APPLICATION SECURITY SELF-EFFICACY OF SOFTWARE DEVELOPERS: A CORRELATIONAL STUDY

Wisdom Umeugo

Independent Researcher, Ottawa, Canada

ABSTRACT

Practicing the secure software development lifecycle (SSDLC) requires a team of application security experts and a proper management structure. Software developers usually lack application security expertise but are often responsible for software security when management undervalues software security, or the SSDLC is too expensive to practice. Software developers must have adequate application security self-efficacy (SE) to execute software security activities and processes effectively. To improve software developers' SE, the factors that impact their SE must be identified. A sample of 200 software developers based in the United States was surveyed for their SE. The relationship between the factors and SE was analyzed using Spearman's rho correlation. Application security awareness, the presence of an application security team, and education level all correlated with software developers' SE. Security training and performing multiple application securities did not correlate with software developers' SE. The results have practical implications for improving software security.

KEYWORDS

Self-efficacy, Application security, Software security, SSDLC, Secure Software development

1. INTRODUCTION

Software security remains vital due to software's increasing value and pervasiveness [1]. Security must be implemented into all stages of the software development lifecycle to ensure that software is released secured [2], [3]. In other words, a secure software development lifecycle (SSDLC) must be practiced. Software security processes and activities are complex, requiring specialized skills and staff, such as application security architects and engineers [4]. However, there are obstacles to practicing the SSDLC, such as the cost and complexity involved and management's reluctance to prioritize software security [4]–[6]. In such situations, application security responsibilities are left to software developers, who often have limited application security expertise. Regardless of whether the SSDLC is practiced, software developers must possess appropriate application security awareness and self-efficacy. Self-efficacy has been shown to positively influence security behavior and innovation adoption [7]–[10]. Therefore, developers with greater application security self-efficacy are more likely to emphasize and effectively implement security features and adopt new tools and practices that improve the security of their developed software. Understanding the factors that impact the application security self-efficacy of software developers is critical to software security improvement efforts. While self-efficacy is a well-researched topic in information security, there is a knowledge gap in the research literature on application security self-efficacy and the application security self-efficacy of software developers. This study empirically examined the relationship between software developers' application security self-efficacy and experience inertia, knowledge inertia, application security awareness, security training, presence of an application security team, application security roles

DOI: 10.5121/ijcsit.2023.15201

performed, and education level. The results of this research are vital to the application security improvement efforts of information security administrators of organizations that develop software.

2. Background and Literature Review

2.1. Secure Software Development

Software must have security built in from the earliest stages of its lifecycle to be secure. This involves incorporating security practices into each stage of the software development lifecycle. According to Umeugo [6], developing secure software requires security tools, security policies, a secure software framework, knowledgeable staff, and a proper management structure to guarantee that security is emphasized as part of software quality assurance. Typical software security processes and activities include risk management, security requirements assessments, security architecture definition, attack surface determination, secure coding, static and dynamic application security testing, penetration testing, security training, and environment hardening [6]. Application security architects, engineers, and champions are the skilled staff that plans and oversee these software security processes and activities [4]. The role of Software developers in software security is primarily limited to security activities in the development phase of the SDLC, such as secure coding and application security testing. However, in many cases, software developers are expected to perform other application security tasks, especially when application security architects and engineers are unavailable. Various research works have argued against leaving application security responsibilities to software developers to manage and implement. Gasiba et al. [11] showed that software developers generally lacked awareness and knowledge of secure software development practices and guidelines. Software developers often need to realize the importance of security requirements and the implications of omitting these requirements [2]. When developers realize the importance of security requirements, they may be too complex to understand and implement effectively [12]. Software developers must be given concrete guidance and guidelines on implementing security features; otherwise, the implementation may fail to adequately meet requirements, resulting in vulnerabilities in the software [2], [12]. Where developers are saddled with executing application security tasks, they must be provided with proper application security training. Figure 1 pictorially illustrates the SSDLC.

2.2. Self-Efficacy

Self-efficacy is a social cognition theoretical construct that refers to a person’s self-belief in their ability to perform a given task [13]. Self-efficacy is considered a predictor of motivation, personal goal-setting, and task performance [13]. According to Bandura [14], self-efficacy beliefs strongly influence people’s choices, goals, level of effort, and perseverance in the face of failure. Perceived self-efficacy is also a predictor of the amount of stress and depression people feel when faced with a perceived daunting task [13]–[15]. Increasing employee self-efficacy in various areas is essential to organizational goal-setting. As a predictor of behavior, Bandura [16] stressed that self-efficacy is a form of response expectancy that determines attitude towards a given behavior. According to [16], two types of personal expectancy influence behavior: outcome and efficacy expectancy. Outcome expectancy is an individual’s estimate that the given behavior will produce the expected outcome. Efficacy expectancy is the individual’s belief that they can successfully execute the behavior to obtain the desired outcome. Suppose an individual retains any doubt in their ability to execute a given behavior regardless of the belief that the behavior will produce the given outcome. In that case, the behavior is unlikely to be executed. Therefore, self-efficacy predicts the likelihood of an individual’s execution of a given behavior, goal, or activity. This explains the importance of self-efficacy in behavior studies.

