Pair Programming Efficacy and Implementation Strategies in Chinese High School IT Curriculum

Zhang Wuwen, Guan Yurong

Huanggang Normal University, Huanggang, Hubei, 438000, China

Abstract: In the past decade, governments worldwide *have* incorporated programming education in primary and secondary schools as a crucial initiative to cultivate technical talent and enhance international competitiveness. Against this backdrop, this paper examines the efficacy and implementation strategies of pair programming in Chinese high school information technology curricula. Pair programming is an effective learning approach that fosters computational thinking, communication and collaboration skills, confidence and self-efficacy, innovative thinking, and problem-solving abilities while simultaneously augmenting students' programming expertise and practical experience. To optimize the implementation of pair programming instruction, this paper proposes several recommendations, including defining students' pair programming roles, supplying essential programming tools and resources, judiciously allocating time, encouraging student sharing and interaction, emphasizing class cohesion, offering personalized guidance, and continually refining teaching methodologies.

1. Concept and Definition of Pair Programming

Pair Programming is a software development methodology involving two programmers collaboratively working on a single coding task[1]. One programmer, known as the "Driver," is responsible for writing code, while the other, the "Observer," reviews the code and provides feedback. Initially proposed by Kent Beck in his book "Extreme Programming Explained," Pair Programming is a component of Extreme Programming and is widely employed in other software development approaches. Its application extends beyond software development, increasingly adopted in educational settings. Since the UK first advocated teaching programming to elementary students in 2013, many countries, including China, have promoted programming activities in primary and secondary schools. Pair programming can help students better master programming skills and enhance learning outcomes[2].

According to Liebenberg's research, students working in pairs demonstrate greater improvement in programming knowledge and computational thinking compared to those working individually[3], and are more likely to express enjoyment in programming and persist in problem-solving when coding with peers[4]. From 2002 to 2003, McDowell and other researchers conducted a study evaluating students' task completion rates and scores, ultimately concluding that Pair Programming is an effective learning model; students learning through this method exhibit higher task completion

rates, superior exam scores, and heightened enthusiasm for learning [5]. These findings indicate that Pair Programming has a positive impact on students' learning and development. Therefore, in high school information technology courses, Pair Programming should be a prevalent teaching method to foster students' computational thinking, communication and collaboration skills, confidence and self-efficacy, as well as cultivating innovative thinking, problem-solving abilities, and enhancing their programming expertise and practical experience.

2. Learning Advantages of Pair Programming

Programming has significant learning advantages in high school information technology courses and offers practical application value for enhancing students' abilities in various aspects. Firstly, pair programming promotes the development of students' computational thinking and innovation abilities[6], enabling them to better master programming skills. Secondly, it strengthens communication and collaboration among students, allowing them to grow and progress together.

Furthermore, pair programming helps improve students' self-confidence and self-efficacy by sharing tasks, collaborating, and providing timely feedback and encouragement, which fosters a positive learning attitude. Additionally, it cultivates students' innovative thinking and problem-solving abilities, teaching them how to analyze problems and optimize solutions.

Lastly, pair programming enhances students' programming skills and practical experience, laying a solid foundation for their future learning and work. In summary, pair programming brings multiple advantages to high school information technology courses, contributing to the comprehensive development of students.

3. Pair Programming Teaching Process

3.1 Selecting Suitable Programming Tasks

Before initiating pair programming, teachers need to carefully select programming tasks to ensure they are appropriately challenging and pedagogically meaningful. Teachers can opt for relatively simple programming tasks to help students familiarize themselves with the programming environment and language, cultivating their interest in programming. Simultaneously, teachers can also choose more challenging programming tasks to stimulate students' learning enthusiasm and enhance their programming abilities and practical experience. For example, teachers can select programming tasks related to real-world issues relevant to the students' courses, allowing students to apply their knowledge and skills while solving problems. Teachers can also refer to excellent programming tasks, such as those from renowned programming education websites like Code.org, to better guide students in pair programming.

3.2 Designing Programming Tasks

When designing programming tasks, teachers should follow students' learning patterns and psychological characteristics, considering the task's difficulty and complexity to ensure its feasibility. According to the "Information Technology Curriculum Standards," programming tasks should have "appropriate difficulty and sufficient challenge" and be able to "exercise students' computational thinking and innovation capabilities." Furthermore, teachers can design programming tasks related to course content based on students' actual situations and learning progress, helping them better understand and grasp the knowledge. In summary, designing suitable programming tasks is one of the essential prerequisites for ensuring the effectiveness of pair programming instruction.

3.3 Explaining Programming Task Requirements

Before starting pair programming, teachers should explain the requirements and objectives of the programming task to students, ensuring they have a clear understanding of the task requirements and can successfully complete them. Teachers should clarify the task's difficulty, completion goals, required time, and resources, and provide guidance and support based on students' actual situations. Additionally, teachers need to provide students with necessary programming tools and resources, such as programming software, tutorials, and literature, to better complete programming tasks.

