International Conference
NEW PERSPECTIVES in SCIENCE EDUCATION

Educational Effects of Pair Programming with Generative AI for Novice Learners

Mikihiko Mori[1], Yu Tsukamoto[2], Shuichiro Ogawa[3]

[1]Hosei University, Japan
[2]Benesse Corporation, Japan
[3]Office Nice, Japan

Abstract

Recent advancements in generative AI (GenAI) have enabled learners to implement software through prompt-based programming. However, novice learners, particularly students from non-computing disciplines, often lack the foundational knowledge and experience needed to interpret AI-generated code. Consequently, they struggle to identify where and how the code should be modified, making it difficult for them to engage even in basic trial-and-error refinement. Such difficulty limits opportunities for learning through debugging and iterative improvement.
In this paper, we propose an instructional approach in which two learners engage in pair programming while collaboratively designing and revising prompts for a GenAI tool and jointly refining the resulting program code to develop an application that addresses social issues relevant to the learners. The approach was carried out with 12 university students from non-computing disciplines who were novice learners with no prior programming background. Pre- and post-intervention surveys using the Design Attitude Measurement and a creativity measure based on the category of creativity called "mini-c" were administered. The Design Attitude Measurement results showed significant increases in optimism and collaboration. In contrast, the mini-c measure yielded significant pre-post differences for four items, including increased self-disappointment when participants did not generate new ideas and decreased endorsement of several positively framed creativity-related behaviors and beliefs. Taken together, these results suggest that the intervention strengthened design-oriented attitudes while also prompting more critical self-evaluation of everyday creativity-related practices.
Keywords: *Use of Generative AI, Programming Education, Pair Programming*

1. Introduction

Toward a more sustainable society, individuals increasingly need to identify and address the issues in their immediate surroundings. As ICT environments have become more accessible, people who previously did not use computers are increasingly encouraged to adopt them, enabling everyday problem solving through ICT. In practice, initiatives known as Civic Tech, in which citizens use technology to solve social problems, have been undertaken [1]. In such contexts, people should not remain mere users of computer software. Instead, they need an environment in which they can readily create software. Such software does not have to be product-level applications used by everyone. Small-scale personal tools that satisfy one's own needs are sufficient. However, for people who are not programming specialists, the barriers to programming remain high. This study proposes an instructional approach that uses generative AI (hereafter, GenAI) within pair programming activities so that complete beginners who do not aim to become experts can engage in programming with fewer barriers to solve problems in their daily lives.
Recent progress in GenAI has lowered these barriers. By simply specifying needs in a prompt, software can be generated automatically. Some of the program code produced in this way is "broken" and cannot be executed. However, much of it is immediately runnable and is likely to satisfy the requester's needs immediately and iteratively. Because GenAI can generate program code so easily, even programming beginners can more readily create software that matches their needs. However, precisely because they are beginners, they often cannot revise the output to better fit those needs.
Accordingly, this study proposes a programming education approach that actively incorporates GenAI into pair programming, in which two novice learners share a single PC and code together, with the aim of helping non-specialist beginners acquire programming skills. In human-human pair programming, learners are expected to develop proficiency through an iterative process of entering appropriate prompts to GenAI, reviewing the output, and revising both the prompts and the program code while building software. Although research on pair programming with GenAI has been conducted, it has often implicitly assumed a baseline level of programming skill by focusing on students such as

information science majors. Therefore, when considering complete beginners who aim first to complete an application, it remains unclear what instructional approach is most effective.

In this paper, we propose a GenAI-supported pair programming approach for beginner learners. We hypothesize that this instructional approach will not only help learners become familiar with coding but will also direct their attention to designing and to everyday creative activities as driving factors in software development. We therefore investigate changes in learners' attitudes toward design and in their self-perceptions of creativity. To do so, we analyze both quantitative data from psychological scales and qualitative data from open-ended responses and classroom observations. We expect that the proposed instructional approach will not only enable a broader range of learners to engage in software development but will also support their motivation to continue doing so.