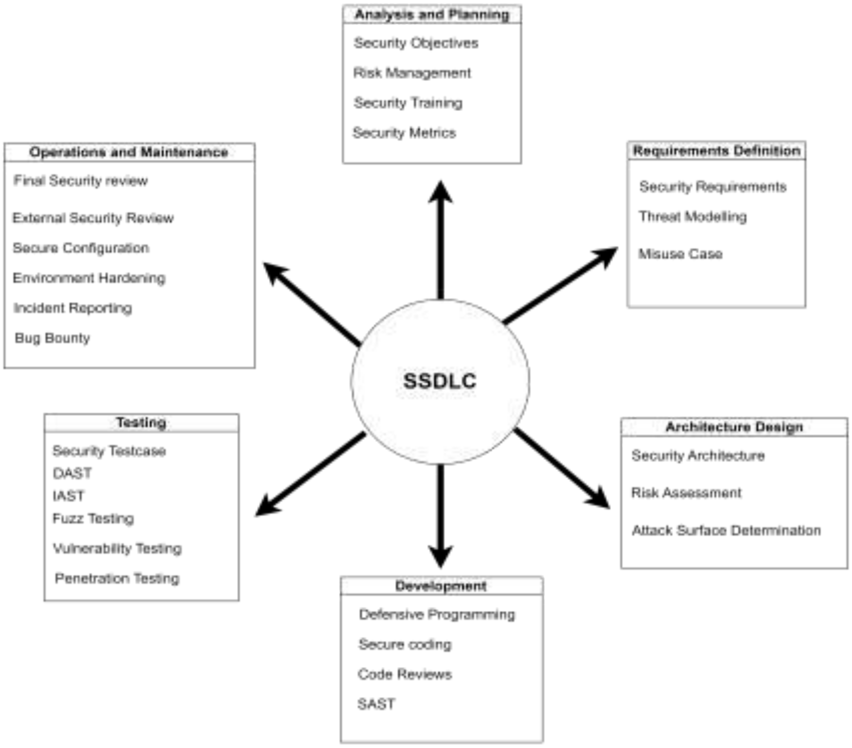

Figure 1. The SSDLC

Self-concept and self-efficacy, although related, are different. While self-concept is an individual's general perception of self in a given domain, self-efficacy is the individual's expectation or belief of what one can accomplish with one's skills and abilities in a given situation [17]._ For example, the expectation that one can update a computer's antivirus software without prior experience is an efficacy judgment because it relies on the strength of one's belief and not an estimate of one's competency or expertise at the given task.

There are three dimensions of self-efficacy: magnitude, strength, and generality [13]. The magnitude dimension of self-efficacy is the level of task difficulty a person believes they can attain [13]. The strength self-efficacy dimension is the degree of belief in the attainability of a certain level of task performance [13]. Generality refers to the extent of applicability of a given self-efficacy belief across different situations [13].

There are Four primary sources of self-efficacy in the literature: performance accomplishments or mastery, vicarious experience, verbal or social persuasion, and emotional arousal or physiological state [13], [16]. Performance accomplishments refer to past performance success, reinforcing the perception of self-efficacy. Repeated successes raise mastery expectations, while repeated failures lower them [16]. Individuals develop resilient self-efficacy through success at challenging tasks that result from perseverance efforts [13]. On the other hand, repeated successes at easy tasks do not help develop resilient self-efficacy because such perceptions of self-efficacy result in the expectation of low efforts in challenging tasks, resulting in failures [13]. Vicarious experiences are the second source of self-efficacy. Vicarious experiences refer to peer observations of successes in a given task. Observation provides information for inference on how one will perform at the same task [15]. Vicarious experiences rely on social comparison and instill a higher sense of self-efficacy by making the observer feel they can succeed at the given task if others can [16]. Vicarious experiences result in the acceptance of a model behavior with all the efficacy information and clear outcomes by an individual [16]. These models then become a source of goal setting, motivation, and a target of imitation [15].

Verbal or social persuasion is verbal or social influences aimed at providing encouragement and motivation to an individual during challenging tasks [15]. Through verbal persuasion, individuals are led to believe they can successfully cope and achieve the expected outcome. However, social persuasion must not create unrealistic high expectations, which may lead to adverse effects upon failure [13]. Encouragement from family, friends, co-workers, tutors, and professionals are examples of social persuasion [15]. Social persuasion is considered a less effective self-efficacy source than performance accomplishments and vicarious experiences [13], [16].

Emotional or physiological arousal plays a part in self-efficacy judgments. Activities may induce emotional states such as stress, fear, anxiety, agitation, and pain in individuals. Such aversive emotional arousal may evoke fear of ineptitude and impending failure [16]. This is because individuals use physiological feedback to judge their capability at a given task [15]. For example, emotional arousal of fatigue and pain can reduce the perception of self-efficacy in physically-demanding tasks. Efforts to improve self-efficacy should reduce the amount of negative emotional arousal perceived in the task [13], [15].

Self-efficacy influences an individual's choice of activities and environments. Individuals avoid activities they expect to be unable to cope with and complete. Still, they readily undertake more challenging tasks in an environment they feel they can cope in [18]. This makes self-efficacy relevant to behavioral intention, which is reflected in the prevalence of self-efficacy as a factor in information security behavioral studies. Hooper and Blunt [19]_found self-efficacy to be an influential determinant of employee information security behavioral intention. Bulgurcu et al. [20] reported that an employee's intention to comply with information security policies was significantly influenced by self-efficacy. White et al. [21] showed that self-efficacy influenced protective behavior. Self-efficacy also positively correlated with phishing threat avoidance motivation and avoidance behavior in Arachchilage and Love's [22] study. Similarly, Self-efficacy positively influenced security practice behavior and security-strengthening efforts in various other studies [7]–[9]. Most information security behavioral studies concerned with self-efficacy as a factor adopted the Health belief theoretical model [8], [9], [23]. In information technology behavioral studies, self-efficacy has also been proven a moderator to various technology usage intentions [24], [25]._Information systems innovation adoption studies have also considered self-efficacy a factor in innovation adoption [26], [27].

Management must instill high self-efficacy in employees in the tasks they execute. Organizations set goals that can only be achieved through the high self-efficacy of their employees. Self-efficacy beliefs influence individuals' perseverance and resilience when confronted with challenging tasks and impending failure [18], [28]. Low self-efficacy leads to capitulation that results in either failure or acceptance of a mediocre option [18], [28]. Management must view the attainment of high self-efficacy by employees in a given field as important as possessing the skills for the field themselves [28].

