The Relationship between Engineering Students' Self-efficacy Beliefs and Their Experience Learning Computer Programming: A Sequential Explanatory Mixed-Methods Investigation

Ms. S. Zahra Atiq, Purdue University, West Lafayette

S. Zahra Atiq is a PhD student at the School of Engineering Education at Purdue University, West Lafayette. Her research interests include: computer science education specifically on teaching computer programming to undergraduates and how to improve their learning experiences. She is also interested in understanding student behaviors and performance in online learning environments specifically MOOCs.

WIP - The relationship between engineering students' self-efficacy beliefs and their experience learning computer programming: A sequential explanatory mixed-methods investigation

1. Introduction

Many engineering curricula require a computer programming course for first-year students. An introductory programming course teaches many concepts that may be difficult for students to learn[1-3]. Students' responses to conceptual difficulties are influenced by their self-efficacy beliefs, that is, by their appraisal of their ability to learn these concepts successfully. Students with higher self-efficacy beliefs are likely to persist in overcoming these difficulties to succeed in the course, and vice versa. As a result of interacting with difficult concepts, students' self-efficacy beliefs are also liable to change and evolve during the course. Changes in students' self-efficacy beliefs may impact their decision to select a certain engineering major or even persistence in the engineering. For instance, if taking a programming course decreases a student's self-efficacy beliefs, then that student may be demotivated enough to drop out of engineering.

Engineering education researchers have been studying the impact of students' self-efficacy beliefs on their academic achievement and persistence in engineering[4]. Studies on self-efficacy of engineering students have been conducted in the context of team discourse and student achievement[5], engineering design projects[6], and developing validated self-efficacy instruments for engineers[7]. Moreover, there is evidence in literature on measuring self-efficacy of engineering students in the context of programming[8-9]. Askar et al., examines factors related to self-efficacy for Java programming in first year engineering students. These factors include gender, computer experience, general computing skills, frequency of computer use, and family computer usage. Findings from this study confirm the link between students' self-efficacy beliefs and their choice of subject. It was also found that computer engineering students had higher self-efficacy beliefs compared to students from electronics or industrial engineering[8]. Another study found that the number of programming courses and the achievements of students in those courses predicts their programming self-efficacy[9]. Although, these studies examine engineering students' self-efficacy beliefs in programming courses, there is little evidence on how programming changes students' self-efficacy beliefs.

Understanding changes in students' self-efficacy beliefs can help programming educators design appropriate interventions to counter the demotivating factors related to self-efficacy. This understanding may also help engineering educators devise appropriate means to help students understand the significance of programming for engineering. This proposed study aims to investigate the relationship between programming self-efficacy beliefs of first-year engineering students and their experience taking an introductory programming course. The self-efficacy construct from Bandura's Social Cognitive Theory (SCT)[10] will be used as a theoretical framework to guide the different stages of the proposed study. We use a sequential explanatory mixed-methods design, in which a main quantitative phase uses a validated computer programming self-efficacy scale, followed by a qualitative phase, using semi-structured student interviews. During the integration phase, findings from both quantitative and qualitative phases

will be mixed to understand the phenomenon in a holistic manner. Finally, SCT will be used as a lens to analyze the combined findings from both phases of the study.

2. Research Question(s)

Mixed-methods research follows from a pragmatic perspective, hence the research questions guide and determine the entire process such as selection of research design, sample size, and data collection methods[11-13] The research questions for this study are:

1. The overarching research question is: "What is the relationship between engineering students' programming self-efficacy beliefs and their experience learning computer programming?"
2. The quantitative research question is, "Are there differences in students' programming self-efficacy beliefs after taking an introductory computer programming course?"
3. The tentative qualitative question is, "Why do we observe differences in students' programming self-efficacy beliefs as a result of taking an introductory programming course?"