2. Related Works

Prompting GenAI to generate program code can be viewed as an extension of various efforts that have existed since immediately after the public release of ChatGPT (e.g., [2], [3]). With advances in GenAI, a style of conducting the entire software development process through natural language prompts has come to be called vibe coding [4]. As the use of GenAI for programming has become more widespread, it has become difficult to directly prevent students from using GenAI for programming assignments. Consequently, instructional approaches that assume GenAI use are required [5], [6].

When experts use GenAI, modification is relatively easy because they have knowledge and experience in programming. In contrast, for programming beginners, modification is difficult because they have too little knowledge and experience to read and understand the program code output by GenAI. As a result, they may indiscriminately repeat a loop of entering prompts, running the generated code, and checking its behavior. Consequently, they may fail to produce the program they intended, or it may never be completed. Even when the code could be fixed with only minor edits, they may give up on revising it.

Pair programming has been widely adopted in programming education [7]. Its advantages include mutual learning that naturally improves software quality by enabling the sharing of knowledge about software and introducing multiple perspectives on design.

In pair programming, it is often considered preferable for the pair's expertise level to be comparable, because there is a risk that opportunities for the less proficient student to speak may be lost [8]. In pairs with different proficiency levels, the lower-proficiency student can gain substantial learning opportunities [9]. Across these studies, tasks tended to progress as the higher-proficiency student led unilaterally. Moreover, when both members of the pair are low proficiency, such as both being beginners, they may become stuck at the stage of making the code executable, and the mutual exchange of knowledge, which is a key benefit of pairing, may not be sufficiently realized.

Accordingly, approaches that combine GenAI-based code generation with pair programming have been considered. In this context, there are two ways to conceptualize pairing.

First, one approach is to form a human-GenAI pair [10]. In this configuration, GenAI effectively replaces a higher-proficiency partner. GenAI generates program code and provides explanations of that code. Learners can also modify the code and ask GenAI to explain their changes, thereby iteratively justifying and refining revisions. Simaremare et al. had students conduct pair programming with GenAI in an information science course and collected open-ended responses [11]. Students reported that GenAI's explanations were difficult and took too long to understand, that it did not provide sufficient information to narrow down what they wanted to know, and that its responses were inaccurate, unexpected, and different from what they had anticipated. This suggests that this pairing configuration may be unsuitable for the beginner learners targeted in the present study.

Second, another approach is to have two learners work as a human-human pair while using GenAI as support during the activity. Lyu et al. showed that assignment performance was higher when GenAI-supported pair programming than when students engaged in pair programming alone or vibe coding alone [12]. They argued that, in application development, it is preferable to conduct human-human pair programming while using GenAI as a supporter rather than relying on it entirely. However, because Lyu et al. targeted students in a Computer Science Department, it should be noted that their participants' baseline knowledge differs from that of non-specialist learners.

3. Methodology

In this study, we implemented a pair-programming-based learning activity for novice learners from non-computing disciplines. The programming task was application development planned by the participants. However, if application development is conducted in an improvised, ad hoc manner, GenAI can immediately output some code, which is insufficient for learners to acquire either a programming perspective or a software development perspective. Therefore, we encouraged participants to focus on creating an application as a service required by users. Accordingly, we incorporated the concept of service design, as well as methods for eliciting user feedback and conducting interviews. Within these constraints, participants were required to achieve software development through GenAI-supported pair programming.

The learning program consisted of two sessions of 200 minutes each, with a two-week interval between Session 1 and Session 2. In Session 1, participants learned the concept of service design. In service design, prototyping is essential. As a prototyping method, participants were introduced to vibe coding. Participants then planned an application individually and conducted GenAI-supported pair programming in pairs (two students per group) to develop a working application as the Session 1 deliverable. At the end of Session 1, participants were assigned homework to continue application development individually through vibe coding until the next session. In Session 2, participants first shared their homework progress by showing their applications to each other. They then learned a method for evaluating applications by interviewing users to solicit feedback. Using the applications they had developed, participants conducted peer interviews among themselves. Based on the interview results, they engaged in another round of GenAI-supported pair programming and completed applications that reflected the user interview findings.

