	0.023
RMSEA	≤ 0.10	0.000



<Figure 2> Results of Structural Model Analysis

with the estimated path coefficients and associated *t* values. Only PU was found to significantly affect INT, while EOU had an indirect impact on INT through its significant direct effect on PU. This finding confirms the results of some previous studies that determined that ease of use had no direct effect on the intention to adopt technology (Igbaria and Iivari, 1995; Igbaria et al., 1995).

The SS, as hypothesized in our research model, significantly influenced the OS. This result confirmed the organizational legitimacy view that organizations aiming to be legitimate will behave congruently with social forces (Dowling and Pfeffer 1975; Terreberry, 1968). The study result also showed that SS had a significant effect on PEOU. This is consistent with the social support role view that social support can mitigate stress at work (Ganster et al., 1986; Viswesvaran et al., 1998), which implies that an increase in perceived ease of use through social support programs may lead to an alleviation of work-related strains.

The effect of SS on PU, however, was not significant. One possible explanation is that the respondents probably viewed secure coding as a complex matter. In fact, secure coding consists of two major steps, detection of vulnerable code and correction of the code in question. The first step can be managed without major difficulty once code analysis tools are provided; however, the second step is an entirely different matter. Just like the diagnosing of a disease cannot guarantee the curing of the disease, the correction of vulnerable code may belong to a different sphere and require a set of more advanced technical capabilities. The respondents may have felt that existing social support programs are not sufficient enough to enhance the capability of developers so that they can handily correct the identified vulnerable code.

The study results showed that OS significantly affected both PU and PEOU. Unlike previous studies (Karahanna and Straub, 1999; Lewis et al., 2003) that did not find an influence of organizational support programs on these two belief constructs, this study identified a direct effect of OS on PEOU and PU. We presume such a difference arises from the technology in question. While they focused on an E-mail system, which is relatively simple and easy to use, secure coding in this study was a more complex problem.

All things considered, the research model explained 32.4% of the variance in INT, 42.2% in PEOU, 18.7% in PU, and 55.4% in OS.

Ⅵ. Discussion

6.1. Theoretical Implication

Perceptions and beliefs about innovation are not identical across individuals (Lewis et al., 2003). Therefore, how individuals form beliefs about their use of innovative technology and, more specifically, what antecedents influence user beliefs about technology use have become an important research topic. Because the theoretical frameworks of most of these studies are individual adoption behavior models such as the TAM and TPB, it is quite natural for these studies to use individual-level antecedents.

However, in terms of innovation or new techniques likewise bearing also a direct relation to social concerns, it is believed that collective antecedents need to be introduced. This study used social support and organizational support in the adoption of secure coding methodologies. It can be noted that this research enriches adoption literature of existing methodologies in that communal factors were considered.

6.2. Practical Implication

Our study determined that PU is a major determinant of adoption intention. It repeatedly demonstrated the previous finding that usefulness is more important than PEOU, even though ease of use has an indirect effect on the intention to adopt technology.

The study findings have key implications on policy makers. SS was found to have an impact on ease of use, implying that current guidelines and relevant training programs about secure coding that are offered by professional and public entities will help software developers feel better prepared to follow secure coding practices. The observed influence of social support on organizational support also indicates that publicly available support programs will lead to an increase in management awareness about security-enhancing measures and will promote security-related investments within their firm.

OS was found to significantly determine both usefulness and ease of use, which implies that management should seek specific support programs when deploying security-enhancing measures. One may assert that, the greater the number of concrete support programs that an organization offers, the more smoothly the security-enhancing measures will be deployed.

6.3. Study Limitation

One major limitation of this study is the sample size of 79, which is much lower than the generally required level of 150 to 200. A caution in applying the TAM is that one must assure that respondents know about the target innovation. Therefore, we had to perform a field study. Accordingly, we decided to use a public seminar event as the field site. While "sample size requirements remain a vexing question in SEM-based studies" (Westland, 2010), the smaller sample size might be an inevitable consequence of our decision. In fact, we could not have accurately known in advance the total number of attendees for the event.

Regarding managerial or social support, a variety of support can be employed. In this study, only support programs, such as direct investment, guidelines set-up, and training programs, are covered. An integrated view combining various types of other support programs could provide a deeper understanding about the determinants of developers' beliefs about secure coding methodologies.

Ⅶ. Conclusion

In this paper, we proposed an extended model based on the TAM to investigate the impact of social and organizational support on software developers' intention to adopt secure coding methodologies. Considering the findings that social and organizational support can foster developers' beliefs about method adoption, all interested parties such as policy makers, professional groups, and top managers of a firm should understand that more support programs that assist software developers' adoption of secure coding are a worthy investment. In future studies, we may evaluate the hypothesized model with a larger sample size. In addition, support can be extended to include other antecedents, such as managers' attitudes and social awareness.

