Investigation of IS Professionals' Intention to Practise Secure Development of Applications

Irene M.Y. Woon

Dept. of Information Systems
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543,
Republic of Singapore

Atreyi Kankanhalli

Dept. of Information Systems
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543,
Republic of Singapore

Forthcoming:

IJHCS Special Issue on

Information Security in the Knowledge Economy

Contact author: Dr. Irene Woon, Phone: (65) 6516-6296; Email: iwoon@comp.nus.edu.sg

Abstract

It is well known that software errors may lead to information security vulnerabilities the breach of which can have considerable negative impacts for organizations. Studies have found that a large percentage of security defects in e-business applications are due to design-related flaws, which could be detected and corrected during applications development. Traditional methods of managing software application vulnerabilities have often been adhoc and inadequate. A recent approach that promises to be more effective is to incorporate security requirements as part of the application development cycle. However, there is limited practice of secure development of applications and lack of research investigating the phenomenon.

Motivated by such concerns, the goal of this research is to investigate the factors that may influence the intention of information systems (IS) professionals to practise secure development of applications (SDA) i.e., incorporate security as part of the application development lifecycle. This study develops two models based on the widely used theory of planned behavior (TPB) and theory of reasoned action (TRA) to explain the phenomenon. Following model operationalization, a field survey of 184 IS professionals was conducted to empirically compare the explanatory power of the TPB based model versus the TRA based model.

Consistent with TPB and TRA predictions, attitude and subjective norm were found to significantly impact intention to practise SDA for the overall survey sample. Attitude was in turn determined by product usefulness and career usefulness of SDA, while subjective norm was determined by interpersonal influence, but not by external influence. Contrary to TPB predictions, perceived behavioral controls, conceptualized in terms of self-efficacy and facilitating conditions, had no significant effect on intention to practise SDA. Thus, a modified TRA based model was found to offer the best explanation of behavioral intention to practise SDA. Implications for research and information security practice are suggested.

Keywords: Secure development of applications, theory of planned behavior, theory of reasoned action, information security.

1. INTRODUCTION

IS applications are programs designed for end users to perform specific tasks ranging from simple word processing and calculations to more sophisticated tasks like electronic payment and supply chain management. With increased organizational reliance on IS applications, they present attractive targets to IS abusers who may create considerable damage and obtain benefits from attacking these systems. In fact it has been noted that in the recent year, hacking attacks tend to be more directed at the web applications rather than at the network or physical layer (Desmond 2004; Schindler 2004).

Although the impact of an application's security breach is significant and would lead one to believe that more effort would be made to avoid them, this is not true in practice. An increasing number of attacks exploit well-known vulnerabilities such as buffer overflow in applications. Recent examples include the Blaster worm, SQL Slammer and the Code Red worm. Buffer overflow attacks were first discovered in 1960 and have been widely known since the Morris worm in 1988 (Schneier 2000). Yet there is still a common recurrence of buffer overflow vulnerabilities in applications. For example, in 2003, 75% of the CERT advisories were related to buffer overflow vulnerabilities (CERT 2003). Overall, Microsoft attributes 50% of its software security problems to design flaws (Davis et al. 2004).

There are several reasons why software applications may be insecure. First, most designers and developers are not trained in general security principles. Thus, they typically do not incorporate security as an explicit requirement for applications (Mead and Stehney 2005; Viega et. al 2001). The issue of security is only addressed as and when vulnerabilities are discovered. Second, market forces dictate that ever more complex software products are delivered at accelerated speeds (Viega et. al 2001). In today's business environment, time-to-market has become crucial to the firm. To achieve this, it is conceivable that shortcuts may be taken e.g., less thorough code reviews and testing, resulting in less robust software that contains security flaws and bugs.

Traditionally, penetration testing is the most common mechanism used to assess software security (Arkin et al 2005). It involves evaluating IS security post-development by simulating attacks on the system. Once vulnerabilities are detected, technical solutions or ways to mitigate

the weaknesses are suggested. While this approach has its uses, there are several limitations as well. Vulnerabilities that are not discovered at the time the product is released or patched on time once the vulnerability is detected, give hackers the window of opportunity they need. Also, extensive penetration testing is prohibitively time and resource intensive. The cost of eliminating a software bug increases enormously the later in the software development cycle it is discovered i.e., from 5 times more if the flaw is caught at coding/unit testing stage to 15 times more if it is detected at beta testing (NIST 2002). Such reactive methods to manage application vulnerabilities have often been adhoc and inadequate (Landwehr et. al 1994; Viega and McGraw 2002; Verton 2002).

A more coherent approach would be to consider security needs right from the start i.e., the requirements gathering phase, and continue right through all the other phases of the development cycle such as design, implementation, testing, and maintenance (Davis et. al 2004; Flechais et. al 2003; Jones and Rastogi 2004; Siponen et. al 2005). The activities required include specifying security needs during requirements gathering, security threat modelling during design, following security guidelines during implementation, security testing during verification, security review and signoff before release, and security response feedback afterwards. Although such practice of secure development of applications (SDA) has been shown to provide security benefits (Lipner 2004), there is limited adoption of SDA and a lack of studies exploring the phenomenon.

With this motivation, our study investigates the intention of software professionals towards integrating software security practices into the software development life cycle. The antecedents of intention considered in this study are derived from the theory of planned behavior (Ajzen 1991), the theory of reasoned action (Fishbein and Ajzen 1975), and related studies in the IT domain (e.g., Taylor and Todd 1995). These two theories have successfully explained and predicted intention and behaviors in a wide range of domains (Albarracin et. al 2001**,** Bamberg et al 2003, Cardono and Frieze 2000**)** including information technology usage (Bhattacherjee 2000, Karahanna et. al 1999, Taylor and Todd 1995). Relevant to our context, they have served as theoretical basis for several studies on the acceptance of design and development methodologies (Johnson et al. 1999; Riemenschneider et al 2002; Hardgrave and Johnson 2003) as well as tools and languages (Lending and Chervany 1998; Agarwal and Prasad 2000) by software developers

and IS professionals. SDA can be considered as an evolving set of practices and guidelines that could be employed and incorporated into the existing software development methodology of the organization. Thus, we expect that the two theories would also be useful to predict the acceptance of SDA. Since there is limited adoption of SDA but a growing number of software professionals are becoming aware of it, our focus is to investigate IS professional's intention to practise SDA rather than its actual practice.

In our study, we use these theories to develop two models for IS professionals' intention of adopting the practise of SDA. The models based on the two theories were validated through a survey of IS professionals and compared for their explanatory power. Finally a third model derived from the results of testing the two models was found to provide the best explanation of the intention to practise SDA. The findings of such a study aim to provide organizations intending to adopt this development paradigm a better idea of how to motivate and encourage their software professionals to do so.

In Section 2, we provide a review of the two theories used in this study. This is followed by a description of the models based on the theories, including constructs and hypotheses. In Section 3, we describe the research methodology including the development of the survey instrument and data collection procedure. Subsequently, the results of data analysis to test the models are presented in Section 4 and the interpretation of the results discussed in Section 5. Finally, we conclude by highlighting the contributions of this study.

2. THEORETICAL BACKGROUND AND RESEARCH MODELS

2.1 Theory of Reasoned Action (TRA)

TRA (Fishbein and Ajzen 1975) postulates that human behavior is determined solely by the individual's intention to perform the behavior and behavioural intention is in turn determined by individual's attitude towards the behavior and subjective norms (see Figure 1). Attitude refers to the degree to which a person has a favourable or unfavourable evaluation of the behavior in question. Subjective norm refers to the perceived social pressure to perform or not perform the behavior. Attitude is in turn determined by behavioural beliefs about the consequences of performing the behavior and subjective norm depends on the normative beliefs about the

expectations of specific referents.

