

‘Doing Good Matters’: The Role of Problem Context in Improving Programming Projects

Emergent Research Forum (ERF)

Lakshmi Iyer

Appalachian State University
iyerls@appstate.edu

Indika Dissanayake

University of North Carolina at
Greensboro
i_dissan@uncg.edu

Rudolph Bedeley

University of Massachusetts Amherst
rbedeley@isenberg.umass.edu

Abstract

Given the rapid advancement in information technology (IT), it is imperative to find ways to increase IT talent pipeline to meet future workforce demands. Specifically, developing programming skills has become one of the core component in any information systems (IS) curriculum. While prior studies have examined factors that impact students’ performance in programming such as personality traits, past academic performance, self-efficacy, and cognitive skills (Hostetler 1983, Ramanlingam and Wiedenbeck 1998, Porter et al. 2013), the topic continues to be investigated given that several students struggle in introductory programming course (Malik 2018). Students perceived programming to be taught in a very abstract manner making it difficult to relate to, and lack of ability to help people (Townsend et al. 2007). Hence, in this study, based on grounding from situated learning theory (Anderson et al. 1996), we wanted to examine if engaging students in meaningful context would make a difference in the programming course.

Thus, we explore the following research questions: a). Does involving students in programming projects with social context increase their task satisfaction, confidence in performance, intrinsic motivation, IT identity and the level of programming self-efficacy? How does the students’ perception about the project context impact these outcomes? b). Do teams work on programming projects with social context outperform teams work on problems with commercial context? Does project team’s gender diversity affect the above?

To answer the above questions, we use an experimental method to investigate the effect of problem context (social Vs. non-social) on students’ performance outcome such as task satisfaction, confidence in performance, and solution quality. Our findings have several implications for research and practice.

REFERENCES

- Anderson, J. R., Reder, L. M., & Simon, H. A. (1996). Situated learning and education. *Educational researcher*, 25(4), 5-11.
- Hostetler, T. R. (1983). Predicting student success in an introductory programming course. *ACM SIGCSE Bulletin*, 15(3), 40-43.
- Malik, S. I. (2018). Improvements in Introductory Programming Course: Action Research Insights and Outcomes. *Systemic Practice and Action Research*, 1-20
- Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013). Success in introductory programming: what works?. *Communications of the ACM*, 56(8), 34-36
- 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.
- Townsend, G. C., Menzel, S., & Siek, K. A. 2007. Leveling the CS1 playing field. Paper presented at the 38th SIGCSE technical symposium on Computer science education, Covington, KY.