3. Mixed-methods research design

The research design selected for this study is the mixed-methods sequential explanatory design[11]. This research design consists of two distinct phases: quantitative followed by qualitative. The rationale behind this research design is that the first quantitative phase (data and analysis) provides a general understanding of the problem. The qualitative data analysis further provides an in-depth and refined understanding of the problem because it explores participants' views in detail[11,14]. Finally, both quantitative and qualitative results will be interpreted and explained. These steps are illustrated in Figure 1 and discussed in subsequent sections.

Phases of the Study	Procedure	Product
Quantitative Data Collection	- Student demographic and background data - Computer Programming Self-Efficacy Scale (pre and post) - Midterm and final exam grades	- Numeric data (n = 296)
Quantitative Data Analysis	- Cleaning the data - hypothesis testing using two-tailed paired t-tests	- descriptive and inferential statistics
Connecting Quantitative and Qualitative Phases	- Refine the qualitative research question based on findings of the quantitative phase - Maximum variation sampling for participant recruitment - Development of qualitative data collection instruments	- Interview protocol - recruitment materials - consent forms
Qualitative Data Collection	- Participant recruitment (using recruitment emails and consent forms) - face-to-face student interviews (audio recorded)	- Interview transcripts (~ n = 10 – 15)
Qualitative Data Analysis	- thematic analysis of the qualitative data	- Code and themes as a result of the thematic analysis procedure
Integrating Quantitative and Qualitative Results	- Interpretation and explanation of findings from both the QUANT and qual phases	- Discussion - Implications - Future research - Visual display tying QUANT and qual results

Figure 1: Phases of the study, procedures used and end product of each step[14]

3.1. Phase 1 - Quantitative study

We will use stratified random sampling[15] to collect data from 12 institutions in the United States that require a computer programming course for undergraduate engineering students (2 baccalaureate institutions, 2 master's universities, and 8 doctoral universities). Since different types of institutions have different academic structures, and different types of academic support for the students, it is important to understand how students' programming self-efficacy beliefs change as a result of taking a programming course in different contexts. Such a multi-institution study will not only help understand student satisfaction and retention in engineering, but will also increase the generalizability of the study[16]. Moreover, we select those universities, which use Java to teach programming as Java is one of the most frequently used languages to teach programming to undergraduate engineering students[17]. The data from each institution will be collected so that it is representative of race, gender, and ethnicity of the students. Using power analysis for two-tailed pair t-tests with alpha = 0.01, power = 0.8, and small effect size[18], the desired sample size for the study is 296 after considering a 10% response rate, which means that the survey should be sent to a minimum of 2960 students across the 12 institutions to be able to collect the desired sample.

To investigate self-efficacy in a computer programming course, we will use the validated Computer Programming Self-Efficacy Scale (CPES), which was developed by Ramalingam and Wiedenbeck[19]. Since self-efficacy of a persons' abilities change depending on the domain, it must be measured by an instrument that is pertinent to that domain[10]. The 32 items of CPES is preceded by the words "I could". The strength of self-efficacy was measured by recording student responses on a 7-point Likert scale ranging from 1 to 7 (1 = "not confident at all", to 7 = "absolutely confident"). The scale was developed in the context of object-oriented programming in C++, but has also been adapted for Java[20]. For our study, we will use the Java adapted version of the scale. The adapted CPES will be administered as a pre- and post-survey, once during the first week of the semester and once during the last week of the semester. We want to see how students' self-efficacy beliefs change over time, hence, we want the two tests to be well-spaced in time so that students do not remember their responses on the pre-test. We will also collect student demographic and background information (e.g., SAT scores, prior programming experience) at the beginning of the semester, midterm grades, and final grades.

We will conduct hypothesis testing using two-tailed paired t-tests on the pre- and post- survey data. During this phase it would be useful to compare the results with the initial research question to determine the extent to which the research question was answered by the results. Moreover, these results can then also be checked for criterion-related validity[11] by relating them with the self-efficacy construct of Bandura's SCT.