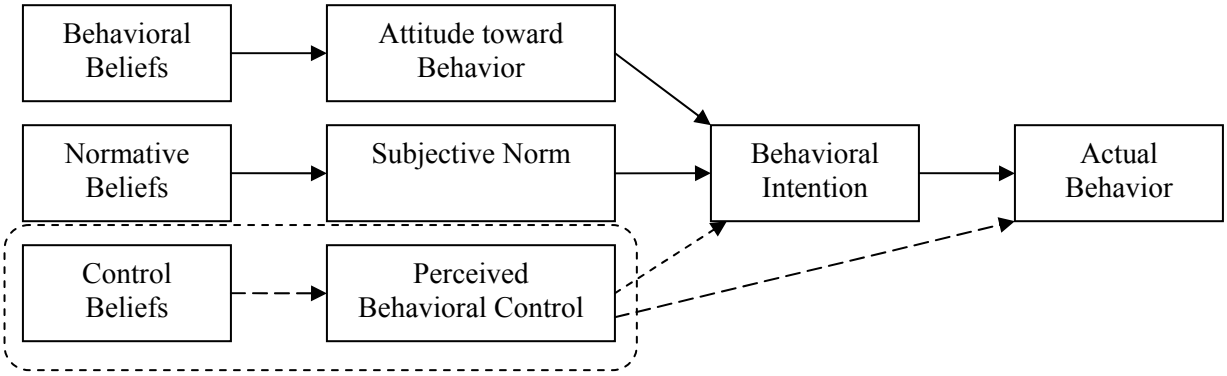


Figure 1: TRA and TPB

Note: Constructs enclosed in the rounded rectangle and links given in dotted lines are only part of the TPB model.

2.2 Theory of Planned Behavior (TPB)

TPB (Ajzen 1991) is an extension of TRA, which takes into account perceived behavioural controls. Perceived behavioral control refers to the perceived ease or difficulty of performing the behavior. In TPB, behavior is a direct function of behavioral intention as well as perceived behavioral control (see Figure 1). Behavioral intention is in turn determined by attitude towards the behavior, subjective norm, and perceived behavioral control. Perceived behavioral control depends on control beliefs about the presence of factors that may facilitate or impede performance of the behavior (Ajzen 1991).

A decomposed version of the TPB (Taylor and Todd 1995) has been popular since it provides a fuller understanding of behavioral intention by focusing on the specific behavioral, normative, and control beliefs that are likely to influence behavior. Based on the decomposed TPB (Taylor and Todd 1995), behavioral beliefs are largely influenced by perceived usefulness; normative beliefs can be decomposed into different referents' influence; and control beliefs depend on self-efficacy and facilitating conditions.

Since both TRA and TPB have been successfully used to explain adoption of particular software development tools and methodologies (e.g., Hardgrave and Johnson 2003; Agarwal and Prasad 2000), we apply them to understand the antecedents of intention to adopt SDA.

2.3 Research Models

This study applies TRA and TPB to develop models to explain the intention of IS professionals to practise secure development of applications (SDA) and tests which model offers a better explanation of intention. Here, the dependent variable was behavioral intention to practise SDA since actual practice has hardly taken place. Also intention has been shown as a significant predictor of actual usage behavior in previous studies (Davis et al 1989; Legris et al 2003).

In both the TRA and TPB based models, behavioral intention is proposed to be influenced by attitude and subjective norms. Further, attitude depends on behavioral beliefs while subjective norm is determined by normative beliefs. The TPB based model additionally includes control beliefs as antecedents of behavioral intention. The relevant behavioral beliefs (product and career usefulness), normative beliefs (external and internal influence), and control beliefs (self efficacy and facilitating conditions) for SDA practise adoption are identified and explained in the next few subsections along with the corresponding hypotheses. The TRA and TPB based models are shown together in Figure 2.

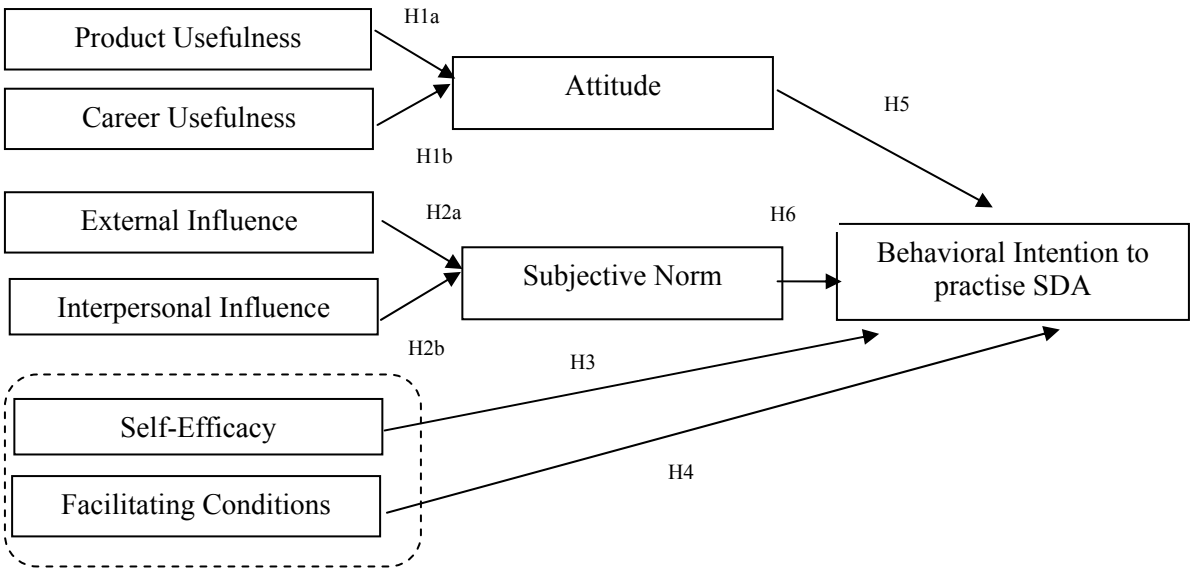


Figure 2: Research Models

Note: Constructs and links enclosed in rounded rectangle are excluded in the test of the TRA model

2.3.1 Behavioral beliefs

According to the decomposed TPB (Taylor and Todd 1995), behavioral beliefs determining attitude towards a technology are largely composed of the perceived usefulness of the technology. Previous studies on developers' beliefs about systems development methods (e.g. Johnson et al. 1999) have suggested that the perceived usefulness construct can be divided into a personal component (the benefits of the behavior to one personally) and a task-related component (the benefits of the behavior to one's situation or task). Personal usefulness depended on career usefulness while task-related usefulness was determined by product usefulness (Johnson et al. 1999). Therefore we consider product usefulness and career usefulness as the behavioral beliefs that influence attitude towards SDA.

In the context of our study, product usefulness refers to the usefulness of practising SDA on improving the product i.e., the application. Product usefulness is likely to influence attitude as the most commonly cited advantages of practising SDA are that it would result in a more secure application and that the resultant product is less costly than retrofitting security after the application is deployed (Schwartz 2003). Career usefulness refers to the extent to which developers would find the knowledge or the practise of SDA useful to their career development. Career usefulness has been determined as a significant factor influencing developers' acceptance of CASE tools (Chau 1996). It is likely that product usefulness and career usefulness would improve developers' attitude towards practice of SDA. Therefore we hypothesize,

H1a: Product usefulness is positively related to attitude towards practice of SDA

H1b: Career usefulness is positively related to attitude towards practice of SDA

2.3.2 Normative beliefs

Based on the TRA (Fishbein and Ajzen 1975) and TPB (Ajzen 1991), normative beliefs determining subjective norms are composed of the influences of important referents. Previous studies in the context of software development methods (e.g., Johnson et al 1999) have found that the important referents for subjective norms can be both external and internal to the organization. External influence refers to extra-organizational pressure exerted by those outside one's organization. In the context of our study, external influence includes the reports by security organizations and mass media that promote the practice of SDA (Schwartz 2003; Viega and

McGraw 2002). These are likely to exert social pressure on individuals to practice SDA.