2.3. Related Works

While various studies in information security have attempted to measure employee security self-efficacy, only some have yet to do so in a descriptive empirical way. Various information security behavioral studies have also examined self-efficacy as a factor in relationships with various other factors and behavioral outcomes. However, none have evaluated the application security self-efficacy of software developers or examined the factors that impact software developers' application security self-efficacy. Ambrose [29] noted that the usual methods to assess developer competency using experience, education and training, academic and professional references, tests, and interviews were inadequate. Ambrose [29] argued for a holistic assessment of software developer competence by including self-measurements of developer self-efficacy in competency evaluations. Ambrose [29] developed a 10-point programmer self-efficacy scale anchored at 'Not at all Confident' and 'Totally Confident.' The model was validated, and self-efficacy strongly correlated with self-awareness.

Anantharaman et al. [30] examined the role of self-efficacy, collective efficacy, and perception of control in the occupational stress of software developers in India. Data was collected from 156 software developers. Multiple regression analyses of the data showed that self-efficacy, collective efficacy, and perception of control were found to moderate the negative consequences of stress due to work exhaustion, organizational commitment, and turnover intent.

Woon and Kankanhalli [31] included self-efficacy in their investigation of the factors that may influence the intention of information systems (IS) professionals to practice Secure development of applications (SDA). Self-efficacy was measured using three items on their questionnaire that asked developers if they were comfortable carrying out SDA on their own, If they could carry out SDA reasonably well on their own, and if they could carry out SDA without the help of others. Woon and Kankanhalli [31] found that self-efficacy was insignificant to developers' behavioral intention to carry out SDA.

Witschey et al. [32] investigated developers' adoption of security tools based on the Diffusion of innovation theory. Their study examined factors such as social system factors describing the organizational culture, communication channel factors representing where developers learn about the tools, and potential adopter factors explaining the individual developer adoption intentions. Results showed that the presence of security policies, a positive security culture, security team structures, developer security training, exposure to tools, and developer inquisitiveness positively impacted the developer's intention to adopt security tools.

Deschene [33] qualitatively studied the requirements for adopting software security practices. A three-round Delphi study was conducted on an expert panel of software development stakeholders. Results revealed that security awareness and education were among the critical factors for improving secure software practices adoption.

Senarath and Arachchilage [34] performed an empirical study by interviewing 36 software developers to determine the challenges they faced incorporating privacy into the software they developed. Results revealed five issues raised by developers: contentions between functional and privacy requirements, issues translating privacy requirements into privacy techniques, inadequate knowledge of testing and verification of implemented privacy features, lack of understanding of privacy practices, and contention between personal beliefs and privacy requirements.

3. HYPOTHESES DEVELOPMENT

3.1. Application Security Self-efficacy

Application security self-efficacy (SE) is the belief in one's ability to complete application security tasks [35]. Application security tasks include identifying security problems during software design, threat analysis, selecting and implementing security controls, security communications, and secure programming. Umarji and Seaman [26] considered self-efficacy as a factor impacting acceptance in Software Process Improvement (SPI). SPI introduces new knowledge, tools, and techniques. Umarji and Seaman [26] viewed developer self-efficacy as a good indicator of the capability to perform SPI activities. Software developers with high SE are expected to have more intention to develop applications securely [31]. However, Woon and Kankanhalli [31]_reported that self-efficacy did not impact software developers' intention to practice secure application development in their study. In this study, software developer SE is measured using the secure software-development self-efficacy scale (SSD-SES) developed by Votipka et al. [35]. SSD-SES is a 15-item scale with two sub-scales: vulnerability identification and mitigation and security communications.

3.2. Application Security Awareness

Information security awareness (ISA) is usually defined in terms of knowledge of information security policies, rules, and guidelines and behavior following these policies, practices, and guidelines [36]. The Knowledge-Attitude-Behaviour (KAB) model is frequently used to determine awareness based on the idea that an employee's knowledge of security behaviors corrects their attitude, resulting in improved information security behaviors [37]. Similarly, application security awareness is concerned with the knowledge of the significance of application security policies, rules, guidelines, and practices.

The research literature often reports a positive relationship between self-efficacy and awareness. Ryan [38] noted that individuals with high computer self-efficacy have high information security awareness. Ryan [38] also reported a positive correlation between ISA factors and computer self-efficacy. Deschene [33] found that application security awareness was critical to improving SSDLC adoption. Based on the highlighted positive relationship between application security awareness and Self-efficacy, this study posits a positive relationship between software developers' application security awareness (AW) and their SE.

H1: There is a positive correlation between Application security Awareness and Application security self-efficacy.

3.3. Application Security Training

Application security training imparts application security awareness, knowledge, and practical skills. Training has a widely-accepted positive influence on self-efficacy. Torkzadeh and Van Dyke [39] reported that training increased the internet self-efficacy of participants in their study. He et al. [40] included the impact of various cybersecurity training methods on employee cybersecurity self-efficacy in their research. They found that self-efficacy increased for all training methods. Arachchilage and Love [22] found that the interaction effect of conceptual and procedural knowledge positively impacted the self-efficacy of phishing threat avoidance. According to Witschey et al. [32], developer security training and exposure to tools positively impacted the developer's intention to adopt security tools. Training provides a conducive environment for performance accomplishments, vicarious experience, and verbal persuasion to thrive. This study, therefore, posits a positive relationship between application security training (TR) and developer SE.

H2: There is a positive correlation between Application security training and Application security self-efficacy.

3.4. Presence of Application Security Team

The presence of an application security team, including application security engineers and architects, is one of the requirements for implementing the secure software development lifecycle (SSDLC) [6]. The presence of security team structures positively impacts developers' intention to adopt security tools [32]. Having an application security management team may also boost collective self-efficacy. Collective self-efficacy provides an effective environment for other efficacy sources, such as mastery experience in a group, observational learning, and social persuasion, to thrive [41]. Mastery experience in a group can make employees feel more empowered and capable, increasing their perception of self-efficacy [41]. Observational learning and social persuasion also boost self-efficacy by demonstrating observable solutions to various problems and providing the necessary feedback on performing specific tasks [41]. It is therefore hypothesized that there is a positive correlation between the presence of an application security team (ST) and software developer SE.

H3: There is a positive correlation between the Presence of an application security team and Application security self-efficacy was supported.