3.2. Phase 2 - Mixing of Quantitative and Qualitative data

In this phase, the quantitative and qualitative strands of the study will be mixed. The quantitative findings from the previous phase will be used to inform the qualitative phase[11,14]. The first step in the mixing phase is to refine the qualitative research question based on the findings of the quantitative data analysis. The second step is to design and pilot the data collection protocol. In this case, we will develop a semi-structured interview protocol, which is the key artifact obtained as a result of mixing the quantitative and qualitative strands of the study. Based on the findings of the quantitative phase and by using Bandura's SCT, the interview protocol will be developed.

Recognizing that a pilot may help us uncover flaws and limitations of the interview protocol[21], one will be conducted and necessary revisions will be made based on it. Moreover, to ensure communicative validity[22], the interview protocol will also be discussed with colleagues and will be refined, based on their feedback.

3.3. Phase 3 - Qualitative study

During this phase, we will select two institutions of each type (six institutions in all) and recruit interview participants from the same group of students who will fill the pre-survey, post-survey, and will provide demographic and grade information. Maximum variation sampling will be used to select interview participants for this phase; this type of sampling not only documents diversity, but also helps identify common patterns across diversity[23]. Selected students will be sent an email inviting them for the interview. Students who volunteer to be interviewed will be asked to sign consent forms. Qualitative research does not rely on large sample sizes[11,23]. Hence, it is expected that at least 10 – 15 students will volunteer to be interviewed from each of the six selected institutions (60 to 90 students). Face-to-face interviews will then be conducted and audio recorded. Interview transcripts will be transcribed verbatim to facilitate subsequent data analysis[15].

Before the interview transcripts are analyzed, they will be sent to interview participants for member checking, to ensure communicative validity[23]. Once member checking is done, the transcripts will be analyzed using thematic analysis procedures[24,25]. Since the study is using Bandura's SCT, it is appropriate to use this framework to guide the data analysis procedures as well. In qualitative research, reliability can be seen as the usage of appropriate methods to ensure consistency in data collection and interpretation[26]. Keeping this point in view, the transcripts will be coded separately by two researchers to ensure inter-rater reliability.

3.4. Phase 4 - Integrating quantitative and qualitative results

The final phase of the mixed-methods study is to integrate the quantitative and qualitative strands of the study. In explanatory designs, the key question to ask during the integration phase is: In what ways do the qualitative data help to explain the quantitative results[11,23]? On the other hand, it is also important to ask how numbers explain the qualitative results[23]. Moreover, the SCT can be used here to make sense of the integrated findings. Finally, a display linking qualitative themes to quantitative findings may aid in understanding the results and in answering the overarching mixed-methods research question[11].

4. Conclusions

This paper discusses the details of a proposed sequential mixed-methods explanatory study, which aims to investigate the change in undergraduate first-year engineering students' self-efficacy beliefs after taking an introductory programming course. In this study, quantitative data will be collected first using a validated computer programming self-efficacy scale. The findings from the quantitative data will inform the design of the qualitative phase of the study. Qualitative data will be collected and analyzed. The quantitative and qualitative findings will be integrated to answer the overarching research question of the study and Bandura's SCT will be used as a lens to analyze the findings of the study.

