Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #45855

BOARD # 310: WIP: Impact of Prior Programming Experience on Self-Efficacy Impacts of WebTA Autocritiquer

Mary Benjamin, Michigan Technological University

PhD Student in the Dept. of Civil, Environmental, & Geospatial Engineering at Michigan Technological university.

Dr. Michelle E Jarvie-Eggart P.E., Michigan Technological University

Dr. Jarvie-Eggart is a registered professional engineer with over a decade of experience as an environmental engineer. She is an Assistant Professor of Engineering Fundamentals at Michigan Technological University. Her research interests include technology adoption, problem based and service learning, and sustainability.

Dr. Leo C. Ureel II, Michigan Technological University

Leo C. Ureel II is an Assistant Professor in Computer Science and in Cognitive and Learning Sciences at Michigan Technological University. He has worked extensively in the field of educational software development. His research interests include intelligent learning environments, computer science education, and Artificial Intelligence

Dr. Jon Sticklen, Michigan Technological University

Jon Sticklen is an Associate Professor with the Engineering Fundamentals Department (EF) and Affiliated Faculty with the Department of Cognitive and Learning Sciences (CLS). He served as Chair of EF from 2014-2020, leading a successful effort to design a

NSF IUSE WIP: Impact of Prior Programming Experience on Self-Efficacy Impacts of WebTA Autocritiquer

Abstract

WebTA WebTA is an automated code critiquer that delivers real-time, formative feedback to first-year engineering students in flipped-class, active-learning environments. Initially developed for Java, WebTA has been extended to MATLAB as part of the IUSE-funded Rich, Immediate Critique of Antipatterns (RICA) project. It examines its impact on the computer programming self-efficacy of novice programmers. Within first-year engineering classes, students were asked to submit MATLAB code to WebTA for feedback, so that they might revise it prior to submission for grading. The tool provides instant feedback on syntax, logic, and style, enabling large cohorts of students to iteratively improve code support traditionally limited by instructional capacity.

Given the foundational role of programming in engineering curricula and the established link between self-efficacy and student persistence, this study examines how WebTA impacts programming self-efficacy among novice engineers. Specifically, we analyze baseline computer programming and engineering self-efficacy in relation to students prior programming experience and initial confidence levels. By introducing both intervention and control sections, this work aims to isolate the effects of automated feedback and inform inclusive strategies for enhancing programming competency and self-efficacy in early engineering education.

Introduction

Programming is a fundamental skill for engineering students, playing a critical role in their academic success and future careers. However, first-year engineering students often face challenges in developing programming self-efficacy, a domain-specific belief in their ability to succeed in programming tasks. Research highlights that low self-efficacy can hinder motivation, persistence, and engagement, particularly in STEM fields where programming is integral [1,2]. Gender disparities exacerbate these challenges, with women consistently reporting lower programming self-efficacy despite comparable performance to men [3].

Automated feedback tools, such as WebTA, provide real-time critique on student code, offering solutions to enhance programming self-efficacy [4]. WebTA, initially designed for Java, has been adapted for MATLAB under the Rich, Immediate Critique of Antipatterns (RICA) project [5]. By integrating WebTA into first-year engineering courses, this research will examine its impact on programming self-efficacy, with specific attention to the role of prior programming experience and gender differences.

Prior research has underscored the critical role of self-efficacy in supporting student persistence and success within engineering programs, while also drawing attention to persistent gender disparities in programming self-efficacy. Our prior literature review examined the influence of automated code critiquers on novice programmers self-efficacy development [6]. Subsequent work documented instructors' perceptions of implementing WebTA in MATLAB-based first-year engineering courses [7], and empirical findings from classroom deployment highlighted gender-specific impacts of WebTA on programming self-efficacy among first-year students [8]. Notably, while both men and women experienced gains in self-efficacy, women reported significantly greater post-intervention increases in confidence related to independent coding and debugging, suggesting that real-time, automated feedback may serve as an important support mechanism in mitigating gender-based confidence gaps.

However, our Spring 2024 study lacked a non-intervention control group, thereby limiting causal inferences regarding the tools effect. The current study addresses this limitation by incorporating a control condition, facilitating a more rigorous evaluation of WebTAs impact on student self-efficacy.

Methods

In Fall 2024, WebTA was integrated into the first semester engineering classroom ENG1101: Engineering Analysis and Problem Solving, where students were required to submit three MATLAB assignments through the tool. These assignments included tasks such as writing functions and implementing conditional statements. After receiving feedback from WebTA, students revised their code based on the feedback before submitting their final versions to the Learning Management System (Canvas). The intervention group used WebTA to receive automated feedback on their programming assignments, while the control group received standard feedback.

Participants were first-year engineering students enrolled in multiple offerings of ENG1101 at Michigan Tech University. A total of 809 students participated across various class sections. Demographics, including gender and prior programming experience, were collected during the first week of the semester via a survey. Of the respondents, 22.4% identified as female (n = 181), 74.3% as male (n = 600), 2.1% as non-binary (n = 17), and 1.2% preferred not to disclose their gender identity (n = 10), indicating that gender-inclusive options were available. Additionally, 27.4% of students (n = 222) reported having prior programming experience, while 72.6% (n =587) reported no prior exposure to programming.

