ATTITUDES TOWARD PROGRAMMING FOR STUDENTS WITH DISABILITIES IN UNDERGRADUATE TECHNOLOGY COURSES THAT USE PAIR PROGRAMMING: A MATCHED SAMPLES APPROACH

Silvana Watson, PhD., Ling Li, PhD., Shana Pribesh, PhD., Li Xu, PhD.
Old Dominion University, Norfolk, VA

Introduction and Rationale for the Study

Students with Disabilities Drop Out of Computer Science Courses:
To succeed in the 21st century, students need to acquire skills that are critical to the workforce such as collaboration, social skills, and technology literacy (World Economic Forum, 2016). Individuals with disabilities (D) must develop the same skills as their peers without disabilities. Unfortunately, college students with disabilities often find computing courses frustrating and are more vulnerable to lower academic self-concept, academic challenges, and disability stigma (Kim & Kutscher, 2021). Although computing disciplines often provide good job opportunities, Students with D who enrolled in computing courses are especially at risk of falling behind and dropping out of introductory programming courses (Richman et al., 2014). To address the problem, we examined the use of pair programming, a collaborative approach to programming, as a pedagogic method to improve students with disabilities' attitudes toward programming in undergraduate computer courses. There is a need to study effective instructional approaches that can facilitate learning and improve the outcomes of students with D.

Pair Programming Holds Promise:
Pair programming, a collaborative form of programming, has been used in some college classrooms by computer science professors. Typically, a programmer acts as the driver who controls the keyboard and mouse and writes the code. Another programmer acts as the observer or navigator, and is responsible for reviewing the code, and, at the same time, preventing and identifying logical and syntactical errors in the code (Estácio & Prikladnicki, 2015). As pair programming offers many educational benefits, this study investigated how pair programming impacts students with disabilities.

Acknowledgments

This study was funded by the National Science Foundation's Division of Undergraduate Education (DUE). Award # 1712251.

Methods

Hypotheses:
1. There is a statistically significant improvement in the attitudes toward programming of students with disabilities who learn with the pair programming approach than those who learn with the conventional programming approach.
2. There is a statistically significant improvement in the learning experience for students with disabilities who learn with the pair programming approach than those without

Participants: Thirty-three undergraduate students with disabilities and 32 undergraduate students without disabilities enrolled in the study.

Design and Data Collection: *Pre-mid- and post-test surveys* to measure participants' programming experiences, programming self-efficacy, and attitude and motivation toward programming. To create a comparison, we located a student without disabilities matching in engagement, gender (15 F and 38 M), prior computing experience (*MwD*=5.39, *MnD*=5.37, *p*=.480), attitudes toward collaboration (MwD3.25 MnD = 340, p = .262), computer confidence (MwD =4.13, MnD= 4.09, p= .861), and attitude toward computing (MwD =4.27, MwD=4.18, p = .600).

Students were asked to rate a series of statements about programming at the middle and end of the course. Using a 5-point scale, ranging from strongly disagree (1) to strongly agree (5), students reacted to statements, such as, "Overall, I enjoy the learning experience in completing a programming exercise, I had the opportunity to think creatively in completing the programming exercises, I am satisfied with the level of effort the programming exercises required, and I enjoyed programming with my partner. " These statements were combined into an average scale variable ranging from 1 to 5. Hierarchical regression analysis was used to examine the relationship among the variables.

Results

Attitudes toward Pair Programming:
Only model 4, which explained 60.1% of the variance in attitudes toward programming, reached significance. It contained D status, pair programming engagement, gender, and baseline computing controls (F (7,57)=4.601, p<.001). Additionally, only attitudes toward collaboration predicted attitudes toward programming (t=2.745, p=.008). In summary, students with disabilities did not differ from those without learning disabilities in their programming attitudes at the middle or end of the course

Learning Experiences:
After the first interaction with pair programming (mid), the matched sample contained 57 subjects (31 with D, 26 without). None of the models reached significance which means that they were not suitable for interpretation. The results for the less rigorous, unmatched sample (31 D, 408 nonD) indicated that only the final, inclusive model reached significance (Model 3 – F(5,433)=2.603, p=.025). That model predicted midpoint pair programming experience with D status, gender, and other baseline computing-related variables. Preexisting computer interest was related to pair programming experience (b=.119, t=2.647, p=.008).

After the second interaction with pair programming (post), the matched sample contained 59 subjects (32 with D, 27 without). None of the models reached significance. The results for the less rigorous, unmatched sample (32 D, 420 nonD) indicated that only the final, inclusive model reached significance (Model 3 – F(5,446)=7.901, p<.001). Preexisting computer interest (b=.263, t=5.943, p<.001) and computer attitudes (b=-.105, t=-2.515, p=.012) were related to pair programming experience.

Conclusion

Students with D did not differ from those without D in their programming attitudes and learning experiences.