Twelve students participated in this learning activity. They were volunteers who responded to a call for participation among second- to fourth-year students enrolled in a course taught by the first author. All participants were from non-computing disciplines and specialized in social science. Although the activity was conducted as part of a class, we assured participants that neither the process nor any outcomes would affect their grades. Pairs were formed by students who happened to be seated next to each other at the time, and pairing was not otherwise controlled. Because the participants were acquainted with one another, we consider it unlikely that pairing differences substantially affected the outcomes.

Participants were asked to develop a web application. Because no server-side environment was prepared, development was restricted to a client-side web application that runs entirely in the browser, primarily using HTML and JavaScript. To enable immediate execution of GenAI-generated code, we initially assumed the use of ChatGPT's Canvas feature and asked participants to use the free version of ChatGPT. In practice, some participants quickly reached usage limits, and therefore also used other GenAI services such as Gemini. Accordingly, the GenAI services used were not restricted. When participants did not use a Canvas-like environment, they were permitted to copy and paste and edit generated code in an editor of their choice and to test it by running it in a web browser.

Pre- and post-intervention surveys using the Design Attitude Measurement [13] and a creativity measure based on the category of creativity called "mini-c" [14] were administered.

The Design Attitude Measurement is an instrument that assesses individual orientations using a self-reported format with a six-point Likert scale [13]. It consists of 15 items rated on a six-point Likert scale and yields five design-attitude categories: experimentalism, optimism, visualization, collaboration, and empathy.

In addition, mini-c is defined as "Intrapersonal creativity that is part of the learning process" [14]. It captures everyday creativity and does not require creativity to be either novel or useful. In this study, we used the measure proposed by Kondo and Nagai [15], which derives 10 categories from 22 items rated on a five-point Likert scale, and treated it as a measure of change in creativity-related processes.

In addition to the scale responses, participants provided written responses to open-ended questions about insights gained from the learning activities. We used these responses to contextualize the quantitative findings and report representative excerpts. We also observed learners' activities. After each session, the three authors shared their observations and reflections. In the following sections, we report the survey and observation results and discuss their implications.

4. Results and Discussion

Table 1 presents participants' pre- and post-intervention responses to the mini-c measure, together with paired-samples t-test results for each item. Significant pre-post differences were found for four items.

The self-reappraisal item "I feel disappointed in myself when I do not think of new ideas to improve something in my daily life" showed a significant increase from pre to post (t(11) = 2.324, p = .042). This suggests that after the learning activity, participants held a more negative impression of themselves when they did not engage in idea generation.

Table 1. mini-c items: pre/post descriptive statistics and paired-samples t-test results.

