120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013

Paper ID #5925

Measuring Computing Self-Efficacy

Hannah Kolar, Arizona State University

Hannah Kolar is currently an undergraduate at Arizona State University. She is a rising senior working toward a Bachelor's of Science in Engineering from the College of Technology and Innovation with a focus on Mechanical Engineering and a secondary focus in Aeronautical Engineering. She has been working on engineering education research with Dr. Adam Carberry since November 2011.

Dr. Adam R Carberry, Arizona State University

Adam R. Carberry, Ph.D., is an Assistant Professor at Arizona State University in the College of Technology & Innovation's Department of Engineering. He earned a B.S. in Materials Science Engineering from Alfred University, and received his M.S. and Ph.D., both from Tufts University, in Chemistry and Engineering Education respectively. Dr. Carberry was previously an employee of the Tufts' Center for Engineering Education & Outreach and manager of the Student Teacher Outreach Mentorship Program (STOMP).

Prof. Ashish Amresh, Arizona State University

Ashish Amresh is an Assistant Professor in the College of Technology and Innovation and is leading the Computer Gaming curriculum initiatives at Arizona State University, where he founded the Computer Gaming Certificate and the Camp Game summer program. In the past he worked for Ronin Entertainment as Graphics Software Engineer known for Star Wars: Force Commander and Bruce Lee Quest of the Dragon. He is the author of Unreal Game Development, a popular book used for teaching game development skills for high school students. He has given numerous talks on using games in the class room for enhancing math and science learning. He is currently researching on how to improve learning of math word problems using games.

Creating and Validating a Computing Self-Efficacy Tool

Abstract

This study was based around the creation of a tool to measure students computing self-efficacy. The tool was an eight-question survey that was validated using content and criterion-related validity. Content validity was conducted to make sure that the questions related to each other and related to the subject of computing self-efficacy. Criterion-related validity allowed us to validate that our tool could test people with different levels of computing skills based on previous experience. The study allowed us to further validate our tool as well as analyze the computing self-efficacy of 270 students in science, technology, engineering, and mathematics (STEM) majors.

Introduction

Universities play a key role in creating future innovations and providing students with the tools and abilities to create the future. Computer-based innovations play a particularly prominent role because of how engrained they have become in many aspects of industry and our lives. It is important to have people who can create, maintain and fix computers and computer software. Unfortunately, high dropout rates in computing majors are far too common.

In this study we examine the effectiveness of a new instrument to measure computing self-efficacy. Such a tool can provide a analysis of an important factor that has been tied to student dropout in STEM majors. Our study explores three research questions that were present throughout the study:

1) What is the computing self-efficacy of university students pursuing a STEM related degree?
2) How do you accurately measure computer self-efficacy?
3) Do past educational experiences with computing improve computing self-efficacy?

We hypothesized that past experience in computing skills would lead to a higher self-efficacy in a student, which would provide us with criteria to use in validating our instrument.

Background

Self-efficacy is a component of Social Cognitive Theory.[1] According to Bandura, self-efficacy is "the belief in one's capabilities to organize and execute the courses of action required to manage prospective situations".[1] This definition implies that self-efficacy is a person's awareness of their ability to accomplish a goal.

The theory states that there are four sources of information that shape self-efficacy – listed in decreasing influence and importance:

1) Performance accomplishments or mastery experiences,
2) Vicarious experiences,
3) Verbal or social persuasions,
4) Physiological states.

The combined effect of these four sources determines someone's self-efficacy toward a given task.[2] A number of studies of self-efficacy in engineering have been conducted.[3-12] Of particular importance is Quade's study, which developed an instrument to measure problem-solving, computer troubleshooting, career encouragement, satisfaction with college major, career exploration, and course anxiety.[6] The analysis revealed gains in problem solving and computer troubleshooting as well as gender differences for computer-science majors and minors after completing a required course. Differences in self-efficacy between genders is not surprising as studies have shown self-efficacy in engineering to be a huge factor in the retention of women.[12-14] These studies have shown that women are less confident in their abilities in engineering design, which causes many women to drop out of these programs.

Our study varies from past literature, specifically Quade's study, by focusing on syllabi from freshman computing courses and expert interviews rather than creating questions that were related directly to Bandura's antecedents for self-efficacy. This study also focused solely on computing skills instead of adding it to their self-efficacy towards career encouragement and exploration.

Research methods

Sample
Our sample included two hundred and seventy students who attended the Intel Ultimate Experience Internship (IUEI) located in Phoenix, AZ in the summer of 2012. Students were all in STEM-based majors (engineering, computer science, or other science-based fields). Participants ranged from freshman to seniors in college (freshman 14%; sophomore 59%; junior 23%; senior 4%) with 97 students being females and 173 being males. Within our sample 58% claimed to have previous computing experience.

