A Computer Security Experience Scale for 4th-Year Software Engineers

Jiajia Zhang
Department of Electrical &
Computer Engineering
University of Alberta
Edmonton, AB, Canada
jiajia3@ece.ualberta.ca

Scott Dick (*)
Department of Electrical &
Computer Engineering
University of Alberta
Edmonton, AB, Canada
dick@ece.ualberta.ca

Nelson Durdle
Department of Electrical &
Computer Engineering
University of Alberta
Edmonton, AB, Canada
durdle@ece.ualberta.ca

Abstract

The University of Alberta ECE department offers a computer security course for 4th-year Computer Engineering students. In order to focus our teaching efforts, we wish to measure the student's previous experience with computer security. However, we discovered that there are no appropriate "computer experience" scales in the literature. Thus an instrument is required that is capable of detecting differences in skill level among our students. We have developed just such an instrument, and piloted it in Fall 2009. Exploratory factor analysis of the pilot study indicates that a two-factor model fits this data best; we interpret the factors as representing the constructs "work experience" and "previous training." Our methodology should be appropriate for other 4th-year computer science courses.

1. Introduction

Prior computer experience is a known predictor of how students will generally perform in a computer-intensive course [1]. Clearly, however, the goal of education is not merely to translate prior experience into a grade; thus, instructors have a need to understand their students' level of computer experience, so that their pedagogy can be adjusted to better serve the students in the class. 4th-year computer engineering classes are no exception; anecdotally, we are all aware that some students (due to hobby interests or work experience) are more familiar with our course topics than others. These more experienced students can be a valuable resource; they can help stimulate debate in the classroom, and bring additional perspectives that are helpful to the less-experienced students. Thus, identifying the degree of computer experience is an important tool for faculty teaching in a computer engineering department.

The Electrical & Computer Engineering Dept. at the University of Alberta offers a Computer Engineering program with three degree streams: traditional Computer Engineering, Software Engineering, and a Nanoscale Systems stream. In these programs, CMPE 420 Reliable and Secure

Systems Design is required for the software stream and is an elective in the others. The course focuses on computer system *dependability*, in particular fault-tolerance and security. Students in this class can be described as very highly experienced computer users – and this is where our efforts to measure "computer experience" hit a roadblock. Existing measurement scales were design for the general public, and are not appropriate for our population. Specifically, the scales would likely all saturate at the "most experienced" response on every item for every student. Thus, if we are to measure different levels of computer experience for our students, we need a new measurement scale that is sensitive enough to recognize differences among these highly-trained individuals.

We have developed a "computer security experience" scale targeted at 4th-year students in computer science or engineering. The scale items were developed by considering security-relevant activities that computer hobbyists, or students with work experience, have likely participated in. We piloted this study in our Fall 2009 section of CMPE 420. We performed an exploratory factor analysis of the results, and we find that a two-factor solution is a good fit to our results. We interpret these factors as the constructs "work experience" and "previous training."

This paper is organized as follows. In section 2 we review previous work on the construct of computer experience, and its relationship with self-efficacy. Essential background about survey design and exploratory factor analysis are presented in section 3. Our methodology is presented in Section 4, and our empirical results in Section 5. We discuss our interpretation of these results in Section 6, and offer a summary and discussion of future work in Section 7.

2. Literature Review

2.1 Computer Experience

Computer experience is usually defined as "the degree to which a person understands how to use a computer. That is, an experienced computer user understands enough about computers in order to use them, more or less independent of specific software packages,

reasons for use and computer hardware features” [2]. Computer experience cannot only be described as the amount of computer use, but must also include the nature and quality of that use [31]. In recent work, computer experience is divided into two different dimensions, objective computer experience (OCE) and subjective computer experience (SCE), which are measured using quantitative and qualitative approaches [3]. Computer experience is also associated with computer anxiety, attitudes, and self-efficacy. The relationship between computer experience and self-efficacy will be reviewed in section 2.2.