Interpersonal influence refers to intra-organizational pressure exerted by those within one's organization. Peers and superiors compose two groups within the organization that are likely to influence the individual (Taylor and Todd 1995). Therefore we expect such interpersonal influence to exert social pressure on IS professionals to practise SDA. Hence we hypothesize,

H2a: External influence is positively related to subjective norms towards practice of SDA

H2b: Interpersonal influence is positively related to subjective norms towards practice of SDA

2.3.3 Control Beliefs

According to the decomposed TPB (Taylor and Todd 1995), control beliefs that determine perceived behavioral controls and thereby influence behavioral intention are composed of self-efficacy and facilitating conditions. In several previous studies (e.g., Riemenschneider et al 2002), perceived behavioral controls has been replaced by its constituent beliefs since the perceived behavioral controls construct can be difficult to conceptualize on its own and may be confused with its constituents (Ajzen 2002). We follow the same approach in this study i.e., self-efficacy and facilitating conditions are directly linked to intention to practise SDA.

Self-efficacy refers to the belief or confidence in one's ability to perform behavior (Bandura 1977). If developers have high self-efficacy i.e., they believe that they are able to carry out SDA by themselves, it is more likely that they have a higher intention of carrying out SDA. Facilitating conditions (e.g., organizational support and availability of resources) can have a positive influence on behavioral intention (Venkatesh 2000). Since most developers have not been taught security in schools (Mead and Stehney 2005; McCown 2002; Viega et. al 2001), those who have these skills would have picked these up only after they have left school - on their own initiative or through training programs provided by their company. Furthermore, many companies are under competitive pressure to turn out applications with new features in a short time, giving developers little time to consider security during the short development time (Viega 2001; Viega and McGraw 2002). If facilitating conditions are available i.e., developers are given the training, tools, resources, and time, it is more likely that they would practise SDA. Therefore

we hypothesize,

H3: Self-efficacy is positively related to behavioral intention to practise SDA

H4: Facilitating conditions is positively related to behavioral intention to practise SDA

The remaining hypotheses in our model are common to the TPB (Ajzen 1991) and TRA (Fishbein and Ajzen 1975). As proposed by these theories and validated in subsequent studies based on the theories, we hypothesize

H5: Attitude is positively related to behavioral intention to practise SDA

H6: Subjective norms is positively related to behavioral intention to practise SDA

We empirically tested both the TRA and TPB based models to identify the salient antecedents of intention to practise SDA as well as to determine which model had better explanatory power.

3. RESEARCH METHODOLODY

The models were tested using survey methodology to aim for greater generalizability of results (Dooley 2001).

3.1 Scale Development

First, a draft instrument was constructed by adapting scales from previous literature to measure the models' constructs. The instrument was pre-tested with three industry experts in computer security to ensure its content validity. An interview was conducted with each expert and changes suggested by the expert were reflected in the instrument, which was then used for the next interview, according to the procedure outlined in Straub (1989). Items were added, reworded and deleted in this pre-test. To assess the construct validity of the various scales and to identify any ambiguous items, judges were asked to sort the items into construct categories according to the procedure in Moore and Benbasat (1991). A total of three sorting rounds were conducted till items were stable and a high item placement ratio (the percentage of items placed correctly in each construct) was achieved. The final survey instrument is shown in Table 1. All the items were measured using 7-point Likert scales ranging from “strongly disagree” (1) to “strongly agree” (7).

Construct	Items	Source
Product Usefulness (PU 1-3)	Practising secure development of applications (SDA) would make my applications more robust (better withstand attacks or misuse).	Iivari (1996), Green and Hevner (1999)
	Practising SDA would enable security requirements to be better captured.	
	Practising SDA would reduce the costs of application maintenance – retrofitting of security into apps. late would be harder and more costly.	
Career Usefulness (CU 1-5)	Knowledge of SDA would put me on the "cutting edge" in my field.	Chau (1996a), Johnson et al (1999), Compeau, Higgins and Huff (1999), Thompson et al (1991)
	Knowledge of SDA would enhance my status/ prestige among my peers.	
	Knowledge of SDA would improve my marketability.	
	Knowledge of/ Practising SDA would increase my job security.	
	Practising SDA would increase my value to the company.	
External Influence (EI 1-5)	Media reports suggest that practising SDA is a good idea.	Pedersen (2001), Bhattacherjee (2000), Johnson et al (1999), Expert source
	Experts consistently recommend practising SDA.	
	Consultants/ Trainers recommend practising SDA.	
	Government/ Professional bodies encourage the practice of SDA.	
	The market/customers demand that security is designed into the applications (i.e. it is considered early in the app. development lifecycle).	
Inter-personal Influence (II 1-5)	Almost all my peers (fellow web developers) practise SDA.	Pedersen (2001), Bhattacherjee (2000), Karahanna, Straub and Chervany (1999) Taylor & Todd (1995)
	Almost all my co-workers think that practising SDA is a good idea.	
	My peers/co-workers/friends think that we should all practise SDA.	
	My peers/co-workers/friends recommended that I should practise SDA.	
	My superior/manager would think that I should practise SDA.	
Self Efficacy (SE 1-3)	I would feel comfortable carrying out SDA on my own.	Bhattacherjee (2000), Taylor & Todd (1995), Pedersen (2001)
	I would be able to carry out SDA reasonably well on my own.	
	I would be able to carry out SDA without the help of others.	
Facil. Conditions (FC 1-5)	My organization provides me with training/guidance to carry out SDA.	Pedersen (2001), Bhattacherjee (2000), Thompson et al (1991), Venkatesh (2000), Mathieson (1991)
	My organization gives me adequate time to carry out SDA during applications development.	
	My organization gives me access to automated tools that help me in carrying out SDA.	
	My organization's current applications development practices (methodologies, techniques, tools, etc) facilitate the practice of SDA.	
	My organization gives me control over carrying out SDA.	
Attitude (A 1-4)	Practising SDA is a good idea.	Taylor and Todd (1995), Expert source
	*Practising SDA is unnecessary.	
	I like the idea of practising SDA.	
	*Practising SDA would be unpleasant.	
Subjective Norm (SN 1-3)	People who influence my behavior think that I should practise SDA.	Taylor and Todd (1995), Bhattacherjee (2000)
	People who are important to me think that I should practise SDA.	
	People whose opinions I value prefer that I practise SDA.	
Behavrial. Intention (BI 1-2)	I would practise SDA whenever possible.	Agarwal & Prasad (2000), Taylor & Todd (1995)
	I intend to practise SDA for critical applications, e.g. involving confidential or private information.	

Note: * indicates reverse coded items

Table 1. Questionnaire Items

3.2 Data Collection

As the software security area is relatively new with the first books and academic classes on the topic appearing only in 2001 (McGraw 2004a), not many IS professionals may be aware of it. Some may mistakenly assume that linking functional security features such as SSL into their software addresses security needs throughout the system (McGraw 2004b). To obtain a sample that is aware of SDA, we surveyed respondents from members of two large security interest groups for working professionals i.e., Special Interest Group in Security and Information inteGrity (SIG^2) and Information Systems Security Association (ISSA) Singapore Chapter, and from the banking industry where there is a greater awareness of security practices. Out of a total of 230 forms sent out, 184 responses were collected yielding a response rate of 80%. While participation was voluntary, the respondents were given a token gift for filling out the survey.