3.5. Number of performed Application Security Activities

Performing multiple related activities is naturally expected to increase knowledge and expertise. Performance accomplishment is an influential source of self-efficacy [16]. Repeated mastery and success in activities raise mastery expectations and increase self-efficacy, which, when enhanced, tend to generalize to related activities [16]. For this reason, a positive relationship between the number of application security activities (NA) performed by software developers and their SE is hypothesized to exist.

H4: There is a positive correlation between the number of application security activities developers perform and their application security self-efficacy.

3.6. Education Level

Higher education levels typically mean greater skills, theoretical understanding, and higher accomplishments which are likely to increase self-efficacy. Education level may also contribute to a developer’s past experiences if the developer undertook application security training in a formal education setting, usually at the college and postgraduate level. For this reason, a positive relationship between software developers’ education level (EL) and their SE is hypothesized.

H5: There is a positive correlation between developers’ education level and Application security self-efficacy.

Figure 2 shows the conceptual model used in the study.

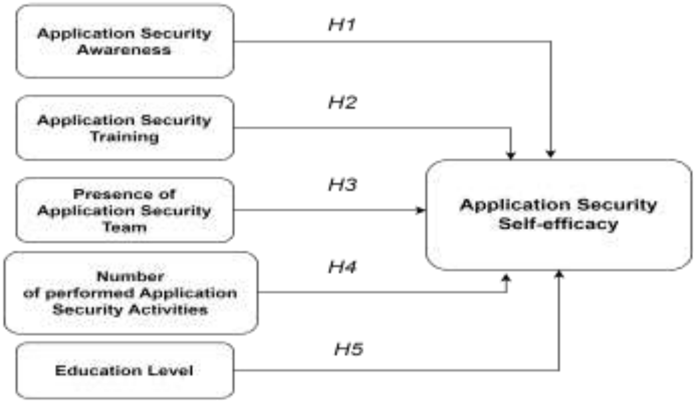
Application Security Awareness
Application Security Training
Presence of Application Security Team
Number of performed Application Security Activities
Education Level
H1
H2
H3
H4
H5
Application Security Self-efficacy

Figure 2. Conceptual model showing the variables.

4. RESEARCH METHODOLOGY

The research was designed to be quantitative non-experimental correlational. An online close-ended survey questionnaire hosted on Pollfish was used as the instrument. The survey was shown to a population of software developers based in the United States working full-time in the software industry. The survey contained close-ended demographic questions requesting age range, sex, application security roles in the participant's organization, and application security activities performed by participants. The questionnaire included the five-item Socially Desirable Response Scale (SDRS-5) proposed by [42] to measure social desirability bias. Participants' application security self-efficacy (SE) was measured using the 15-item validated scale measuring secure software-development self-efficacy scale (SSD-SES) by Votipka et al. [35]. To reduce social desirability bias, participants were informed of the anonymous nature of the survey and asked to answer truthfully. A total of 200 valid responses were accepted after removing incompletes and speeders. The resulting data was imported into Jamovi for statistical analysis. Descriptive statistical analysis was performed to examine the study's demographics. Participants' SE scores were averaged to obtain a mean score of 5 or less. The interpretation of the SE score was based on Salem et al. [43]. A score between 4.5 and 5 was interpreted as high SE, 4.0 – 4.5 average, and scores lower than 4.0 were judged poor SE. Spearman's Rho inter-variable correlation was performed to determine correlations between the SE and AW, TR, NA, ST, and EL.

5. RESULTS

A total of 200 responses were accepted after data quality filtering. 102 or 51% of the participants were female, and 98 (49%) were males. Most participants were aged between 35 – 44 (*n*=82) and 25-34 (*n*=73). One hundred twenty-seven participants stated that they did not have application security training, while 73 said they had application security training. One hundred eighty-four participants, or 92%, claimed their organization had a security team in place. Table 1 summarizes the study's participant demographics.

Table 1. Participant demographics

Demographic variable	**Group**	**Counts**	**% of Total**
Gender	female	102	51.0 %
	male	98	49.0 %
Age	18 - 24	23	11.5 %
	25 - 34	73	36.5 %
	35 - 44	82	41.0 %
	45 - 54	16	8.0 %
	> 54	6	3.0 %
Education	Elementary school	1	0.5 %
	High school	10	5.0 %
	Middle school	2	1.0 %
	Postgraduate	66	33.0 %
	University	98	49.0 %
	Vocational college	23	11.5 %
Application Security Training	No	127	63.5%
	Yes	73	36.5%
Security team	No	16	8.0%
	Yes	184	92.0%

Participants were asked about application-security activities they were involved in. Application security training was the most frequently cited application security activity, with 36.5% of participants stating they underwent it. Managing the security risk of using third-party components (34.50%) and Dynamic Application Security Testing (33.00%) were the following two most performed application security activities. Table 2 shows participants' application security activities demographics. Table 3 summarizes the frequency of the number of application security roles developers perform.

Table 2. Participants' application security activities demographics

Application security activity	**Respondents(%)**	**Answers(%)**	**Count**
Static Application Security Testing (SAST)	32.00%	8.60%	64
Dynamic Application Security Testing (DAST)	33.00%	8.87%	66
Secure coding	32.50%	8.74%	65
Penetration Testing	22.50%	6.05%	45
Selecting cryptography Standards	20.00%	5.38%	40
Managing the security risk of using third-party components	34.50%	9.27%	69
Threat modeling	15.00%	4.03%	30
Security requirements engineering	26.00%	6.99%	52
Security training	36.50%	9.81%	73
Attack surface determination	20.50%	5.51%	41
Vulnerability testing	28.00%	7.53%	56
Security architecture definition	25.00%	6.72%	50
Environment Hardening	15.50%	4.17%	31
Bug bounty management	13.50%	3.63%	27
DevSecOps	14.50%	3.90%	29
None of the above	3.00%	0.81%	6