References

1. Clancy, M. (2004). Misconceptions and attitudes that interfere with learning to program. *Computer Science Education Research*, 85–100.
2. Kaczmarczyk, L. C., Petrick, E. R., Philip, J., Geoffrey, E., & Herman, L. (2010). Identifying student misconceptions of programming. In *In Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 107–111).
3. Shinners-Kennedy, D., & Fincher, S. A. (2013). Identifying threshold concepts: from dead end to a new direction (pp. 9–18). Presented at the Proceedings of the ninth annual international ACM conference on International computing education research, ACM.
4. Loo, C. W., & Choy, J. L. F. (2013). Sources of Self-Efficacy Influencing Academic Performance of Engineering Students. *American Journal of Educational Research*, *1*(3), 86–92. https://doi.org/10.12691/education-1-3-4
5. Purzer, Şe. (2011). The Relationship Between Team Discourse, Self-Efficacy, and Individual Achievement: A Sequential Mixed-Methods Study. *Journal of Engineering Education*, *100*(4), 655–679. https://doi.org/10.1002/j.2168-9830.2011.tb00031.x
6. Booth, J.-M. J., Doyle, T. E., & Musson, D. M. (2013). Influence of learning preference on self-efficacy and performance in mixed-modality first-year engineering design. *Proceedings of the Canadian Engineering Education Association*, *0*(0). Retrieved from http://ojs.library.queensu.ca/index.php/PCEEA/article/view/4883
7. Mamaril, N. A., Usher, E. L., Li, C. R., Economy, D. R., & Kennedy, M. S. (2016). Measuring Undergraduate Students' Engineering Self-Efficacy: A Validation Study. *Journal of Engineering Education*, *105*(2), 366–395. https://doi.org/10.1002/jee.20121
8. Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *Turkish Online Journal of Educational Technology*, *8*(1), 26–32.
9. Jegede, P. O. (2009). Predictors Of Java Programming Self Efficacy Among Engineering Students In A Nigerian University.
10. Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191–215. https://doi.org/10.1037/0033-295X.84.2.191
11. Creswell, J. W., & Clark, V. L. P. (2011). *Designing and Conducting Mixed Methods Research*. SAGE.
12. Onwuegbuzie, A., & Leech, N. (2007). Sampling Designs in Qualitative Research: Making the Sampling Process More Public. *The Qualitative Report*, *12*(2), 238–254.
13. Tashakkori, A., & Teddlie, C. (2010). *SAGE Handbook of Mixed Methods in Social & Behavioral Research*. SAGE.
14. Ivankova, N. V., Creswell, J. W., & Stick, S. L. (2006). Using Mixed-Methods Sequential Explanatory Design: From Theory to Practice. *Field Methods*, *18*(1), 3–20. https://doi.org/10.1177/1525822X05282260
15. Creswell, J. W. (2011). *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research* (4 edition). Boston: Pearson.
16. Marra, R. M., Rodgers, K. A., Shen, D., & 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*, *98*(1), 27–38. https://doi.org/10.1002/j.2168-9830.2009.tb01003.x

17. Evans, F. (2015, January 20). Which of these top 20 programming languages should your school teach?
18. Cohen, J. (1992). A power primer. *Psychological Bulletin*, *112*(1), 155–159. https://doi.org/10.1037/0033-2909.112.1.155
19. Ramalingam, V., & Wiedenbeck, S. (1998). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research*, *19*(4), 367–381. https://doi.org/10.2190/C670-Y3C8-LTJ1-CT3P
20. Doyle, E., Stamouli, I., & Huggard, M. (2005). Computer Anxiety, Self-Efficacy, Computer Experience: An investigation throughout a Computer Science degree (pp. S2H–3–S2H–7). IEEE. https://doi.org/10.1109/FIE.2005.1612246
21. Turner, D. W. (2010). Qualitative Interview Design: A Practical Guide for Novice Investigators. *The Qualitative Report*, *15*(3).
22. Walther, J., Sochacka, N. W., & Kellam, N. N. (2013). Quality in Interpretive Engineering Education Research: Reflections on an Example Study. *Journal of Engineering Education*, *102*(4), 626–659. https://doi.org/10.1002/jee.20029
23. Patton, M. Q. (2014). *Qualitative Research & Evaluation Methods: Integrating Theory and Practice* (4 edition). Thousand Oaks, California: SAGE Publications, Inc.
24. Aronson, J. (1995). A Pragmatic View of Thematic Analysis. *The Qualitative Report*, *2*(1), 1–3.
25. Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa
26. Peterson. (1997). Interviews: An Introduction to Qualitative Research Interviewing - ProQuest. *Journal of Phenomenological Psychology*.