Descriptor	Percentage	Descriptor	Percentage
Position		**Industry**	
o Programmer	14.1	o Education	29.9
o Analyst	32.1	o Finance/ Insurance	13.0
o Designer	7.6	o Government/Defence/Public Service	11.4
o Project Manager	15.2		
o IT Manager	7.1	o Information Technology	30.4
o Other IT Professionals (e.g. systems specialists)	23.9	o Manufacturing	8.7
		o Others	6.5
Highest Qualification		**Experience in Applications Development**	
o Secondary	0.5		
o Pre-University	3.8	o Less than 1 year	6.5
o Diploma	8.7	o 1 – 3.5 years	31.0
o Bachelor's Degree	58.2	o 4 – 6.5 years	34.2
o Master's Degree	28.8	o 7 – 10 years	19.0
		o More than 10 years	9.2
Age		**Gender**	
o 20 – 29 years	38.0		
o 30 – 39 years	47.8	o Female	23.9
o 40 – 49 years	12.5	o Male	76.1
o 50 – 59 years	1.6		

Table 2: Summary of Demographic Information of Respondents

The majority of the survey respondents were 30-39 year old males holding bachelor's degrees and having 4-6.5 years of experience in applications development (see Table 2). The most common job title was analyst and the largest sector of respondents was from the IT industry.

4. DATA ANALYSIS AND RESULTS

The survey instrument was initially tested for reliability and validity following which, the path models were assessed using structural equation modelling. The best model was then tested for differences between subgroups of respondents i.e., analyst / programmers versus designer / managers.

4.1 Reliability and Validity

The models' constructs were assessed for reliability using Cronbach's alpha (Cronbach 1951). All the constructs had adequate reliability of at least 0.7 (Nunnally 1978) (see Table 3).

Construct	# Items	Cronbach's alpha
Product Usefulness (PU)	3	0.77
Career Usefulness (CU)	5	0.86
External Influence (EI)	5	0.82
Interpersonal Influence (II)	5	0.87
Self Efficacy (SE)	3	0.84
Facilitating Conditions (FC)	5	0.87
Attitude (A)	4	0.79
Subjective Norm (SN)	3	0.90
Behavioral Intention (BI)	2	0.76

Table 3: Reliability of Constructs

The items were tested for validity using factor analysis with principal components analysis and varimax rotation. Convergent validity was assessed by checking loadings to see if items for the same construct correlate highly amongst themselves. Discriminant validity was assessed by examining the factor loadings to see if items loaded more highly on their intended constructs than on other constructs (Cook and Campbell 1979). Loadings of 0.45-0.54 are considered fair, 0.55-0.62 good, 0.63-0.70 very good, and above 0.71 excellent (Comrey 1973). Factor analysis yielded eight components with eigenvalues above 1 (see Table 4).

	Component							
	1	2	3	4	5	6	7	8
PU1	0.41	-0.04	0.16	0.23	0.10	0.00	0.17	**0.69**
PU2	0.49	-0.06	0.00	0.13	0.17	0.08	0.11	**0.60**
PU3	**0.39**	-0.08	0.11	0.30	0.23	-0.12	0.23	0.33
CU1	**0.72**	0.02	0.02	0.03	0.28	0.12	0.16	0.28
CU2	**0.78**	-0.01	0.08	0.07	0.19	0.24	0.21	0.14
CU3	**0.81**	-0.02	0.03	0.20	0.16	0.14	-0.02	0.09
CU4	**0.78**	0.12	0.18	0.18	0.02	-0.06	0.04	-0.10
CU5	**0.63**	0.19	0.03	0.28	0.13	-0.01	0.00	0.26
EI1	0.26	-0.10	0.16	0.16	**0.68**	0.08	0.10	0.06
EI2	0.10	0.02	0.19	0.08	**0.81**	0.17	0.13	-0.04
EI3	0.20	0.24	0.28	0.16	**0.74**	0.01	0.06	0.13
EI4	0.15	0.28	0.18	0.12	**0.68**	0.08	-0.07	0.26
EI5	0.16	0.22	0.33	0.06	**0.45**	0.10	-0.19	0.07
II1	0.02	0.28	**0.71**	-0.14	0.21	0.08	0.12	-0.25
II2	0.03	0.01	**0.77**	0.10	0.27	0.18	0.07	0.07
II3	0.07	0.15	**0.81**	0.10	0.24	0.20	0.04	0.05
II4	0.18	0.29	**0.75**	0.01	0.05	0.17	0.06	0.22
II5	0.12	0.37	**0.61**	0.10	0.26	0.20	-0.02	0.20
SE1	0.21	0.10	0.03	0.39	0.04	0.20	**0.72**	0.11
SE2	0.07	0.21	0.06	0.19	0.00	0.07	**0.84**	0.08
SE3	0.09	0.25	0.08	-0.02	0.06	0.02	**0.81**	0.04
FC1	0.13	**0.80**	0.15	-0.05	0.02	0.11	0.16	-0.10
FC2	0.04	**0.84**	0.12	-0.08	0.09	0.15	0.10	-0.05
FC3	0.04	**0.77**	0.20	-0.18	0.07	0.10	0.17	-0.08
FC4	-0.09	**0.79**	0.23	0.18	0.13	0.15	0.03	-0.02
FC5	0.02	**0.56**	0.10	0.03	0.09	0.24	0.36	0.27
A1	0.26	-0.14	0.06	**0.69**	0.17	0.23	0.21	0.31
A2	0.09	-0.11	0.03	**0.75**	0.09	0.07	0.08	0.18
A3	0.33	-0.07	0.09	**0.70**	0.12	0.26	0.24	0.23
A4	0.16	0.12	-0.06	**0.70**	0.11	-0.16	-0.01	-0.18
SN1	0.16	0.20	0.24	0.08	0.09	**0.81**	0.10	0.05
SN2	0.09	0.29	0.31	0.09	0.17	**0.79**	0.10	0.01
SN3	0.08	0.32	0.23	0.17	0.15	**0.74**	0.08	0.16
BI1	0.21	0.09	0.14	**0.57**	0.09	0.23	0.23	0.41
BI2	0.09	-0.11	0.08	0.50	0.17	0.22	-0.01	**0.59**

Table 4: Validity of Constructs

All questions had at least fair loadings on their intended constructs except for PU3. Although PU3 did not load strongly onto any factor, it was retained as its deletion would result in a loss of content validity. BI1 and BI2 may have loaded together with attitude (A) and product usefulness (PU) respectively due to the strong relationships between BI and these constructs.

4.2 Structural Equation Modelling

The research models were tested by structural equation modelling (SEM) using AMOS with maximum likelihood estimation. AMOS is a covariance-based approach towards SEM. Covariance-based SEM is best suited for confirmatory research with a sound theory base (Gefen et al 2000), as in the case of this study based on the well-established TPB and TRA. In addition as the sample size was between the minimum range of 100-150 and the maximum of 200 and the constructs were all reflective, AMOS was also chosen over LISREL (which has larger sample size requirements) and PLS (which is suitable for exploratory research).

Model fit is indicated by multiple indices including the model chi-square (χ^2), chi-square to degrees of freedom ratio (χ^2/df), AGFI (Adjusted Goodness of Fit Index), CFI (Comparative Fit Index) and RMSEA (Root Mean Square Error of Approximation). As the chi-square test is extremely sensitive to sample size (Gefen et al 2000; Bhattacherjee 2000), the chi-square to degrees of freedom ratio (which is less sensitive) is used instead. Acceptable model fit is indicated by values of χ^2/df less than 3, AGFI greater than 0.80 (Gefen et al 2000), CFI greater than 0.90, and RMSEA less than 0.08 for reasonable fit and less than 0.06 for a good model fit (Hu and Bentler 1999).

Model testing and refinement was done in a progressive manner. First the larger TPB based model was tested followed by the TRA based model, which is a subset of the TPB-based model. The TRA-based model provided a better explanation of intention to practise SDA than the TPB-based model and also had better model fit indices. However since the TRA-based model did not have acceptable values for all model fit indices, further refinement was done to it. A modified TRA-based model was tested which provided the best explanatory power of all the three models and also had acceptable values for all fit indices. Each of these results is described below.