Objective Computer Experience (OCE) is defined as “the totality of externally observable, direct and/or indirect human-computer interactions which transpire across time” [3]. OCE in its simple form only focus on the amount of computer use [4]. Jones & Clark propose a 3-factor definition of OCE, including the amount of computer use (in time), opportunities to use computers (i.e. at home or at school) and the diversity of computer experience (computer software packages or applications) [5]. These three factors measure direct OCE, while the construct “sources of information” measures “indirect” OCE (vicarious experiences) [3]. The four components provide a unified foundation for the various items and constructs identified in current researches.

Subjective computer experience (SCE) is defined as a “private psychological state reflecting the thoughts and feelings a person ascribes to some previous or existing computer event” [3]. It is a state of being consciously aware of one’s own private thoughts, feelings and actions at any given moment [6]. Referring to SCE as a private psychological state, B. Smith implies that SCE is a latent process which individuals possess but it cannot be observed directly. SCE incorporates both cognitive and affective aspects. These cognitive aspects include perceived competency, perceived usefulness and control [7]. The affective components of SCE include perceived enjoyment, anxiety, and relaxation [7].

Quantitative measures focus on externally observable or objective properties of computer experience [3]: courses completed, computer ownership, familiarity with various software packages; the frequency and duration of use; and what objective differences in their magnitude can be measured across time [8]. These quantitative measures appear to provide a valuable approach for investigating the subsequent effects of initial compute experience [9]. However, these measures do not consider the individual’s “inner life.”

Qualitative measures assess the individuals’ private thoughts and/or feelings ascribed to some previous

computing experience [3]. As a means of better understanding the dynamics of human-computer interaction, several researches adopted the qualitative assessment and examined different key variables or relationships under or affected computer experience. For example, Weil et al. examined the respondents’ retrospective perceptions and feelings [10]; Igbaria, Iivari and, Maragahh investigated subjective factors facilitating computer use [11]; Hall and Cooper examined how computer experience influences the personal-tool attribution made to the computers [12].

2.2 Self-Efficacy

Self-efficacy is the individual’s belief about their ability to accomplish a behaviour. It is one of the core aspects of Bandura’s Social Cognitive theory [17]. People with low self-efficacy for a task are less likely to attempt and perform that task [20], while high self-efficacy disposes people to perform more challenging tasks and activities creatively [17], leading to greater success. Computer self-efficacy (CSE) is derived from self-efficacy and refers to individuals’ judgment of their capabilities to use computers in diverse situations [18]. It’s generally agreed that an individual’s perceived self-efficacy has a positive relationship with IT use, and computer self-efficacy is important to successful IT deployments in organizations [19].

Self-efficacy has a positive relationship with computer experience [21]. Self-efficacy and prior computer experience significantly predict and contribute to the creativity of the subjects [17]. Some research proposed that individuals with prior computer experience are more likely to evidence higher levels of CSE than individuals without such experience [20]. However, Bandura observed that students who perceive themselves as supremely self-efficacious in a leaning task feel little need to invest much preparatory effort in it [22]. It is thus possible that students who have more prior computer experience may believe computer engineering courses are easy, and thus expend little effort toward learning.

3. Essential Background

3.1 Survey Design

In general, surveys are developed from a theoretical model that specifies a set of constructs, the relationships between them, and the outcome they influence. Items are then developed to measure the constructs. Items in the survey should be relevant, concise, unambiguous, and avoid double negatives. Each item should only reflect one clear thought. Surveys are then assessed for face validity (the item appears to measure what it was intended to) by one or more judges. The survey is then piloted; assuming the

conceptual model is well-established, items for each construct should be uni-dimensional; those violating this property will most likely be discarded. In a survey where the theoretical model itself (or its constructs) is the object of study, this will not necessarily be so [29]. Reliability and validity are key properties for any survey instrument. Reliability refers to the degree to which observed scores obtained from a questionnaire are systematically related to some underlying true scores [29]. Low reliability decreases the observed correlation between two variables. Validity is the extent to which inferences from scores are meaningful.