Table 3. Frequency of the number of application security roles

No. of Activities	**Counts**	**% of Total**
0	6	3.0 %
1	18	9.0 %
2	33	16.5 %
3	55	27.5 %
4	37	18.5 %
5	19	9.5 %
6	10	5.0 %
7	8	4.0 %
8	6	3.0 %
9	3	1.5 %
10	2	1.0 %
11	1	0.5 %
12	2	1.0 %

Participants were also asked for the presence of several application security roles, including Application security managers and architects. Approximately 62% (*n* = 124) of respondents said their organization had Application security architects, 57% (*n* = 114) said they had application security managers, 46% (*n* = 92) had application security engineers, 23.5% (*n* = 47) had application security champions, 32.50% (*n* = 65) had DevSecOps engineers, 33% (*n* = 66) had Application security testers, and 30.5% (*n* = 61) said their organization employed Application penetration testers. 1.5% (*n* = 3) said their organization hired none of the security roles. Based on the data, 92% % (*n* = 184) of the participants had an application security management team in place, while 8% % (*n* = 16) did not. Table 4 summarizes the demographics of Application security roles of participants' organizations.

Table 4. Application security roles

Answers	Respondents(%)	Answers(%)	Count
Application security architects	62.00%	21.68%	124
Application security engineers	46.00%	16.08%	92
Application security managers	57.00%	19.93%	114
Application security champions	23.50%	8.22%	47
Application penetration testers	30.50%	10.66%	61
DevSecOps engineers	32.50%	11.36%	65
Application security testers	33.00%	11.54%	66
None of the above	1.50%	0.52%	3

The effect of social desirability bias (SDB) was evaluated by examining Spearman's rho for the correlation of all items with the social desirability bias (SDB) variable. The inter-variable correlation with SDB is presented in Table 5. Self-efficacy (r_s(298) = . 143, *p* < .05) statistically significantly correlated with SDB. The Spearman's rho correlation value is 0.3 or less, a weak correlation [44]. There was no statistically significant correlation between SDB and awareness (r_s(298) = -0.055, *p* > .05). Therefore, the effects of SDB can be ignored in this study.

Table 5. SDB Factor Correlations

Variable	Statistic	AW	SE
SDB	Spearman's rho	-0.055	-0.143*
	p-value	0.438	0.044

Note: * p < .05, ** p < .01, *** p < .001

Participants had a mean SE score of 3.82, an average SE score. A Spearman's rho correlation matrix of all hypothesis variables was calculated to test hypotheses one to five. Table 6 shows the variable correlation matrix.

Table 6. Variable correlation matrix

		AW	TR	ST	NA	EL
AW	Spearman's rho	—				
	p-value	—				
TR	Spearman's rho	-0.033	—			
	p-value	0.639	—			
ST	Spearman's rho	0.183**	-0.006	—		
	p-value	0.009	0.931	—		
NA	Spearman's rho	0.013	0.339***	0.112	—	
	p-value	0.857	< .001	0.116	—	
EL	Spearman's rho	0.239***	-0.072	0.138	0.013	—
	p-value	< .001	0.311	0.051	0.85	—
SE	Spearman's rho	0.542***	-0.121	0.211**	-0.027	0.316***
	p-value	< .001	0.088	0.003	0.707	< .001

Note: * p < .05, ** p < .01, *** p < .001

5.1. Hypothesis One

Application security Awareness correlated significantly and positively with Application security self-efficacy ($r_s(298) = .542, p < .001$). Hypothesis One, stating that there is a positive correlation between application security awareness and application security self-efficacy, was supported.

5.2. Hypothesis Two

The mean SE for participants with security training (*N*=127) was 3.9, and 3.67 for participants without security training (*N*=127) was 3.9. There was no correlation between ST and SE ($r_s(298) = -0.121, p > .05$). Spearman's rho calculated for the relationship showed neither statistical significance nor positive correlation. Hypothesis Two, stating that there is a positive correlation between application security training and application security self-efficacy, was unsupported.

5.3. Hypothesis Three

The mean SE for participants with an application security team (*N*=184) was 3.86, and 3.28 for participants without an application security team (*N*=16) was 3.86. Spearman's Rho calculated for the relationship between the presence of an application security team and developers' application security self-efficacy was $r_s(298) = .211, p < .01$. The presence of an application security team had a statistically significant positive impact on developer self-efficacy. Hypothesis Three, stating that there is a positive correlation between the Presence of application security team and application security self-efficacy, was supported.

5.4. Hypothesis Four

The distribution of SE scores by the number of application security activities participants performed is shown in Table 7. Participants who performed nine application security activities had the highest SE score of 4.09. SE scores tended to be lowest for participants that performed more than nine application security activities. Application security did not correlate with the number of security roles performed ($r_s(298) = -0.027, p > .05$). Performing more security activities did not increase reported application security self-efficacy. Hypothesis Four stating that there is a positive correlation between the number of application security activities developers performed and application security self-efficacy was unsupported.

Table 7. SE scores per application security activities

Count of Security Activities	N	Mean SE Score
0	6	3.86
1	18	3.97
2	33	3.61
3	55	3.97
4	37	3.76
5	19	4.02
6	10	3.62
7	8	3.78
8	6	3.74
9	3	4.09
10	2	2.23
11	1	2.53
12	2	3.73

5.5. Hypothesis Five

The distribution of SE scores by participants' highest education level is shown in Table 8. Participants' SE scores increased with education level. The Vocational college group was the exception, with SE scores lower than the High school group. Participants with postgraduate degrees had the highest SE score of 4.05. Spearman's Rho calculated for the relationship between education level and software developers' application security self-efficacy was $r_s(298) = .316$, $p < .001$. Higher education levels had a statistically significant positive impact on developer self-efficacy. Hypothesis five, stating that there is a positive correlation between Software developers' education level and application security self-efficacy, was supported.