<References>

[1] Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50*(2), 179-211.

[2] Ali, A. (1988). A cross-national perspective of managerial work value systems. *Advances in International Comparative Management*, *3*(15), 151-165.

[3] Anandarajan, M., Igbaria, M. and Anakwe, U. (2002). IT acceptance in a less-developed country: a motivational factor perspective. *International Journal of Information Management*, *22*(1), 47-65.

[4] Bhattacherjee, A. (2001). Understanding information systems continuance: An expectation-confirmation model. *MIS Quarterly*, *25*(3), 351-370.

[5] Chess, B. and West, J. (2007). *Security Programming with Static Analysis*. Addison-Wesley.

[6] Compeau, D. and Higgins, C. (1991). A social cognitive theory perspective on individual reactions to computing technology. *In Proceedings of the Twelfth International Conference on Information Systems*, 187-198.

[7] Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, *13*(3), 319-340.

[8] Dowling, J. and Pfeffer, J. (1975). Organizational legitimacy: Social values and organizational behavior. *Pacific Sociological Review*, *18*(1), 122-136.

[9] Emanuelsson, P. and Nilsson, U. (2008). A comparative study of industrial static analysis tools. *Electronic Notes in Theoretical Computer Science*, *217*, 5-21.

[10] Fishbein, M. and Ajzen, I. (1975). *Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research*. Addison-Wesley.

[11] Ganster, D., Fusilier, M. and Mayes, B. (1986). Role of social support in the experience of stress at work. *Journal of Applied Psychology*, *71*(1), 102-110.

[12] Graff, M. and van Wyk, K. (2003). *Secure Coding: Principles & Practices*. O'Reilly.

[13] Gregoire, J., Buyens, K., De Win, B., Scandariato, R. and Joosen, W. (2007). On the secure software development process: CLASP and SDL compared, *Third International Workshop on Software Engineering for Secure Systems*.

[14] Halfond, W. and Orso, A. (2005). Combining static analysis and runtime monitoring to counter SQL-injection attacks. *ACM SIGSOFT Software Engineering Notes*, *30*(4), 1-7.

[15] Hardgrave, B., Davis, F. and Riemenschneider, C. (2003). Investigating determinants of software developers' intentions to follow methodologies. *Journal of Management Information Systems*, *2*(1), 123-152.

[16] Igbaria, M. and Iivari, J. (1995). The effects of self-efficacy on computer usage. *Omega*, *23*(6), 587-605.

[17] Igbaria, M., Guimaraes, T. and Davis, G. (1995). Testing the determinants of microcomputer usage via a structural equation model. *Journal of Management Information Systems*, *11*(4), 87-114.

[18] Jang, Y. S., and Choi, J. Y. (2014). Detecting SQL injection attacks using query result size. *Computers & Security*, *44*, 104-118.

[19] Jones, R. and Rastogi, A. (2004). Secure coding: Building security into the software development cycle. *Information Systems Security*, *13*(5), 29-39.

[20] Karahanna, E. and Straub, D. (1999). The psychological origins of perceived usefulness and ease-of-use. *Information & Management*, *35*(4), 237-250.

[21] King, W. and He, J. (2006). A meta-analysis of the technology acceptance model. *Information & Management*, *43*(6), 740-755.

[22] Klein, T. (2011). *A Bug Hunter's Diary*. No Starch Press.

[23] Kumar, R., Pandey, S. and Ahson, S. (2007). Security in Coding Phase of SDLC. *Third International Conference on Wireless Communications and Sensor Networks*, Dec. 13-15, 118-120.

[24] Leavy, R. (1983). Social support and psychological

disorder: A review. *Journal of Community Psychology*, *11*(1), 3-21.

[25] Lederer, A., Manpin, D., Sena, M. and Zhuang, Y. (2005). The technology acceptance model and the World Wide Web. *Decision Support Systems*, *29*(3), 269-282.

[26] Lee, H., Lee, Y. and Kwon, D. (2005). The intention to use computerized reservation systems: the moderating effects of organizational support and supplier incentive. *Journal of Business Research*, *58*, 1552-1561.

[27] Lewis, W., Agarwal, R. and Sambamurthy, V. (2003). Sources of influence on beliefs about information technology use: An empirical study of knowledge workers. *MIS Quarterly*, *27*(4), 657-678.

[28] Lucas, H. and Spitler, V. (1999). Technology use and performance: A field study of broker workstations. *Decision Sciences*, *30*(2), 291-311.

[29] Mahmood, M., Burn, J., Gemoets, L. and Jacquez, C. (2000). Variables affecting information technology end-user satisfaction: a meta-analysis of the empirical literature. *International Journal of Human-Computer Studies*, *52*(4), 751-771.

[30] OWASP. (2010). *OWASP Secure Coding Practices Quick Reference Guide*.