Instrument Design
The developed instrument included demographic questions (gender, major, and computing experience) and eight items pertaining to computing self-efficacy. The items utilized a 100-point range Likert scale with 10-unit intervals; 0 being "cannot do at all" and 100 being "highly certain can do". This scale was used based on previous research identifying familiarity of students with being scored on a scale of 100 in most of their courses.[15] The eight questions were designed to measure self-efficacy toward computing as a whole (item 1) and seven different specific computing skills:

1) solve a computing task
2) develop computer software
3) write coding scripts using text-based normal code

4) use visual drag and drop code
5) relate coding scripts to visual code
6) troubleshoot software problems/errors
7) wire parts of a hardware system together
8) troubleshoot hardware problems

These questions allowed us to measure mean scores of self-efficacy for individual computing tasks as well as an overall mean score for computing self-efficacy of students in STEM majors.

The instrument was created and validated using both content and criterion-related validity. Content validity concerns the extent to which a measurement adequately samples a specific domain represented in an instrument.[16-18] Content validity came by way of two resources. First, we researched past studies on the field of computing and computing-related self-efficacy. We used freshman computing and fused computing-engineering project class syllabi to create the items for our instrument. Second, we conducted fifteen-minute in-person interviews with computing professors at a large southwest university to make sure our questions were relevant and related to the field of computing.

Criterion-related validity concerns the ability of an instrument to predict an externally related criterion.[16-18] In this study, criterion related validity was conducted by using our measure of previous programming coursework by asking the following question: "Have you ever taken a class or course to learn how to program?" The assumption was that respondents with previous programming experience would have a higher initial computing self-efficacy.

Data Collection and Analysis
The survey was emailed to all of the students attending the IUEI through the course instructor. The survey was administered through an online surveying tool called Survey Monkey©. Each respondent was required to answer all of the questions to complete the survey. The results were tested for validity and reliability using factor analysis and internal consistency (Cronbach's α). A series of t-tests analyses were then conducted to assess differences across groups.

Results

Our study was conducted to examine three areas of interest: 1) validating our computing self-efficacy tool for our sample, 2) measuring computing self-efficacy of students in STEM majors, and 3) comparing self-efficacy differences between students who had past programming class experience and those who hadn't.

We first tested the validity and reliability of our tool. Our factor analysis testing revealed one main factor, which we subsequently named *computing self-efficacy*. The factor weightings ranged between 0.730 and 0.923 (Table I). To ensure our factor was indeed computing self-efficacy, we then compared our emergent factor to our first broad question (Item 1) to do a correlation analysis ($r = 0.711$). Our high correlation confirmed that our broad first item related to the sum of the other questions at a significance level of $p \leq 0.01$. For reliability we used inter-item reliability to ensure a value greater than 0.700.[19] A Cronbach's α value of 0.929 resulted giving an above average reliability.

Table I: Factor weightings of each question compared to the first broad item

Item	**Factor Weights**
2. develop computer software	0.874
3. write coding scripts using text-based 0rmal code	0.886
4. use visual drag and drop code	0.854
5. relate coding scripts to visual code	0.923
6. troubleshoot software problems/errors	0.840
7. wire parts of a hardware system together	0.730
8. troubleshoot hardware problems	0.780

We then measured the computing self-efficacy of students in STEM majors. Mean (M) scores for the respondents' were calculated for each of the questions (Table II). We found that the computing self-efficacy of STEM majors was lower than 50% (M = 42.12 out of 100). We also saw that the software related questions seemed to have the lowest mean scores.

Table II: Mean sum of computing self-efficacy

Item	**M**
1. solve a computing task	53.40
2. develop computer software	31.49
3. write coding scripts using text-based 0rmal code	33.53
4. use visual drag and drop code	36.25
5. relate coding scripts to visual code	31.12
6. troubleshoot software problems/errors	41.37
7. wire parts of a hardware system together	52.83
8. troubleshoot hardware problems	49.67

Thirdly, we compared the difference of computing self-efficacy of the students who had past programming experience and the students who hadn't. We did a t-test analysis to compare the two groups. We found that for all of the items there was a significant difference ($p \leq 0.005$) between groups.

Discussion and Implications

This study led us to three distinct findings. First we further validated a tool created to measure computing self-efficacy for our sample population of students. This allowed us to be able to use this tool to its full potential for this study in order to present evidence for future use of this instrument. Professors can use this tool to see what skills are specifically causing low self-efficacy, so they are able to improve upon future class interventions.

We also confirmed that computing self-efficacy was generally low for students in a variety of STEM majors. We believe this is a major factor when considering dropout rates for these majors at the university level. This information has the potential to be very interesting for universities who are trying to fix their dropout rates. We believe that awareness of self-efficacy will help instructors address confidence issues that in turn should impact dropout

Lastly, we provided evidence that our pre-study hypothesis was correct; students who had previous computing experience had a higher self-efficacy toward computing skills. This is likely because they had seen it before and knew what to expect. This may appear obvious, but the possibility did exist that those having previous experience may have been more likely to rate themselves lower due to how difficult it can be to learn many computing skills. This knowledge was important in the validation of our tool and its ability to measure self-efficacy correctly because it allowed us to use experience as a criterion to base group comparisons and expected levels.