Table 8. SE scores per education level

Education Level	N	Mean
Elementary school	1	3
Middle school	2	3.4
High school	10	3.67
Vocational College	23	3.36
University	98	3.8
Postgraduate	66	4.05

5.6. Summary of Hypotheses Test

Application security self-efficacy correlated positively with AW (r_s(298) = . 542, $p < .001$), presence of security team (r_s(298) = . 211, $p < .01$), and Education level (was r_s(298) = .316, $p < .001$). Hypotheses One, Three, and Five positing a positive relationship between application security awareness, the presence of an application security management team, and education level, respectively, were therefore supported. Application security did not correlate with TR (r_s(298) = -0.121, $p > .05$) and NA (r_s(298) = -0.027, $p > .05$). Hypotheses Two and Four hypothesizing a positive correlation between application security self-efficacy and Application security training, and the number of application security activities performed were unsupported. Table 9 summarizes the hypotheses testing results.

Table 9. Summary of hypotheses testing

Hypothesis	Result	Decision
H1: There is a positive correlation between Application security Awareness and Application security self-efficacy.	Positive correlation (r_s(298) = . 542, $p < .001$).	supported
H2: There is a positive correlation between Application security training and Application security self-efficacy.	No correlation (r_s(298) = -0.121, $p > .05$).	unsupported
H3: There is a positive correlation between the Presence of an application security team and Application security self-efficacy.	Positively correlated (r_s(298) = . 211, $p < .01$).	supported
H4: There is a positive correlation between the number of application security activities developers performed and Application security self-efficacy.	No correlation (r_s(298) = -0.027, $p > .05$).	unsupported
H5: There is a positive correlation between Software developers' education level and Application security self-efficacy was supported.	Positively correlated (r_s(298) = . 316, $p < .001$).	Supported

6. DISCUSSION

Application security Awareness correlated significantly and positively with Application security self-efficacy. Higher levels of application security awareness imply greater knowledge and a positive attitude toward application security. Software developers with greater application security awareness will, therefore, have a greater tendency to value, prioritize and implement application security features.

The absence of a correlation between application security training and application security self-efficacy is an unexpected result. Application security developers who engage in application security training should naturally show greater application security self-efficacy through improved awareness and performance accomplishments. One reason for the unexpected result may be that software developers find application security training complex or irrelevant. Software developers may also feel that application security is not their responsibility. Other reasons may be the amotivations mentioned by [45], such as induced passivity, exclusion from security responsibilities, and personal philosophical resistance. The reason why application security training did not positively affect software developers' perceived application security self-efficacy should be further investigated.

The positive effect of the presence of an application security team on software developers' application security self-efficacy was an expected result. One of the amotivations for software developers adopting security practices is the perception of a lack of security competency [45]. The presence of application security managers, architects, and engineers increases the perception of application security competency. Their activities provide vicarious experience and social persuasions, essential sources of self-efficacy [13]. The presence of an application security team simplifies application security activities and guides software developers, increasing their application security self-efficacy.

Application security did not correlate with the number of application security roles performed. The lack of a positive relationship between application security self-efficacy and the number of application security activities performed is an unexpected result. Application security varies, and most are not carried out in the development phase of the SSDLC, which is the SSDLC phase software developers are most active. Examining Table 7 shows that developers who engaged in nine application security activities across the SSDLC had the highest SE score of 4.09, after which participation in more application security activities drastically reduced the SE score. However, the group with the highest SE score consisted of only three software developers. This could imply that the group may have consisted of application security engineers that also performed software development tasks. Another observation was that the most populous groups performed between two and four application security activities, with the majority performing three tasks (n = 55). But these most populous groups had mean SE scores between 3.61 and 3.97. The accumulation of software developers in the group that performed fewer application security activities might explain the negative and statistically insignificant correlation between the number of application security roles performed and SE. The results suggest that software developers' participation across multiple application security activities should be limited.

7. LIMITATIONS

The study has several limitations. A relatively small sample size of Software developers was surveyed, which is not fully representative of the population of software developers in the United States. A third-party audience service, Pollfish, was used to recruit the study population. Therefore, the study relied on participant information given to Pollfish and screening questions to target the study's population. The study was limited to a population of software developers based in the United States. Therefore, generalization should be limited to the United States.

8. IMPLICATIONS

The study has several implications. The study reveals the factors that impact developers' application security self-efficacy. This is vital to software quality assurance managers and product and information security managers of small and medium enterprises that develop software. The study's results showed that improving the application security awareness of software developers through proper awareness training will increase their application security self-efficacy. Application security awareness training should be included in general information security awareness training provided to software developers.

The study's results suggest that software developers should receive general application security training. However, this training should emphasize the importance of application security and the roles software developers play in application security. To prevent loss of motivation and improve application security self-efficacy, the training should be relevant to the tasks software developers perform practically. Another suggestion is that software developers routinely participate in other application security activities to reinforce their application security self-efficacy. However, given the lack of a positive correlation between the number of application securities developers perform, and their application security self-efficacy, the breadth of non-development or testing stage application security activities software developers should participate in should be limited to no more than five activities.

The study also showed that both developer education levels and the presence of security teams have a positive effect on developer application security self-efficacy. Well-educated software developers are, therefore, more likely to readily accept, understand and implement application security activities and features. Organizations that produce software should have a highly skilled application security team in place to improve software developer application security self-efficacy and the security of their developed software.

The literature review provided additional inferential information on improving application security self-efficacy by exploiting the sources of self-efficacy. Software developers should be immersed in an environment with a strong security culture and the required application security staff, management structure, and tools. Application security activities should also be visible to software developers. This environment is necessary to enable the existence of all four self-efficacy sources. Performance accomplishments can be helped by training challenges and including software developers in guided application security tasks. More challenging tasks should be assigned after each accomplished difficulty level. Vicarious experiences are enabled by peer observation of application security engineers, architects, and fellow software developers performing application security tasks. Application security engineers, architects, and champions should provide verbal persuasion to software developers performing application security tasks. Finally, adverse emotional arousal can be avoided by monitoring the efficacy levels of software developers to ensure they are not given application security tasks much higher than their capabilities.