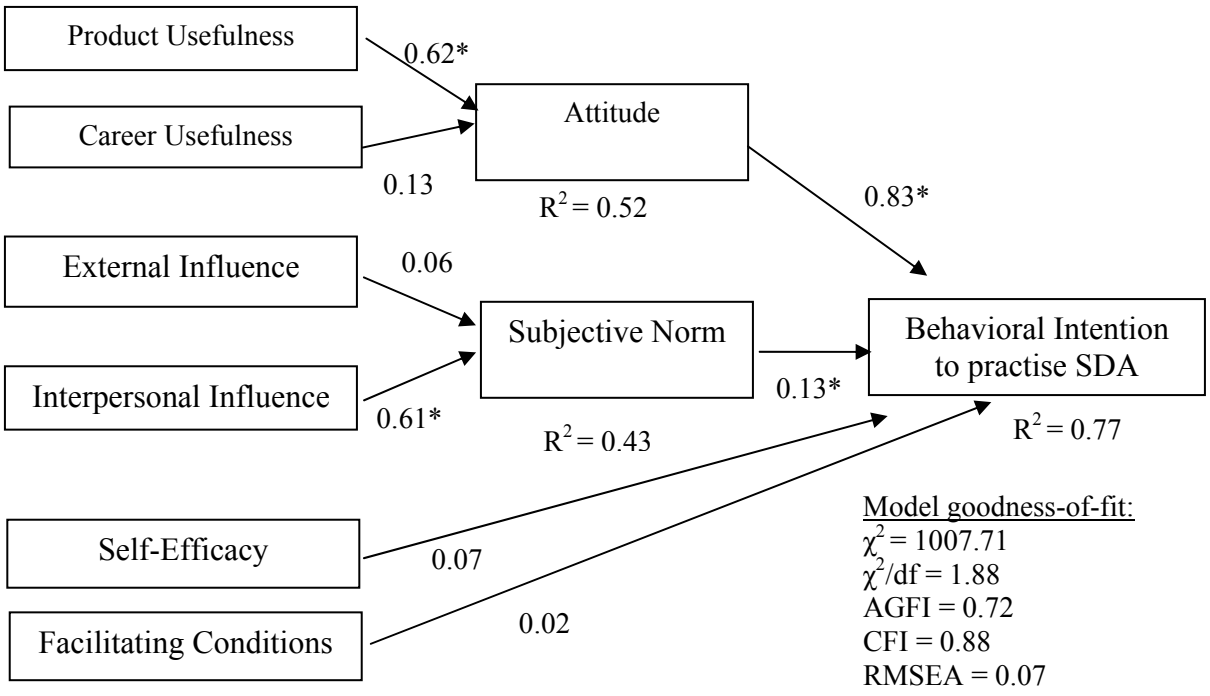


Figure 3: Standardized Path Coefficients and Model Fit Indices for TPB based Model

*Note: Paths significant at * $p < 0.05$*

First, the full research model based on the TPB was tested. As shown in Figure 3, the χ^2/df value of 1.88 indicated a valid model and the RMSEA value of 0.07 suggested a reasonable fit. However, the AGFI (0.72) and CFI (0.88) were somewhat lower than acceptable. Of the eight hypothesized paths in the research model based on TPB, four paths i.e., from product usefulness to attitude, interpersonal influence to subjective norm, attitude to behavioral intention, and subjective norm to behavioral intention, were significant. In other words, H1a, H2b, H5, and H6 were supported but H1b, H2a, H3, and H4 were not. The model accounted for 77.3% of the variance in behavioral intention, 52.0% of the variance in attitude, and 42.7% of the variance in subjective norm.

The non-significant paths from self-efficacy and facilitating conditions to behavioral intention seem to suggest that the TRA would be a better model to explain developers' intention to practise SDA. Therefore, the hypothesized research model based on the TRA was tested next.

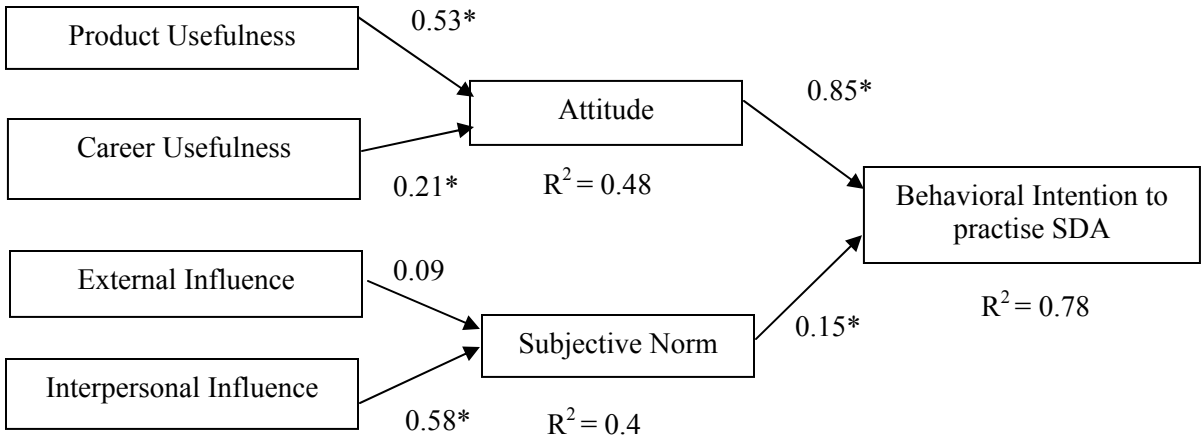


Figure 4: Standardized Path Coefficients and Model Fit Indices for TRA based Model

*Note: Paths significant at * $p < 0.05$*

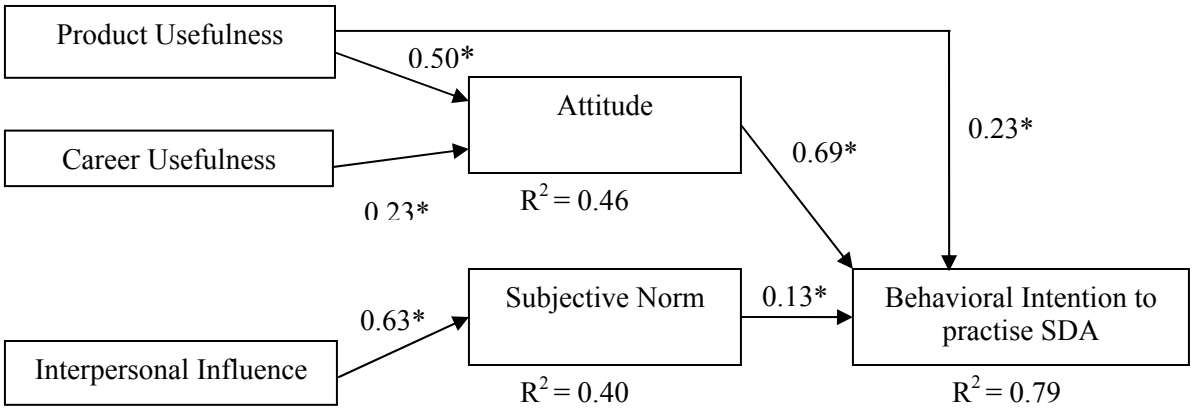
As shown in Figure 4, the model based on the TRA provided a reasonably good fit to the data (χ^2/df = 1.89; AGFI = 0.78; CFI = 0.91; RMSEA = 0.07), with improvements in all the model fit indices compared to the TPB model. However, AGFI was still below the acceptable level of 0.8. All paths in the TRA model were found to be significant, except the path from external influence to subjective norm i.e., all hypotheses except H2a were supported. The explained variance in behavioral intention increased slightly to 78%, while the explained variances in attitude and subjective norm decreased to 48% and 40% respectively.