3.4 Performing Programming Tasks

While performing programming tasks, students need to work according to their role division, with the driver primarily responsible for writing code and the observer reviewing the code and providing timely feedback. Students should collaborate closely during the pair programming process, exchanging ideas and negotiating, jointly solving problems and identifying errors and defects to ensure high-quality task completion. Simultaneously, students need to apply their knowledge and skills flexibly, analyzing and understanding problem requirements, designing and implementing solutions to achieve programming task goals. Throughout this process, students should continuously debug and optimize their code to make it more standardized, concise, and efficient.

3.5 Providing Evaluation and Feedback

After completing pair programming, teachers need to evaluate and provide feedback on students' performance so they can understand their strengths, weaknesses, and areas for improvement. The evaluation should focus on students' programming abilities and teamwork skills, helping students better comprehend their advantages and shortcomings, thereby improving their skills. Concurrently, teachers should offer necessary guidance and encouragement to promote student growth and progress. These evaluations and feedback should be timely and accurate, enabling students to understand their situation and improve their abilities promptly.

3.6 Summarizing and Sharing Experiences

After pair programming ends, students should summarize the programming task and share their learning and work experiences. This helps students better understand and master programming knowledge and skills, and facilitates their communication and collaboration abilities. Students can summarize by reviewing task completion, analyzing the problem-solving process, and reflecting on their work experience, thus considering their learning and work situations and identifying areas for improvement. Simultaneously, students can share their learning and work experiences, exchanging ideas and discussing with peers, promoting communication and collaboration among students and expanding their perspectives.

The pair programming teaching process requires considering various factors, such as students' programming levels, interests, and personal traits. Teachers need to adjust and optimize the process based on students' actual situations

4. Pair programming teaching implementation recommendations

4.1 Define student roles

Clarify driver and observer roles for students, employing voluntary selection or teacher assignment,

implementing role rotation to enhance collaboration efficiency and comprehensive skills.

4.2 Supply programming tools and resources

Provide students with programming software, textbooks, online materials, etc., ensuring resources align with course objectives and student needs while guiding students towards familiarity with usage methods.

4.3 Rationally allocate time

Ensure sufficient time for students to complete programming tasks, taking into account academic workload, course progress, and mental and physical health, adjusting task difficulty and rest periods.

4.4 Encourage sharing and communication

Instructors should promote active student participation, experience sharing, organizing pair programming sharing sessions, and utilizing online platforms to share work and insights.

Focus on class construction: Strengthen class construction, fostering mutual trust and collaborative relationships among students, organizing interactive communication activities, and cultivating teamwork spirit and responsibility.

4.5 Offer personalized guidance

Tailor teaching to individual student characteristics and needs, paying attention to student feedback, adjusting teaching strategies, and encouraging proactive participation and autonomous learning.

4.6 Improve teaching methods

Continuously optimize teaching strategies, participate in professional development activities, engage in peer learning and communication, learn from successful cases, and periodically reflect on and adjust teaching plans.

Pair programming education can enhance students' computational thinking, communication and collaboration abilities, and innovative thinking in multiple aspects. By adhering to the above suggestions, teachers can guide students more effectively in pair programming, improving teaching outcomes and learning quality.

5. Conclusion and Prospects

Pair programming is an effective teaching method with numerous significant learning advantages. It fosters students' computational thinking, enhances communication and collaboration abilities, bolsters self-confidence and self-efficacy, cultivates innovative thinking and problem-solving skills, and improves programming expertise and practical experience. These benefits demonstrate the crucial practical application value and educational significance of pair programming in high school information technology courses. Moreover, to better implement pair programming instruction, teachers should select appropriate programming tasks for students based on the teaching process, provide detailed explanations of task requirements, supply necessary programming tools and resources, allocate time reasonably, and encourage sharing and communication. They should also focus on class construction, offer personalized guidance, and continuously refine teaching methods.

In the future, pair programming instruction can further integrate with modern educational technologies, employing novel teaching tools such as online teaching platforms and virtual

laboratories, making teaching more flexible and efficient. Simultaneously, pair programming instruction can be combined with artificial intelligence technologies, improving teaching outcomes and student learning experiences through AI-assisted instruction and adaptive learning. In summary, as an effective teaching method, pair programming can play an essential role in high school information technology courses. Teachers should emphasize teaching practice, enhance teaching quality and standards, and continuously improve teaching methods to promote students' comprehensive development and growth.

References

[1] Laurie Williams, Robert Kessler. Translated by Yang Tao and Yang Xiaoyun. Pair Programming Techniques. Beijing: Mechanical Industry Press, 2004.

[2] Campe Denner, Green Torres. Pair programming in middle school: variations in interactions and behaviors. Computer Science Education, 2020, 30(1), 22-46.

[3] Denner J, Werner L, Campe S, et al. Pair Programming: Under What Conditions Is It Advantageous for Middle School Students [J]. Journal of Research on Technology in Education, 2014, 46(3):277-296.

[4] Liebenberg J, Mentz E, Breed B. Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT)[J]. Computer Science Education, 2012, 22(3):219-236.

[5] Chen Jie. Research on teaching strategies for pair programming in primary schools [D]. Nanjing Normal University, 2016.

[6] Tsai M. J., Wang C. Y., & Hsu P. F. Developing the computer programming self-efficacy scale for computer literacy education. Journal of Educational Computing Research, 2019, 56(8), 1345-1360.