Conclusion

This study allowed us to come to many interesting and helpful conclusions for disciplines that embed computing within their curriculum. The tested computing self-efficacy tool can now be used for more in depth studies looking at other background information that might be affecting computing self-efficacy of STEM students. Knowing the present self-efficacy of these students provides a valuable measure that can impact classroom pedagogy and STEM curricula. This information for universities may be a key to decrease dropout rates, especially for females and first-year students. It is also important to know that past experience in computing improves computing self-efficacy. Universities should obtain this data from students to identify when material should be added to a course that allows all students to be brought up to speed on their computing skills before launching into STEM-based majors. Future investigations utilizing this tool will attempt to understand the impact of computing self-efficacy on student performance, i.e. time to complete a task and academic achievement.

Bibliography

1. Bandura, A. 1995. “Self-Efficacy in Changing Societies.” Cambridge University Press.
2. Bandura, A. 1997. “Self-efficacy: The exercise of control.” New York, NY: W.H. Freeman and Company.
3. Baker, D., Krause S., and Purzer S. Y. 2008. “Developing an instrument to measure tinkering and technical self-efficacy in engineering.” American Society for Engineering Education Annual Conference and Exposition. Pittsburgh, PA.
4. Carberry, A.R., Lee, H-S., and Ohland, M.W. 2010. “Measuring engineering design self-efficacy.” Journal of Engineering Education, Vol. 99(1), pp. 71-79.
5. Compeau, D.R., & Higgins, C.A. 1995. “Computer Self-Efficacy: Development of a Measure and Initial Test.” MIS Quarterly, Vol. 19 (2), pp. 189-211.

6. Quade, A. 2003, “Development and validation of a computer science self-efficacy scale for CS0 courses and the group analysis of CS0 student self-efficacy.” Proceedings of the International Conference on Information Technology: Computers and Communications, Las Vegas, NV.
7. Richardson, A.L. 2008. “Tinkering interactions on freshman engineering design teams.” Proceedings of the American Society for Engineering Education Annual Conference and Exposition. Pittsburgh, PA.
8. Ponton, M.; Edmister, J.; Ukeiley, L. & Seiner, J. (2001), 'Understanding the role of self-efficacy in engineering education', Journal of Engineering Education - Washington 90(2), 247--252
9. Reed, K. (2012), 'Assessment of Student’s Confidence of Learned Knowledge', American Society for Engineering Education (ASEE) 2012 Annual Conference .
10. Fantz, T.; Siller, T. & Demiranda, M. (2011), 'Pre-Collegiate Factors Influencing the Self-Efficacy of Engineering Students', Journal of Engineering Education 100(3), 604--623.
11. Hutchison, M.; Follman, D.; Sumpter, M. & Bodner, G. (2006), 'Factors influencing the self-efficacy beliefs of first-year engineering students', Journal of Engineering Education - Washington 95(1), 39.
12. Kilgore, D., Atman, C.J., Yasuhara, K., Barker, T.J. and Morozov, A. 2007. “Considering Context: A Study of First-Year Engineering Students.” Journal of Engineering Education, Vol. 96 (4), pp. 321-334.
13. Marra, R.,M. Rodgers, K.A., Shen, D., and Bogue, B. 2009. “Women Engineering Students and Self-Efficacy: A Multi-Year, Multi-Institution Study of Women Engineering Student Self-Efficacy”. Journal of Engineering Education, Vol. 98 (1), pp. 27-38.
14. Whitt, E.J., Pascarella, E.T., Nesheim, B.S.E., Marth, B.P., and Pierson, C.T. 2003. “Differences between women and men in objectively measured outcomes, and the factors that influence those outcomes, in the first three years of college.” Journal of College Student Development. Vol. 44 (5), pp. 587-610.
15. Pajares, F., J. Hartley, and G. Valiante. 2001. “Response format in writing self-efficacy assessment: Greater discrimination increases prediction.” Measurement and Evaluation in Counseling and Development. Vol. 33, p. 214-221.
16. Carminer and Zeller, 1979. Reliability and validity assessment. Thousand Oaks, CA: Sage Publications.
17. Messick, S. (1989). Validity. In Educational Measurement, edited by R. L. Linn , p. 13-103. Washington, D.C.: The American Council on Education and the National Council on Measurement in Education.
18. Moskal, B.M., J.A. Leydens, and M.J. Pavelich (2002). Validity, reliability and the assessment of engineering education. Journal of Engineering Education. 91 (3): 351-354.
19. George, D. and Mallery, P. (2003). SPSS for Windows step by step: A simple guide and reference. Boston, MA: Allyn and Bacon.