The third model tested was based on the TRA but with two modifications. First, the non-significant path from external influence to subjective norm was deleted. Second, a direct link from product usefulness to behavioral intention was added to test whether this relationship would be significant, consistent with the Technology Acceptance Model [1](Davis et al 1989). The results shown in Figure 5 indicate that this model has the best fit amongst the three tested. All model fit indices are satisfactory, all hypotheses are supported, and 79% of variance in behavioral

[1] The technology acceptance model (TAM), derived from TRA/TPB, proposes that users come to accept and use a technology based on their beliefs of the perceived usefulness and perceived ease-of-use of the technology. While the initial conceptualization of TAM was for particular technologies, it has subsequently been applied successfully to explain adoption of collections of software, tools, and techniques (see Legris et al. 2003).

intention is explained.

After the data analysis, e-mails were sent to survey respondents to obtain more insights into the results. The feedback was useful in explaining the results and adding to our understanding of the phenomenon.



Model goodness-of-fit: $\chi^2 = 360.82$, $\chi^2/df = 1.81$, AGFI = 0.81, CFI = 0.94, RMSEA = 0.07

Figure 5: Modified TRA based Model (Note: Paths significant at * $p < 0.05$)

In order to better understand the phenomenon, we performed further analysis of different sub-groups of respondents to observe if there were differences in the final model (modified TRA-based model) results for the sub-groups. The respondents whose job responsibilities involved applications or software development could be divided into two groups. The first group consisted of Analysts and Programmers (APs) who are the implementers of the system. The second group consisted of Designers and Managers (DMs) who were responsible for the design and outcomes of the system development. Looking at the final model constructs (Figure 5), the main difference we expected between the two groups is with respect to the subjective norms (other antecedents such as product usefulness and career usefulness are likely to be similar for both groups). Interpersonal influence in terms of peer and superior influence is likely to be important for APs as compared to DMs. DMs are more likely to be influenced by their subordinates than their superiors in adopting new methods (Hardgrave et al 2003) such as SDA. There were 85 respondents in the AP group and 55 in the DM group. The results of running the analysis for each of these groups are shown in Table 5.

	All (184)	APs (85)	DMs (55)
Goodness of Fit			
χ^2/df (<3)	1.88	1.56	1.52
AGFI (> 0.8)	0.72	0.90	0.62
CFI (> 0.9)	0.88	0.70	0.86
RMSEA (< 0.08)	0.07	0.08	0.10
Std Regression Weights			
PU → Attitude	0.50*	0.49*	0.48*
CU → Attitude	0.23*	0.34*	0.42*
II → Subjective Norms	0.63*	0.62*	0.79
Subjective Norms → Intention	0.13*	0.71	0.12*
Attitude → Intention	0.69*	0.40*	0.49*
PU→ Intention	0.23*	0.53*	0.46*
Squared Multiple Correlations			
Subjective Norms	0.40	0.39	0.62
Attitude	0.46	0.58	0.54
Intention	0.79	0.77	0.86

* $p < 0.05$

Table 5: Modified TRA Model Results for APs vs DMs

As expected, the path from interpersonal influence (II) to subjective norms is not significant for the DMs. However, the relationship between subjective norms and intention is significant for them. As mentioned before, the major influence for SDA adoption for DMs could be from their subordinates, which is not captured in our measures of interpersonal influence. This bottom-up influence process would occur where developer interest in new methodologies could propel organizations towards adopting them (Hardgrave et al. 2003). The path from interpersonal influence (II) to subjective norms is significant for the APs. However, the relationship between subjective norms and intention is not significant for them. This situation could be explained in terms of the conflicting pressures that APs face i.e., to meet security requirements constrained by the specified time and functional goals. Thus although APs perceive the pressure to conform to their superiors and co-workers, their intention to practise SDA may not be affected as the pressure to meet deadlines and satisfy functionality requirements may be stronger (Davis et al 2004; Jones and Rastogi 2004).

5. DISCUSSION AND IMPLICATIONS

5.1 Discussion and Implications for Theory

The goal of this research was to identify the factors that would influence IS professionals' intention to practise secure development of applications (SDA). In addition, this research

examined the extent to which the Theory of Planned Behavior (TPB) and the Theory of Reasoned Action (TRA) can explain the intention to practise SDA. Compared to prior research using the TPB model (e.g., Bhattacherjee 2000; Taylor and Todd 1995; Mathieson 1991) or the TRA model (e.g., Karahanna et al 1999; Davis et al 1989), the total variance in the outcome variable explained in this study is considerably high at 77% for the TPB based model and 79% for the modified TRA based model i.e., both models offer substantial explanation.

The results of this research indicated that attitude has the strongest influence on intention to practise SDA, while subjective norm has a lesser influence on intention. This finding is consistent with prior research in which attitude was found to have a strong effect on intention while subjective norm had a relatively weaker effect on intention (Taylor and Todd 1995). Particularly, subjective norm is not a significant antecedent of intention for a subgroup in our sample i.e., analysts and programmers. Perceived behavioral controls, represented by self-efficacy and facilitating conditions, was not found to affect intention to practise SDA.

The lack of effect of perceived behavioral control on intention and the better model fit statistics and higher variance explained by the TRA based model compared to the TPB based model indicates that TRA is more appropriate to explain intention to practise SDA. This implies that the professionals surveyed have a high volitional control over carrying out SDA. Self-efficacy may not have been influential since the respondents did not have much experience with actually practising SDA. Also, since their organizations did not do much to facilitate practice of SDA other than allow the respondents to attend seminars on the topic, facilitating conditions were not significant in our study. This finding was further confirmed in the e-mail follow-up where the majority of the respondents indicated that they have considerable freedom in deciding whether or not to practise SDA. As one respondent noted, "*If I have time, then I might do it just to check it out and learn something. But if the deadline is very tight, then I will not risk it. As such, my organization does not support me to practise SDA nor does it ask me to.*" Their organizations do not reward or give recognition for the extra effort to practise SDA. Typically, top management is more interested in results and cares about this issue if negative publicity arises from an application compromise. These conclusions were suggested by statements like "*The management doesn't care about it (SDA) as long as nothing happens. They don't really care about how you*

program securely as long as you meet the costs and time for the project and the customer does not make any noise (i.e. is satisfied).”

The significant direct effect of product usefulness on intention was consistent with the predictions of the widely-used Technology Acceptance Model (TAM), which postulates that intention is jointly determined by attitude and perceived usefulness (Davis et al 1989). However, in contrast to the findings of Davis et al (1989) that perceived usefulness has a greater direct effect on intention than the indirect effect mediated through attitude, the results of this research indicated the opposite i.e., the indirect effect on intention appeared larger than the direct effect (in accordance with the TPB or TRA models).

Attitude towards practising SDA was explained mainly by product usefulness and to a lesser extent, by career usefulness. The significant influence of product usefulness indicates that professionals strongly believe that practising SDA will improve the security aspects of the application. This has been the experience of Microsoft when they adopted the security life-cycle approach to developing Windows Server 2003 (Lipner 2004) – numbers and severity of security vulnerabilities were noticeably reduced. Comparatively, IS professionals have weaker beliefs in the career usefulness of practising SDA. This was confirmed by the respondents in the email follow-up since organizations do not reward or recognise the effort to practise SDA. However management would be concerned if security breaches occur, and thus respondents would somewhat consider practise of SDA to be useful to secure their careers.