3.2 Exploratory Factor Analysis

There are two classical techniques used for reducing the dimensionality of a data set: principal components analysis (PCA), and exploratory factor analysis (EFA) [30]. When using PCA, a set of manifest variables or items are transformed into new and fewer uncorrelated variables (principal components), each representing a dimension in the data [30]. In EFA, instead of transforming the data into new variables, the original variables are the indicators of underlying dimensions (factors). The properties of models of psychological phenomenon can be explored and empirically tested using EFA. The two techniques will usually give nearly the same results. Although EFA has a weak tendency to find a solution in fewer dimensions than PCA, each technique has its own strengths. When the goal is to summarize a number of correlating variables in a few new variables with the smallest possible loss of information, PCA is appropriate. When the goal is to explain the correlations in a data set as a result of a few underlying factors, EFA is the best choice [30].

4. Methodology

4.1 Educational Context

CMPE 420 is a three-month lecture-oriented course that introduces students to the principles of fault tolerance and computer systems security. The course prerequisite is CMPE 300, Introduction to Software Engineering, which is required for all three computer engineering degree streams; thus, students in the course can be assumed to have some level of experience with software development and engineering. However, the software-stream students will have had a number of additional courses in software engineering and computer science between CMPE 300 and CMPE 420 (and CMPE 420 is required for this stream). The traditional and Nanoscale Systems stream students may have had little additional instruction in software engineering. For all students, this is their first formal exposure to computer security, and they are thus novice learners.

In Fall 2008, we introduced a course blog as a major graded component of the course. The objective of this blog is to serve as a “community of practice” for practicing what has been called the “security mindset;” a critical perspective on the security of the student’s environment (including the physical, social and technological milieu) and how built systems may be subverted to an attacker’s advantage. Students are required to post one “thoughtful” contribution to the blog each week; this could be a comment or a post, but it must demonstrate higher-order critical thinking (i.e. an identifiable analytic, evaluative or synthesis component, in Bloom’s terminology). What we have anecdotally observed is that, while the instructor plays the role of both facilitator and initiator for the community in the early part of the semester, students (particularly the ones with more apparent experience in computing) tend to take up the initiator role as the semester goes on, allowing the instructor to concentrate on the facilitator role. Feedback on the blog has been very positive, and thus we began planning a formal evaluation of the blog.

The outcome that we sought to measure was a construct we call “computer security self-efficacy.” As with other self-efficacy studies, we are interested in how our pedagogy leads to changes in self-efficacy over the course of the semester, implying a pretest-posttest design to measure this change. Following the existing literature, our conceptual framework for the pretest assumes that *computer security self-efficacy* is influenced by *computer security experience*. However, there is to the best of our knowledge no existing scale for computer experience experience, nor any computer experience scale that could be appropriately modified for this context. We thus developed our own scale for this construct, and modified an existing scale for the self-efficacy construct. These two scales were combined in a survey instrument, and administered to the CMPE 420 students in Fall 2009.

4.2. Survey Instrument and Administration

The pretest instruments consists of 14 items in total; 5 items measure previous *computer security experience*, while the other nine examine *computer security self-efficacy*. The items for computer security experience were constructed by considering security-relevant tasks that an advanced computer user might have undertaken at work or at home:

- Setting up a computer network
- Installing & configuring router-based firewalls
- Administering web and email servers
- Working experience as a systems administrator
- Website creation (particularly dynamic websites)
- Software application development

In addition, many students working within or outside the computer industry will have received some level of security training from their employers; cash drawer handling, store opening and closing procedures, and restricted-access areas are all aspects of security, and those experiences can also assist the student. After considering these points, we designed five items to measure what we believe are the underlying concepts:

1. *What is the largest computer network (including both computers and switching devices such as routers) you have been responsible for designing or administering, at home or professionally?*
2. *How many hours of computer-security training have you had (from a professional instructor or at work)?*
3. *What is the most complex website you have ever had primary responsibility for designing?*
4. *How many hours of security training (Other than computer security) have you had from a professional instructor or at work?*
5. *What is the most complex software system you have had lead responsibility for designing and implementing in a profession setting?*

The items were designed as multiple choice questions with five possible responses, with each choice representing roughly an “order-of-magnitude” increase (in terms of time or complexity) over the previous one.