9. Conclusion

Software must be released secured. A secure software development lifecycle must be practiced to ensure the security of released software. Practicing the SSDLC requires staff skilled in application security and a management structure. In some organizations, software security is treated as an afterthought and product add-on rather than a core feature. Some organizations cannot afford the complexity and cost of practicing the SSDLC. Software developers are saddled with the responsibility for software security in such situations. Software developers must have good application security awareness, training, and self-efficacy. Self-efficacy is important because it is a predictor of security policy compliance, personal enthusiasm and perseverance to complete security tasks, and intention to adopt new security tools and practices. This study examined the factors that influence software developers' application security self-efficacy. Application security awareness, the presence of the application security team, and education level are all positively correlated with software developers' application security self-efficacy. Security training and performing multiple application securities did not correlate with software developers' application security self-efficacy. The study has several practical implications, which were discussed.

Future research could look into additional factors, such as knowledge inertia and developers' experiences. The reasons for the lack of correlation between software developers' self-efficacy and security training and the number of security activities performed should also be qualitatively investigated. The results of this study may also be generalized by conducting the study across other countries and geographical regions.

REFERENCES

[1] E. Venson, R. Alfayez, M. M. F. Gomes, R. M. C. Figueiredo, and B. Boehm, “The impact of software security practices on development effort: an initial survey,” in 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Sep. 2019, pp. 1–12, doi: 10.1109/ESEM.2019.8870153.

[2] U.S. Department of Homeland Security, “Security in the softwarelifecycle: Making software development processes—and software produced by them—more secure. DRAFT Version 1.2. ,” 2006, Accessed: Jan. 28, 2023. [Online]. Available: http://www.cert.org/books/secureswe/SecuritySL.pdf.

[3] R. Fujdiak et al., “Managing the secure software development,” in 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Jun. 2019, pp. 1–4, doi: 10.1109/NTMS.2019.8763845.

[4] J. Ransome and A. Misra, Core Software Security. Auerbach Publications, 2018.

[5] H. Al-Matouq, S. Mahmood, M. Alshayeb, and M. Niazi, “A maturity model for secure software design: A multivocal study,” IEEE Access, vol. 8, pp. 215758–215776, 2020, doi: 10.1109/ACCESS.2020.3040220.

[6] W. C. Umeugo, “Secure software development lifecycle: A case for adoption in software SMEs,” International Journal of Advanced Research in Computer Science, Feb. 2023.

[7] H.-S. Rhee, C. Kim, and Y. U. Ryu, “Self-efficacy in information security: Its influence on end users’ information security practice behavior,” Computers & Security, vol. 28, no. 8, pp. 816–826, Nov. 2009, doi: 10.1016/j.cose.2009.05.008.

[8] B.-Y. Ng, A. Kankanhalli, and Y. (Calvin) Xu, “Studying users’ computer security behavior: A health belief perspective,” Decis. Support Syst., vol. 46, no. 4, pp. 815–825, Mar. 2009, doi: 10.1016/j.dss.2008.11.010.

[9] B. Y. Ng and Y. Xu, “Studying users’ computer security behavior using the Health Belief Model.,” PACIS 2007 Proceedings, 2007.

[10] R. W. Rogers, “Cognitive and psychological processes in fear appeals and attitude change: A revised theory of protection motivation.,” Social psychophysiology: A sourcebook, pp. 153–176, 1983.

[11] T. E. Gasiba, U. Lechner, M. Pinto-Albuquerque, and D. M. Fernandez, “Awareness of Secure Coding Guidelines in the Industry-A first data analysis.,” 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), p. 345, 2020.

[12] Z. A. Maher, H. Shaikh, M. S. Khan, A. Arbaaeen, and A. Shah, “Factors Affecting Secure Software Development Practices Among Developers - An Investigation,” in 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Nov. 2018, pp. 1–6, doi: 10.1109/ICETAS.2018.8629168.

[13] S. H. Appelbaum and A. Hare, “Self- efficacy as a mediator of goal setting and performance,” Journal of Managerial Psych, vol. 11, no. 3, pp. 33–47, May 1996, doi: 10.1108/02683949610113584.

[14] A. Bandura, “Social cognitive theory of self-regulation,” Organ. Behav. Hum. Decis. Process., vol. 50, no. 2, pp. 248–287, Dec. 1991, doi: 10.1016/0749-5978(91)90022-L.

[15] M. Kara and T. Aşti, “Effect of education on self-efficacy of Turkish patients with chronic obstructive pulmonary disease.,” Patient Educ. Couns., vol. 55, no. 1, pp. 114–120, Oct. 2004, doi: 10.1016/j.pec.2003.08.006.

[16] A. Bandura, “Self-efficacy: Toward a unifying theory of behavioral change.,” Psychol. Rev., vol. 84, no. 2, pp. 191–215, 1977, doi: 10.1037/0033-295X.84.2.191.

[17] M. Bong and E. M. Skaalvik, :“{unav),” Springer Science and Business Media LLC, 2003, doi: 10.1023/a:1021302408382.

[18] R. Wood and A. Bandura, “Social cognitive theory of organizational management,” Academy of Management Review, vol. 14, no. 3, pp. 361–384, Jul. 1989, doi: 10.5465/amr.1989.4279067.

[19] V. Hooper and C. Blunt, “Factors influencing the information security behaviour of IT employees,” Behav. Inf. Technol., pp. 1–13, May 2019, doi: 10.1080/0144929X.2019.1623322.

[20] Bulgurcu, Cavusoglu, and Benbasat, “Information Security Policy Compliance: An Empirical Study of Rationality-Based Beliefs and Information Security Awareness,” MIS Quarterly, vol. 34, no. 3, p. 523, 2010, doi: 10.2307/25750690.

[21] G. White, T. Ekin, and L. Visinescu, “Analysis of protective behavior and security incidents for home computers,” Journal of Computer Information Systems, vol. 57, no. 4, pp. 353–363, Oct. 2017, doi: 10.1080/08874417.2016.1232991.

[22] N. A. G. Arachchilage and S. Love, “Security awareness of computer users: A phishing threat avoidance perspective,” Comput. Human Behav., vol. 38, pp. 304–312, Sep. 2014, doi: 10.1016/j.chb.2014.05.046.