As expected, subjective norm was determined by interpersonal influence for analysts and programmers and for the sample as a whole. These findings appear to fit with the nested theory of structuration (Perlow et al 2004) where the individual professional's actions and interactions occur within the organizational context and are influenced by it. However, external influence did not appear to affect subjective norm. These findings agree with previous research on the effect of internal versus external influence on subjective norm (Bhattacherjee 2000). External influence may not have impacted subjective norm due to several reasons. First, the discipline of software security is immature compared to disciplines like computer science and software engineering. Thus, literature on the subject is scarce and until recently confined to publications of specific

interest groups. Second, the positive effect of media promotion of SDA may be countered by the external pressures that value functionality and time-to-market above application security (Anderson 2001; Davis et. al 2004).

5.2 Implications for Practice

The findings of this research indicate that the intention to practise SDA is largely influenced by individual's attitude towards practising SDA, which is determined by beliefs that practising SDA would be useful to the application and to a lesser extent, their career. This finding indicates that organizations intending to introduce SDA should highlight the product and career benefits of SDA. Product benefits include the increased robustness of resultant applications and the cost savings in comparison to retrofitting security after development. Career benefits of SDA include increasing the image, marketability, job security, and value of the individual IS professional practising it.

Further, most of the IS professionals in our sample, excluding analysts and programmers, are also influenced by perceived social pressure (subjective norm) to practise SDA. Therefore management can motivate individual employees to practise SDA through publicising its product and career benefits and subsequently use informal and formal networks to spread the message. This could assist in building a critical mass of enthusiasts for SDA.

Currently most organizations do not provide facilitating conditions for the practise of SDA. Professionals acquire SDA skills through on-the-job experience as most institutions of learning do not teach one how to design or write secure code (Conrath 2004; McCown 2002). This situation is gradually changing as more academic institutions are introducing several programs in computer security and there is an ongoing effort to institute a common body of knowledge (Crowley, 2003). However until this becomes more established, organizations that provide such training for their IS personnel will reinforce the importance of SDA as well as keep the employees abreast of the latest advances in software security and consequently, empower them to carry out robust application development.

The current situation of most organizations not recognising or facilitating the practise of SDA suggests that organizations must first be educated about the importance of practising SDA before they could start motivating IS professionals to practise it (Davis et. al 2004). The impetus for the needed change was given a boost when Microsoft announced its Trustworthy Computing Initiative in 2002 where it pledged to make its products more secure and remains committed to this promise two years later (Udel 2004). The latest report showing the effectiveness of the initiative (Lipner 2004) will no doubt provide further encouragement for those companies thinking of mandating SDA practice.

5.3 Limitations and Future Research

This research is a preliminary effort towards examining factors that influence intention to practise SDA based on a sample size of 184 IS professionals. Further research can be conducted with larger sample sizes across different industry sectors to validate the results. Another avenue of research would be to compare the similar applications of two organizations, one where SDA is mandated and one where it is not. This would provide sorely needed empirical evidence on the effectiveness and payoff of SDA e.g. data on the extra effort required and its associated costs for the application where SDA is mandated, data on the number and severity of security vulnerabilities of both applications, costs of the impact of vulnerabilities and the maintenance costs of both applications. At the same time, the differences in the attitudes of the developers in these two organizations would provide deeper insight into understanding their motivation and behavior.

This study also develops an instrument to measure developers' perceptions of SDA. Since this instrument measures the dependent variable through self-reported behavior, future research may attempt to devise more objective ways of measuring actual behavior with respect to SDA. Also measures may need to be developed to assess the costs and benefits of SDA practice in organizations.

Although our modified TRA based model of Figure 5 is largely successful in explaining intention to practise SDA, further research could examine other factors to increase explanatory power. Particularly factors that were mentioned by the respondents in the follow-up could be

included e.g., habit, awareness, and professional ethics. Another interesting avenue for research would be action or intervention based studies to observe how successful efforts to increase SDA practise may be.

6. CONCLUSION

This research investigated the factors influencing the intention of applications developers to practise secure development of applications (SDA). It compared the predictive power of the theory of planned behavior (TPB) and the theory of reasoned action (TRA) with respect to intention to practise SDA. As a part of the empirical validation, the study developed a survey instrument to measure the factors likely to impact intention. Product usefulness was found to have both an indirect (through attitude) and a direct effect on intention. Career usefulness and interpersonal influence had indirect effects on intention, mediated by attitude and subjective norm respectively. Intention was determined primarily by attitude, followed by product usefulness, and subjective norm. External influence did not impact subjective norm. Self-efficacy and facilitating conditions did not appear to impact intention to practise SDA. With the lack of effect of perceived behavioral control on intention, TRA based models were found to be a better predictor of intention to practise SDA than the TPB based model. It is hoped that through knowledge of the factors that influence developers' intention to practise SDA, companies would be better able to motivate developers to practise SDA and produce secure applications throughout the system's life cycle.

REFERENCES

Agarwal, R., and Prasad, J. (2000) A Field Study of the Adoption of Software Process Innovations by Information Systems Professionals, *IEEE Transactions on Engineering Management*, 47:3, pp. 295-308.

Ajzen, I. (1991) The Theory of Planned Behavior, *Organizational Behavior and Human Decision Processes*, 50, pp. 179-211.

Ajzen, I. (2002) Perceived Behavioral Control, Self-Efficacy, Locus of Control, and the Theory of Planned Behavior, *Journal of Applied Social Psychology*, 32, pp. 1-20.

Albarracin, D., Johnson, B.T., Fishbein, M., and Muellerleile, P.A. (2001) Theories of Reasoned Action and Planned Behavior as Models of Condom Use: A Meta-Analysis, *Psychological*

Bulletin, 127:1, pp. 142-161.

Anderson, R. (2001) Why Information Security is Hard – An Economic Perspective, *Annual Computer Security Applications Conference*, New Orleans.

Arkin, B., Stender, S and McGraw, G. (2005) Software Penetration Testing, *IEEE Security and Privacy*, pp. 84-87.

Bamberg, S., Ajzen, I., and Schmidt, P. (2003) Choice of Travel Mode in the Theory of Planned Behavior: The Roles of Past Behavior, Habit and Reasoned Action, *Basic and Applied Social Psychology*, 25:3, pp.175-188.

Bandura, A. (1977) Self-efficacy: Toward a Unifying Theory of Behavioral Change, *Psychology Review*, 84, pp. 191-215.

Bhattacherjee, A. (2000) Acceptance of E-commerce Services: The Case of Electronic Brokerages, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 30:4, pp. 411-420.

Cardono, M., and Frieze, I.H. (2000) Pollution Reduction Preferences of U.S. Environmental Managers: Applying Ajzen's Theory of Planned Behavior, *Academy of Management Journal*, 43:4, pp. 627-642.

CERT (2003). http://www.cert.org/advisories/#2003

Chau, P.Y.K. (1996) An Empirical Investigation on Factors Affecting the Acceptance of CASE by Systems Developers, *Information and Management*, 30, pp. 296-280.

Compeau, D., Higgins, C.A., and Huff, S. (1999) Social Cognitive Theory and Individual Reactions to Computing Technology: A Longitudinal Study, *MIS Quarterly*, 23:2, pp. 145-158.

Comrey, A.L. (1973) *A First Course in Factor Analysis*, New York, NY, Academic Press.

Conrath, C. (2004) Closing the Web App Hole, *Computer World* http://computerworld.co.nz/news.nsf/UNID/2D37638FD4A59D50CC256E7E0076FFB8?OpenDocument&Highlight=2,closing,the,web,app,hole

Cook, M., and Campbell, D.T. (1979) *Quasi-Experimentation: Design and Analysis Issues for Field Settings*, Boston, MA, Houghton Mifflin.

Cronbach, L. (1951) Coefficient Alpha and the Internal Structure of Tests, *Psychometrika*, 16:3, pp. 297-334.

Crowley, E. (2003). Information system Security curricula development, *Proceedings of the 4th conference on Information technology curriculum*, Lafayette, Indiana, USA, October 16 - 18, 2003, pp.249-255.