CMPE 420 is a fourth year course; thus, due to the reduced IT degree enrolments of three years ago, the class is relatively small. In Fall 2009, 18 students registered the course in total, most of which are males; only four students are female. They are in the same age group, which is 20 to 25 years old. 17 students completed all the questions in the survey, and one student hand in an uncompleted answer sheet. As students in a course are a particularly vulnerable population, U of Alberta procedures require that the instructor not be present; surveys & consents are administered, collected and held by a neutral 3rd-party faculty member, and only released once course grades are posted. To facilitate the pretest-posttest design, students were privately given a randomly-generated 6-digit number by the 3rd party. A single list linking the student’s identity to this random number was retained by the 3rd party, allowing that number to be given back to the student for the posttest. After the posttest was completed, this list was destroyed, ensuring confidentiality.

5. Results

The mean and standard deviation of each *computer security experience* item (converted to a score between 1 and 5) are presented in Table 1. We can generally say that the means are low, hovering around the second-lowest response; however, the standard deviations are high. In a population of this size, this

probably means just a few respondents had significantly more experience than the majority. The correlations between items are presented in Table 2.

Given the relatively small population for this survey, we first test to see if our sample is sufficient to proceed with any analysis. We examine the Kaiser-Meyer-Olkin (KMO) sampling-adequacy statistic and Bartlett’s test of sphericity for this purpose. The KMO statistic is 0.615, indicating a reasonable degree of common variance among the responses. Bartlett’s test rejects the null hypothesis of an identity correlation matrix with p-value <0.001, indicating the presence of important correlations between items. Reliability for these items is good, as we measure a Cronbach’s Alpha of 0.808. Thus, proceeding with a factor analysis seems reasonable; however, we will be conservative in forming our conclusions due to the small sample size.

Table 1. Variable statistics

Variable	Mean	S.Dev
LargestNetwork (1)	1.94	.429
ComputerSecurityTraining(2)	2.00	1.500
ComplexWebsite(3)	2.12	.928
OtherSecurityTrainingTime(4)	1.65	1.057
ComplexSoftware(5)	1.71	1.213

Table 2. Variable correlations

Variable	(1)	(2)	(3)	(4)	(5)
(1)	1.00				
(2)	.194	1.00			
(3)	.176	.764	1.00		
(4)	.227	.591	.555	1.00	
(5)	.445	.481	.755	.45	1.00

We next proceed with EFA, using the PCA technique to examine the latent factors in the *computer security experience* scale. Table 3 presents the eigenvalues of the covariance matrix, and the percentage of total variance explained by each component. A common rule of thumb is to retain only factors with an eigenvalue greater than 1.0; we believe that the first two components should be kept, as 0.966 is extremely close. A scree test (see Figure 1) bears this out. Table 4 presents the factor loadings for the five items after varimax rotation.

Table 3. Component/factor matrix

Component	Eigenvalue	% of var.	Cum. %
1	2.948	58.954	58.954
2	0.966	19.329	78.283
3	0.578	11.564	89.846
4	0.397	7.938	97.784
5	0.111	2.216	100.0

Figure 1: Scree plot for Computer Security Experience items

Table 4. Rotated component/factor matrix (varimax)

Variable	PCA Component	
	1	2
(1)	.067	.963
(2)	.881	.053
(3)	.915	.153
(4)	.761	.132
(5)	.666	.551

6 Discussion

The results of Section 5 indicate that the computer security experience scale is best represented by a two-factor solution. Items 1 through 4 load quite unambiguously on a single factor; loadings on the primary factor are all greater than 0.7, while all cross-loadings are less than 0.2. Item 5 cross-loads heavily (both greater than 0.5). We believe that these two factors can be interpreted as flows: items 2, 3, and 4 together form a factor we will call *Training/Education*, while item 1 represents the factor *Practice*.

Items 2 and 4 explicitly ask about the level of security-related training a student has had, and their interpretation is obvious. The reason why we believe that item 3 reasonably belongs to this group has to do with the particular program of study at the U of Alberta. In one of the previous courses, students are required to design a simple interactive webpage (i.e. something on the order of CGI scripting) as a graded exercise. Observing the statistics of item 3, the average response was response #2 (student has designed a simple interactive webpage). We thus believe that, for a large number of our students, creating a complex webpage is part of their syllabus of study, rather than work or hobby experience.