Two validated scales were used to assess changes in both computer programming and engineering self-efficacy:

1. **Computer Programming Self-Efficacy Scale (CPSES):** Measures programming confidence across constructs such as independence, persistence, and complex task handling [9].
2. **Longitudinal Assessment of Engineering Self-Efficacy (LAESE):** Assesses confidence in engineering-related tasks, career expectations, and sense of belonging [10].

The instruments were administered as pre- and post-surveys to capture baseline and post-intervention self-efficacy data. The CPSES and LAESE surveys both used a 7-point Likert scale ranging from “not confident at all” to “absolutely confident.

Preliminary Results:

Preliminary data presented includes the mean pre-intervention score for the LAESE and CPSES for each class offering section of ENG1101, the introductory engineering course. Each class combines several recitation sections together, as presented below in Table 1.

Class Sections	Condition	Mean LAESE	Mean CPSES
L1-5	Intervention	5.75	3.61
L6-10	Intervention	5.81	3.52

L12-14	Intervention	5.72	3.63
L16-20	Intervention	5.65	3.50
L21-25	Non-Intervention	5.65	3.34
L26-30	Intervention	5.41	3.55
L33-35	Non-Intervention	5.73	3.56
L37-39	Intervention	5.54	3.52
L51-53	Non-Intervention	5.54	3.51
L54-56	Non-Intervention	5.72	3.36
Average across all classes	Both	5.65	3.52

Table 1. Pre-Intervention Self-Efficacy Scores

The pre-intervention data indicates that students have higher self-efficacy in engineering-related tasks, as measured by the LAESE (mean: 5.65), compared to programming tasks, as measured by the CPSES (mean: 3.51). This suggests that first year engineering students feel more confident in their engineering abilities than in their programming skills, highlighting a potential area for targeted interventions to bridge the gap in programming self-efficacy. As programming is a foundational skill for engineering programs required by ABET General Criteria for Baccalaureate Level Programs, Criterion 5. Curriculum, which requires both computer science and "modern engineering tools" [11], programming is integrated into first year engineering courses. For students with high CPSES, these novice programming courses may feel like a stepping stone. But for students with low CPSES, these introductory classes may become a gatekeeper.

Future Work

This research aims to inform scalable, inclusive teaching practices in engineering education, addressing self-efficacy disparities and fostering persistence among diverse student populations. The next phase of the study will measure changes in self-efficacy across both genders between pre- and post-surveys, and analyze gender-specific impacts of WebTA on programming self-efficacy. Qualitative interviews will be conducted to understand student experiences and perceptions of WebTA feedback.

Funding Acknowledgement

The work presented in this article was supported by the U.S. National Science Foundation (NSF) under Grant No. 2142309. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

1. Bandura, A. (1997). *Self-efficacy: The exercise of control.* W.H. Freeman and Company.
2. Kinnunen, P., & Simon, B. (2012). My program is ok – am I? Computing freshmen's experiences of doing programming assignments. *Computer Science Education, 22*(1), 1–28. https://doi.org/10.1080/08993408.2012.655091
3. Zeldin, A. L., & Pajares, F. (2000). Against the odds: Self-efficacy beliefs of women in STEM careers. *AERJ*, 37(1), 215–246. https://doi.org/10.3102/00028312037001215
4. Ureel, L. C., & Wallace, C. (2015). WebTA: Automated iterative critique of student programming assignments. *2015 IEEE Frontiers in Education Conference (FIE)*, 1–9. https://doi.org/10.1109/FIE.2015.7344200
5. Ureel II, L. C., Brown, L. E., Sticklen, J., Jarvie-Eggart, M., & Benjamin, M.* (2022). The RICA Project: Rich, Immediate Critique of Antipatterns in Student Code. CSEDM Workshop. https://doi.org/10.5281/zenodo.6983498
6. Benjamin, M.*, Jarvie-Eggart, M., Ureel, L., Brown, L., & Sticklen, J. (2023). Engaging Novice Programmers: A Literature Review of the Effect of Code Critiquers on Programming Self-efficacy. FIE. https://doi.org/10.1109/FIE58773.2023.10342975
7. Benjamin, M., Albrant, L., Jarvie-Eggart, M., Sticklen, J., Brown, L., & Ureel, L. (2024). Exploring Instructors Insights to a MATLAB Code Critiquer. FYEE. https://peer.asee.org/48581
8. Benjamin, M., Albrant, L., Jarvie-Eggart, M. E., Ureel, L. C., Brown, L. E., Sticklen, J., & Hamlin, A. (2025). Gendered Impacts of Code Critiquers on Self-Efficacy in First-Year Engineering Students. CoNECD. https://peer.asee.org/54094
9. Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale. J. Educational Computing Research, 19(4), 367381. https://doi.org/10.2190/4U0A-3WEG-R874-6PYD
10. Yoon, S. Y., & Sorby, S. A. (2020). Rescaling the Longitudinal Assessment of Engineering Self-Efficacy V3.0 for Undergraduate Engineering Students. Journal of Psychoeducational Assessment, 38(2), 209-221. https://doi-org.services.lib.mtu.edu/10.1177/0734282919830564
11. ABET (2023). Criteria for Accrediting Engineering Programs, 2024 – 2025. Accessed Jan 11, 2024: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2024-2025/