Davis, F.D., Bagozzi, R.P., and Warshaw, P.R. (1989) User Acceptance of Computer Technology: A Comparison of Two Theoretical Models, *Management Science*, 35:8, pp. 982-1003.

Davis, N., Humphrey, W., Redwine, S.T., Zibulski, G. and McGraw, G (2004) Processes for Producing Secure Software, Summary of US National Cybersecurity Summit Subgroup Report, *IEEE Security and Privacy*, pp. 18-25.

Desmond, P. (2004) All-out blitz against Web app attacks, Network World, 17th May 2004, http://www.networkworld.com/techinsider/2004/0517techinsidermain.html

Dooley, D. (2001) *Social Research Methods,* (4th ed.) Upper Saddle River, NJ, Prentice-Hall.

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

Flechais, I., Sasse, M.A. and Hailes, S.M.V. (2003) Bringing Security Home: A process for developing secure and usable systems. *New Security Paradigms Workshop* http://www.softeng.ox.ac.uk/personal/Ivan.Flechais/downloads/nspw2003.pdf

Gefen, D., Straub, D.W., and Boudreau, M. (2000) Structural Equation Modelling and Regression: Guidelines for Research Practice, *Communications of AIS*, 4:7.

Green, G., and Hevner, A.R. (1999) Perceived Control of Software Developers and Its Impact on the Successful Diffusion of Information Technology, *CMU/SEI-98-SR-013*, Carnegie Mellon, Software Engineering Institute, Pittsburgh, PA.

Hardgrave, B.C., Davis, F.D. and Riemenschneider, C. K. (2003) Investigating Determinants of Software Developers' Intention to Follow Methodologies, *Journal of Management Information Systems*, 20:1, pp. 123-151.

Hardgrave, B.C. and Johnson, R.A. (2003) Towards an Information Systems Development Acceptance Model: The Case of Object-Oriented Systems Development, *IEEE Transactions on Engineering Management*, 50:3, pp. 322-336.

Hu, L., and Bentler, P. M. (1999) Cutoff Criteria for Fit Indexes in Covariance Structural Analysis: Conventional Criteria versus New Alternatives, *Structural Equation Modelling*, 6:1, pp. 1-55.

Iivari, J. (1996) Why are CASE Tools Not Used? *Communications of the ACM*, 39:10, pp. 94-103.

Johnson, R.A., Hardgrave, B.C., and Doke, E.R. (1999) An Industry Analysis of Developer Beliefs about Object-Oriented Systems Development, *The DATA BASE for Advances in Information Systems*, 30:1, pp. 47-64.

Jones, R.L. and Rastogi, A. (2004) Secure Coding: Building Security into the Software Development Life Cycle, *Information Systems Security*, 13:5, pp. 29-39.

Karahanna, E., Straub, D.W., and Chervany, N.L. (1999) Information Technology Adoption across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs, *MIS Quarterly*, 23:2, pp. 183-213.

Landwehr, C.E., Bull, A.R., McDermott, J.P. and Choi, W.S. (1994) A Taxonomy of Computer Program Security Flaws, *ACM Computing Surveys*, 26:3, pp. 211-254.

Legris, P., Ingham, J., and Collerette, P. (2003) Why Do People Use Information Technology? A Critical Review of the Technology Acceptance Model, *Information and Management*, 40:3, pp. 191-204.

Lending, D. and Chervany, N. L. (1998) The Use of CASE Tools, *Proceedings of the 1998 ACM SIGCPR Conference on Computer Personnel Research*, pp. 49-58.

Lipner, S. (2004) The Trustworthy Computing Security Development Lifecycle, *20th Annual Computer Security Applications Conference*, http://www.acsa-admin.org/2004/papers/Lipner.pdf

Mathieson, K. (1991) Predicting User Intentions: Comparing the Technology Acceptance Model with the Theory of Planned Behavior, *Information Systems Research,* 2:3, pp. 173-191.

Mead, N. and Stehney, T. (2005) Security quality requirements engineering (SQUARE) methodology, *ACM SIGSOFT Software Engineering Notes,* 30:4, pp. 1-7.

McCown, C. (2002) Framework for Secure Application Design and Development, Foundation, Principles and Design Guidelines, *SANS Institute* http://www.sans.org/rr/whitepapers/application/842.php

McGraw, G. (2004a) Software Security, *IEEE Security and Privacy*, March/April, pp. 80-83

McGraw, G. (2004b) Software Security Testing, *IEEE Security and Privacy*, Sept/Oct, pp. 81-85.

Moore, G.C., and Benbasat, I. (1991) Development of an Instrument to Measure the Perceptions

of Adopting an Information Technology Innovation, *Information Systems Research*, 2:3, pp. 192-222.

National Institute of Standards and Technology (2002) The Economic Impacts of Inadequate Infrastructure for Software Testing. www.nist.gov/director/prog-ofc/report02-3.pdf

Nunnally, J. (1978) *Psychometric Theory*, New York, McGraw-Hill.

Pedersen, P.E. (2001) Adoption of Mobile Commerce: An Exploratory Analysis, *SNF-report no. 51/01,* Foundation for Research in Economics and Business Administration, Bergen, Norway, http://ikt.hia.no/perep/snf_51_2001.pdf

Perlow, L.A., J.H. Gittell, and N. Katz (2004) Contextualizing Patterns of Work Group Interaction: Toward a Nested Theory of Structuration, *Organization Science*, 15:5, pp. 520-536.

Riemenschneider, C.K., Hardgrave, B.C., and Davis, F.D. (2002) Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models, *IEEE Transactions on Software Engineering*, 28:12, pp. 1135-1145.

Schindler, E (2004) Application Developers need to redouble security efforts, *eWeek*, 30th Sept. 2004 http://www.eweek.com/article2/0,1895,1663671,00.asp

Schneier, B. (2000) The Process of Security, *Information Security Magazine*. http://infosecuritymag.techtarget.com/articles/april00/columns_cryptorhythms.shtml

Schwartz, M. (2003) Q&A: Arresting Bugs Earlier in Development Cycle Cuts Security Costs, *Enterprise Systems*, http://www.esj.com/news/article.asp?EditorialsID=801

Siponen, M., Baskerville, R and Kuivalainen, T. (2005) Integrating Security into Agile Development Methods, *Proceedings of the 38th Hawaii International Conference on System Sciences* http://csdl2.computer.org/comp/proceedings/hicss/2005/2268/07/22680185a.pdf

Straub, D.W. (1989) Validating Instruments in MIS Research, *MIS Quarterly*, 13:2, pp. 147-169.

Taylor, S., and Todd, P.A. (1995) Understanding Information Technology Usage: A Test of Competing Models, *Information Systems Research*, 6:2, pp. 144-176.

Thompson, R.L., Higgins, C.A., and Howell, J.M. (1991) Personal Computing: Toward a Conceptual Model of Utilization, *MIS Quarterly*, 15:1, pp. 125-143.

Udel, J. (2004). Is Microsoft trustworthy yet? 8th Oct. 2004, http://www.infoworld.com/article/04/10/08/41FEmssecure_1.html?s=feature

Venkatesh, V. (2000) Determinants of Perceived Ease of Use: Integrating Control, Intrinsic

Motivation, and Emotion into the Technology Acceptance Model, *Information Systems Research*, 11:4, pp. 342-365.

Verton, D. (2002) Airline Web Sites Seen as Riddled with Security Holes, *Computer World*.http://www.computerworld.com/securitytopics/security/story/0,10801,67973,00.html

Viega, J., Kohno, T. and Potter, B. (2001) Trust and Mistrust in secure applications, *Communications of the ACM*, 44:2, pp. 31-36

Viega, J., and McGraw, G. (2002) *Building Secure Software*, Boston, Addison-Wesley.