[31] Pfleeger, S. (1999). Understanding and Improving Technology Transfer in Software Engineering. *Journal of Systems and Software*, *47*(2-3), 111-124.

[32] Recker, J. (2010a). Explaining usage of process modeling grammars: Comparing three theoretical models in the study of two grammars. *Information* and *Management*, *47*(5), 316-324.

[33] Recker, J. (2010b). Continued use of process grammars: The impact of individual difference factors. *European Journal of Information Systems*, *19*(1), 76-92.

[34] Riemenschneider, C., Hardgrave, B. and Davis, F. (2002). Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE Transactions on Software Engineering*, *28*(12), 1135-1144.

[35] Riemenschneider, C., Harrison, D. and Mykytyn, P. (2003). Understanding IT adoption decisions in small business: Integrating current theories. *Information & Management*, *40*(4), 269-285.

[36] Roberts, T., Gibson, M., Fields, K. and Rainer, R. (1988). Factors that impact implementing a system development methodology. *IEEE Transaction on Software Engineering*, *24*(8), 640-649.

[37] Rogers, E. (1995). *Diffusion of innovations*, Free Press.

[38] Seacord, R. (2006). Secure coding in C and C++ of strings and integers. *IEEE Security & Privacy*, *4*(1), 74-76.

[39] Tan, X. (2006). *Understanding Information Systems Developers' Modeling Method Continuance: A Theoretical Model and an Empirical Test*, Ph.D. Thesis, University of Nebraska.

[40] Taylor, B. and Kaza, S. (2011). Security Injections: Modules to help students remember, understand, and apply secure coding techniques. *ITiCSE*, 27-29.

[41] Taylor, S., and Todd, P. (1968). Understanding information technology usage: a test of competing models. *Information Systems Research*, *6*(2), 144-176.

[42] Terreberry, S. (1968). The evolution of organizational environments. *Administrative Science Quarterly*, *12*(4), 590-613.

[43] Venkatesh, V. and Davis, F. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, *46*(2), 186-204.

[44] Venkatesh, V., Morris, M., Davis, G. and Davis, F. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, *27*(3), 425-478.

[45] Viswesvaran, C., Sanchez, J. and Fisher, J. (1999). The role of social support in the process of work stress: A meta-analysis. *Journal of Vocational Behavior*, *54*(2), 314-334.

[46] Westland, J. (2010). Lower bounds on sample size in structural equation modeling. *Journal of Electronic Commerce Research and Applications*, *9*(6), 476-487.

[47] Whittaker, J. (2003). Why secure applications are difficult to write. *IEEE Security & Privacy*, *1*(2), 81-83.

[48] Woon, I. and Kankanhalli, A. (2007). Investigation

of IS professional s' intention to practice secure development of applications. *International Journal of Human-Computer Studies*, *65*(1), 29-41.

[49] Zhao, H., Yang, J. and Wang, X. (2010). *Research on antecedent of adoption intentions of collaborative commerce based on TOE and TAM*. 2010 International Conference on E-Business and E-Government, 332-335.

<Appendix A> Constructs and Measurement Items

Perceived Usefulness	PU1 Following secure coding practices would improve the robustness of the software that I develop. PU2 Following secure coding practices would reduce the total development costs of software that I develop. PU3 Following secure coding practices would reduce the maintenance costs of software that I develop.
Perceived Ease of Use	PEOU1 I would easily acquire knowledge needed for secure coding. PEOU2 I would easily apply secure coding-related knowledge to my work practices.
Social Support	SS1 I am regularly provided with a list of software vulnerabilities, which are identified from actual security incidents and released by professional communities. SS2 I would not have a problem finding and taking secure coding-related training programs provided by public agencies.
Organizational Support	OS1 I would seek assistance from secure coding specialists within my organization if needed. OS2 I would be able to use secure coding analysis tools within my organization if needed.
Intention to Adopt	INT1 I intend to follow secure coding practices. INT2 I intend to comply with secure coding standards and guidelines.

◆ About the Authors ◆

Sung Kun Kim

Sung Kun Kim is a Professor in the Department of Business Administration at Chung-Ang University, Seoul, KOREA. He holds a Ph.D. in information systems from the Stern Business School of New York University. His current research interests focus on the managerial issues related to the development of IT systems and the adoption of new technologies. His work has been published in the Expert Systems with Applications, Asia Pacific Journal of Information Systems, Information Systems Review, and Journal of IT Applications & Management.

Ji-Young Kim

Ji-Young Kim is a Research Associate at the Business School of Chung-Ang University. He holds an M.S. degree in information systems from Chung-Ang University. His current research interests focus on information security management. His work has been published in Journal of IT Applications & Management.

Submitted: April 25, 2016; 1st Revision: June 1, 2016; Accepted: June 2, 2016