Project-Based Learning Improves Critical Thinking for Software Development Students

Cengiz Günay
Anca Doloc-Mihu
Thomas Gluick
cgunay@ggc.edu
adolocmihu@ggc.edu
tgluick@ggc.edu

Sch. of Science and Technology, Georgia Gwinnett College
Lawrenceville, Georgia

Catherine A. Moore

cmoore10@ggc.edu

School of Liberal Arts, Georgia Gwinnett College
Lawrenceville, Georgia

ABSTRACT

Here we share our experiences in applying project-based learning and its impact on critical thinking skills. In the software development (SD) track of information technology majors, upper-level SD courses are designed to combine students' accumulated programming knowledge for learning to develop software applications, reaching their ultimate goal. Usually, SD courses aim to maximize student success and achievement by training students with instructor-defined projects having known solutions. On one hand, students apply past knowledge and skills and improve problem-solving while realizing the project. On the other, students can become disconnected from, or bored with, the problems since students do not own the project. In addition, the projects do not present unknown challenges requiring to exercise critical thinking and creativity. As an alternative, we assign students novel real-world projects, proposed by external clients who direct the development of the respective software product. In-class simulation of real-world scenarios is known to be more effective for learning than lecturing in software development and engineering education [1]. We propose that project-based learning using a previously unsolved problem will improve student critical thinking and problem solving skills, two key STEM competencies important for software engineering. Furthermore, we expect a positive synergy between student success, as project goals are attained, and motivation to complete the project. We measured a significant increase of critical thinking skills in a senior-level undergraduate Software Development course. We assessed these skills using the CAT (Critical Thinking Assessment Test), a standard instrument developed by the Tennessee Tech University [2]. We administered pre- and post- CAT in two sections of the course in Fall 2018, at the beginning and end of the semester, respectively. We observed a statistically significant (one-tailed, paired t-test, $p < 0.05$) increase of 11% between pre- and post CAT scores (Fig. 1). Our preliminary results suggest that the project-based approach to teaching software development increases

general critical thinking skills. Student projects can be accessed at <https://soft-eng-practicum.github.io/>.

CCS CONCEPTS

• **Social and professional topics** → **Software engineering education**; **Student assessment**; *Information technology education*;
• **Applied computing** → **Education**; • **Software and its engineering** → **Designing software**; **Programming teams**; *Agile software development*; *Documentation*.

KEYWORDS

software development, critical thinking, CAT, project based learning, PBL

Figure 1: CAT scores of 16 students that took both pre- and post-tests.

ACKNOWLEDGMENTS

The authors would like to thank Judy Awong-Taylor, Clay Runck, Dave Pursell, Joshua Edwards, Tom Lily, GGC IRB committee (#17082), and SST staff for funding and support. Partially funded by USG-STEM EIP and NSF-IUSE 1623779 grants.

REFERENCES

- [1] Kirti Garg and Vasudeva Varma. 2007. A Study of the Effectiveness of Case Study approach in Software Engineering Education. In *20th Conference on Software Engineering Education & Training, CSEET'07*. IEEE.
- [2] Barry Stein, Ada Haynes, Michael Redding, Theresa Ennis, and Misty Cecil. 2007. Assessing Critical Thinking in STEM and Beyond. In *Innovations in E-learning, Instruction Technology, Assessment, and Engineering Education*. Springer, 79–82.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGITE '19, October 3–5, 2019, Tacoma, WA, USA
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6921-3/19/10.
<https://doi.org/10.1145/3349266.3351362>