[23] N. Humaidi, V. Balakrishnan, and M. Shahrom, “Exploring user’s compliance behavior towards Health Information System security policies based on extended Health Belief Model,” in 2014 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Dec. 2014, pp. 30–35, doi: 10.1109/IC3e.2014.7081237.

[24] M. A. Albashrawi, L. Turner, and S. Balasubramanian, “Adoption of mobile ERP in educational environment,” International Journal of Enterprise Information Systems, vol. 16, no. 4, pp. 184–200, Oct. 2020, doi: 10.4018/IJEIS.2020100109.

[25] Md. A. Islam, M. A. Khan, T. Ramayah, and M. M. Hossain, “The Adoption of Mobile Commerce Service among Employed Mobile Phone Users in Bangladesh: Self-efficacy as a Moderator,” IBR, vol. 4, no. 2, Mar. 2011, doi: 10.5539/ibr.v4n2p80.

[26] M. Umarji and C. Seaman, “Predicting acceptance of Software Process Improvement,” in Proceedings of the 2005 workshop on Human and social factors of software engineering - HSSE ’05, New York, New York, USA, May 2005, pp. 1–6, doi: 10.1145/1083106.1083121.

[27] M. A. Hameed and N. A. G. Arachchilage, “The role of self-efficacy on the adoption of information systems security innovations: a meta-analysis assessment,” Pers. Ubiquitous Comput., vol. 25, no. 5, pp. 911–925, Oct. 2021, doi: 10.1007/s00779-021-01560-1.

[28] L. Mannila, L.-Å. Nordén, and A. Pears, “Digital Competence, Teacher Self-Efficacy and Training Needs,” in Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER ’18, New York, New York, USA, Aug. 2018, pp. 78–85, doi: 10.1145/3230977.3230993.

[29] P. J. Ambrose, “Metacognition and software developer competency: construct development and empirical validation.,” Issues in Information Systems, vol. 8, no. 2, pp. 273–9, 2007.

[30] R. N. Anantharaman, Rajeswari K. S., A. Angusamy, and J. Kuppusamy, “Role of Self-Efficacy and Collective Efficacy as Moderators of Occupational Stress Among Software Development Professionals,” in Social issues in the workplace: breakthroughs in research and practice, I. R. Management Association, Ed. IGI Global, 2018, pp. 854–868.

[31] I. M. Y. Woon and A. Kankanhalli, “Investigation of IS professionals’ intention to practise secure development of applications,” Int. J. Hum. Comput. Stud., vol. 65, no. 1, pp. 29–41, Jan. 2007, doi: 10.1016/j.ijhcs.2006.08.003.

[32] J. Witschey, O. Zielinska, A. Welk, E. Murphy-Hill, C. Mayhorn, and T. Zimmermann, “Quantifying developers’ adoption of security tools,” in Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2015, New York, New York, USA, Aug. 2015, pp. 260–271, doi: 10.1145/2786805.2786816.

[33] M. Deschene, “Embracing security in all phases of the software development life cycle: A Delphi study,” Undergraduate thesis, 2016.

[34] A. Senarath and N. A. G. Arachchilage, “Why developers cannot embed privacy into software systems? An empirical investigation,” in Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 - EASE’18, New York, New York, USA, Jun. 2018, pp. 211–216, doi: 10.1145/3210459.3210484.

[35] D. Votipka, D. Abrokwa, and M. L. Mazurek, “Building and Validating a Scale for Secure Software Development Self-Efficacy,” in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, Apr. 2020, pp. 1–20, doi: 10.1145/3313831.3376754.

[36] J. Zhen, K. Dong, Z. Xie, and L. Chen, “Factors influencing employees’ information security awareness in the telework environment,” Electronics, vol. 11, no. 21, p. 3458, Oct. 2022, doi: 10.3390/electronics11213458.

[37] K. Parsons, A. McCormac, M. Butavicius, M. Pattinson, and C. Jerram, “Determining employee awareness using the Human Aspects of Information Security Questionnaire (HAIS-Q),” Computers & Security, vol. 42, pp. 165–176, May 2014, doi: 10.1016/j.cose.2013.12.003.

[38] J. Ryan, “Information security awareness: an evaluation among business students with regard to computer self-efficacy and personal innovation.,” AMCIS 2007 Proceedings, 2007.

[39] G. Torkzadeh and T. P. Van Dyke, “Effects of training on Internet self-efficacy and computer user attitudes,” Comput. Human Behav., vol. 18, no. 5, pp. 479–494, Sep. 2002, doi: 10.1016/S0747-5632(02)00010-9.

[40] W. He et al., “Improving employees’ intellectual capacity for cybersecurity through evidence-based malware training,” JIC, vol. 21, no. 2, pp. 203–213, Nov. 2019, doi: 10.1108/JIC-05-2019-0112.

[41] C. W. Yoo, I. Hur, and J. Goo, “Workgroup collective efficacy to information security management: manifestation of its antecedents and empirical examination,” Inf. Syst. Front., Jan. 2023, doi: 10.1007/s10796-022-10367-1.

[42] R. D. Hays, T. Hayashi, and A. L. Stewart, “A Five-Item Measure of Socially Desirable Response Set,” Educ. Psychol. Meas., vol. 49, no. 3, pp. 629–636, Sep. 1989, doi: 10.1177/001316448904900315.

[43] Y. Salem, M. Moreb, and K. S. Rabayah, “Evaluation of Information Security Awareness among Palestinian Learners,” in 2021 International Conference on Information Technology (ICIT), Jul. 2021, pp. 21–26, doi: 10.1109/ICIT52682.2021.9491639.

[44] C. P. Dancey and J. Reidy, “Statistics without maths for psychology,” Statistics without maths for psychology, 2007.

[45] H. Assal and S. Chiasson, “Motivations and amotivations for software security.,” SOUPS Workshop on Security Information Workers (WSIW). USENIX Association, p. 1, 2018.