Item 1, on the other hand, has no parallel in the course syllabus. the average response is again close to 2 (student designed/administered a network of 1-5

nodes, connecting to one access point). There is no course in the syllabus that asks students to perform this task, and thus this question probably reflects a home network (it seems too small for a professional environment). We interpret this factor as *Practice* because such hobby use will be an almost purely constructivist, trial-and-error learning experience, in contrast to the more formal experiences reflected in the *Training/Education* factor.

Item 5 asked students specifically about the largest system they had built *as professionals*. The mean for this item was the lowest among all items at 1.71 (response 1 is “no professional experience,” response 2 is “less than 500 lines of code”). The standard deviation, however, was relatively high at 1.213, indicating a small number of students had some degree of outside experience. This would seem to be why this item cross-loads on the *Practice* factor; professional work experiences are, after all, characterized by a very goal-oriented environment. Item 5 does seem to be important, in spite of its heavy cross-loading. If this item is dropped, Cronbach’s Alpha drops from 0.808 to 0.751; still acceptable but much closer to the usual cut-off of 0.7. The corrected item-total correlation for Item 5 is moderately strong at 0.657. It seems likely that the observed cross-loading is an artifact of the small sample size for this survey.

7 Summary and Future Work

In the study we developed an instrument with five items to investigate prior computer security experience for students in a fourth-year computer security course. Exploratory factor analysis of the pilot study indicates a two-factor model fits this data best; we interpret the factors as representing the constructs *Training/Coursework* and *Practice*. Our methodology should be appropriate for developing experience scales other 4th-year computer science courses, which also share this unique population of very highly experienced computer users.

The principal limitation of this research is that the data sample is small, due to the well-known reduced IT enrolments of 3 years ago; this course has previously had as many as 54 students registered. We intend to repeat this survey in Fall 2010 and Fall 2011, to gather additional evidence on the performance of this scale and its interpretation. In further research, we will use the pretest and posttest results to evaluate the effectiveness of our course blog in improving computer security self-efficacy.

References

- [1] M. Huggard, and C.M. Goldrick, "Computer Experience-Enhancing Engineering Education", 9th International Conference on Engineering Education, San Juan, PR, 2006.
- [2] D. Potosky, and P. Bobko, "The Computer Understanding and Experience Scale: a self-report measure of computer experience", *Computers in Human Behavior*, 1998, 14 (2), pp. 337-348.
- [3] B. Smith, P. Caputi, N. Crittenden, R. Jayasuriya and P. Rawstorne., "A review of the construct of computer experience", *Computers in Human Behavior*, 1999, 15(2), pp. 227-242.
- [4] K.J. Garland, J. M. Noyesb., "Computer experience: a poor predictor of computer attitudes", *Computers in Human Behavior*, 2004, 20 (6), pp. 823-840.
- [5] T. Jones, and V.A. Clarke, "Diversity as a determinant of attitudes: a possible explanation of the apparent advantage of single-sex settings", *Journal of Educational Computing Research*, 1995, 12(1), pp. 51-64.
- [6] A. S. Reber, *Dictionary of psychology*, Penguin Group, London, 1985.
- [7] M. Igbaria, and A. Chakrabarti, "Computer Anxiety and Attitudes towards Microcomputer Use", *Behaviour and Information Technology*, 1990, 9(3), pp. 229-241.
- [8] G.M. Breakwell, S. Hammond, C.R. Fife-Schaw (Eds.), *Research methods in psychology*, Sage Publications, London, 1995.
- [9] J. Todman, and E. Monaghan, "Qualitative differences in computer experience, computer anxiety, and students use of computers: a path model", *Computers in Human Behaviour*, 1994, 10(4), pp. 529-539.
- [10] M.M. Weil, L.D. Rosen, and S.E. Wugalter, "The Etiology of Computerphobia", *Computers in Human Behaviour*, 1990, 6, pp. 361-379.
- [11] M. Igbaria, J. Iivari, and H. Maragahh, "Why Do Individuals Use Computer Technology: A Finnish case study", *Information and Management*, 1995, 29(5), pp. 227-23.
- [12] J. Hall, and J. Cooper, "Gender, experience and attributions to computers", *Journal of Education and Computer Research*, 1991, 7(1), pp. 51-60.
- [13] S.L. Chua, D.-T. Chen, and A.F.L. Wong, "Computer anxiety and its correlates: a meta-analysis", *Computers in Human Behavior*, 1999, 15(5), pp. 609-623.
- [14] M. M. Maurer, "Computer Anxiety Correlates and What They Tell Us: A Literature Review", *Computers in Human Behavior*, 1994, 10(3), pp. 369-376.
- [15] N. Bozionelos, "Computer anxiety: relationship with computer experience and prevalence", *Computers in Human Behavior*, 2001, 17(2), pp. 213-224.
- [16] V. Mcinerney, D.M. Mcinerney, and K.E. Sinclair, "Student Teachers, Computer Anxiety and Computer Experience", *Educational Computing Research*, 1994, 11(1), pp. 27-50.
- [17] A. Tella, and C.O. Ayeni, "The Impact of Self-Efficacy and Prior Computer Experience on the Creativity of New Librarians in Selected Universities Libraries in Southwest Nigeria", *Library Philosophy and Practice*, 2006, 8 (2).
- [18] D.R. Compeau, and C.A. Higgins, "Computer Self-efficacy: Development of a Measure and Initial Test", *MIS Quarterly*, 1995, 19 (2), pp. 189-211.
- [19] D. M. Ball, *An Empirical Investigation of the Contribution of Computer Self-Efficacy, Computer Anxiety, and Instructors' Experience with the Use of Technology to Their Intention to Use Emerging Educational Technology in Traditional Classrooms*, ProQuest Information and Learning Company, Nova Southeastern University, 2008.
- [20] R. Karsten, and R.M. Roth, "Computer Self-Efficacy: A Practical Indicator of Student Computer Competency in Introductory IS Courses", *Informing Science*, University of 1998, 1(3), pp. 61-68.
- [21] E. Doyle, I. Stamouli, and M. Huggard, "Computer Anxiety, Self-Efficacy, Computer Experience: An investigation throughout a Computer Science degree", 35th ASEE/IEEE Frontiers in Education Conference, Indianapolis, IN, October 19 - 22, 2005.
- [22] A. Bandura, "Self-Efficacy Mechanism in Human Agency", *American Psychologist*, 1982, 37(2), pp. 122-147.
- [23] K.J. Garland, and J.M. Noyes, "Computer attitude scales: How relevant today", *Computers in Human Behavior*, Elsevier Science Publishers B. V. Amsterdam, the Netherlands, March 2008, 24(2), pp. 563-575.
- [24] B.H. Loyd, and C. Gressard, "Reliability and Factorial Validity of Computer Attitude Scales", *Educational and Psychological Measurement*, 1984, 20, pp. 501-505.
- [25] K.J. Garland, and J.M. Noyes, "Computer Experience: a poor predictor of computer attitudes", *Computers in Human Behavior*, 2004, 20, pp. 823-840.
- [26] B. Yang, and D. Lester, "Liaw's measures of attitudes toward computers and the Internet: A supportive comment", *Computers in Human Behavior*, 2003, 19, pp. 649-651.
- [27] F. Yaghmaie, R. Jayasuriya, and P. Rawstorne, "Computer experience and computer attitude: a model to predict the use of computerised information systems", *Computers in Human Behavior*, 1998, 19, pp. 649-651.
- [28] W. Arthur, and Jr.E. Olson, "Computer Attitudes, Computer Experience, and Their correlates: An Investigation of Path Linkages", *Teaching of Psychology*, February 1991, 18(1), pp. 51-54.
- [29] B.E. Hayes, *Measuring customer satisfaction: survey design, use, and statistical analysis methods*, ASQ Quality Press, Milwaukee Wisconsin, 1998.
- [30] N.J. Blunch, *Introduction to structural equation modelling using SPSS and AMOS*, SAGE, Los Angeles; London, 2008.
- [31] J.J. Beckers, and H.G. Schmidt., "Computer experience and computer anxiety", *Computers in Human Behavior*, 2003, 19(6), pp. 785-797.
- [32] B. Schneier, "Inside the Twisted Mind of the Security Professional," *Wired*, Mar. 20, 2008.