Category	Item	Pre M	Pre SD	Post M	Post SD	t	p
Raising awareness	I collect new information related to thinking of new ideas to improve something in my daily life.	4.182	0.603	3.727	0.467	-1.838	0.096
Raising awareness	I look for methods that would help me become able to think of new ideas to improve something in my daily life.	3.818	0.874	3.545	0.522	-0.711	0.493
Emotional arousal	I feel upset when I realize the negative consequences caused by not thinking of new ideas to improve something in my daily life.	3.091	1.136	3.273	0.905	0.392	0.703
Emotional arousal	I personally feel the negative consequences caused by not thinking of new ideas to improve something in my daily life.	3.091	1.221	3.273	0.647	0.392	0.703
Self-reappraisal	I feel disappointed in myself when I do not think of new ideas to improve something in my daily life.	2.364	1.286	3.182	0.874	2.324	0.042*
Self-reappraisal	I feel good about myself when I think of new ideas to improve something in my daily life.	4.000	0.775	3.727	0.467	-0.896	0.391
Environmental reappraisal	I think that if I become able to think of new ideas to improve something in my daily life, I can contribute to society.	4.000	1.095	3.545	0.688	-1.336	0.211
Environmental reappraisal	I think that if I become able to think of new ideas to improve something in my daily life, the people around me will also be happier.	3.909	1.300	3.636	0.505	-0.582	0.574
Self-liberation	I tell myself that I can think of new ideas to improve something in my daily life.	3.000	1.342	3.273	0.905	0.463	0.653
Self-liberation	I pledge to myself that I will think of new ideas to improve something in my daily life.	3.091	1.375	3.636	0.505	1.200	0.258
Self-liberation	I openly tell my family and friends that I will think of new ideas to improve something in my daily life.	2.455	1.128	2.909	1.300	0.787	0.450
Helping relationships	When I cannot think of new ideas to improve something in my daily life, there is someone who will listen to me.	3.455	1.293	3.182	0.982	-0.637	0.539
Helping relationships	There is someone who encourages me to think of new ideas to improve something in my daily life.	3.000	1.414	3.364	1.027	0.770	0.459
Counterconditioning	When I cannot come up with a new idea to improve something in my daily life, I pause and think about it at another time.	4.455	0.522	3.455	0.688	-3.317	0.008**
Counterconditioning	When I cannot come up with a new idea to improve something in my daily life, I try to relax.	3.818	1.401	3.182	0.874	-1.641	0.132
Reinforcement management	When I manage to think of a new idea to improve something in my daily life, I praise myself.	4.455	0.522	3.727	0.467	-3.068	0.012*
Reinforcement management	When I manage to think of a new idea to improve something in my daily life, there is someone who praises me.	3.545	1.368	3.273	1.009	-0.559	0.588
Reinforcement management	I believe that thinking of new ideas to improve something in my daily life is a good thing.	4.364	0.674	3.455	0.522	-2.887	0.016*
Stimulus control	I try to always be prepared to think of new ideas to improve something in my daily life.	2.909	1.221	3.182	0.982	0.671	0.518
Stimulus control	I make plans so that I can think of new ideas to improve something in my daily life.	2.273	1.009	2.727	1.009	1.838	0.096
Social liberation	I feel that society expects people to think of new ideas to improve something in my daily life.	3.727	1.104	3.364	0.924	-1.789	0.104
Social liberation	I feel that more people have started thinking of new ideas to improve something in my daily life.	3.273	1.348	3.273	0.786	0.000	1.000

Note. Paired-samples t-tests were conducted for each item (n = 12; df = 11). Differences were computed as Post−Pre. p values are two-tailed. * $p < .05$, ** $p < .01$.

Table 2. Design Attitude Measurement items: pre/post descriptive statistics and paired-samples t-test results.

Category	Item	Pre M	Pre SD	Post M	Post SD	t	p
Experimentalism	When an idea fails, I mainly focus on what I can learn from that experience.	4.364	0.505	4.636	0.924	0.896	0.391
Experimentalism	Even from a solution that fails, I gain a better opportunity to learn more about the problem.	4.636	0.809	5.091	0.944	1.102	0.296
Experimentalism	I always focus on what I can learn from a failed idea.	4.000	1.183	4.273	1.009	0.539	0.602
Experimentalism	When I receive feedback on a solution, I pause to consider how that information should be used.	4.545	1.214	4.545	0.820	0.000	1.000
Optimism	No matter how difficult a problem is, it can be solved.	3.727	1.679	4.364	0.924	1.472	0.172
Optimism	For any kind of problem, it is possible to create a solution.	3.636	1.567	4.636	0.924	2.803	0.019*
Optimism	I believe that all problems can be solved.	3.364	1.912	3.727	1.555	0.559	0.588
Trust in visualization	To show how a solution works, I prefer to use visual representations.	4.182	1.079	4.909	0.944	2.185	0.054
Trust in visualization	I like to explain my ideas to others using visual representations.	4.364	1.206	4.273	1.348	-0.161	0.875
Trust in visualization	Visual representations are a very good way to explain ideas.	5.273	0.905	5.091	0.831	-0.559	0.588
Collaboration	I know that the best solutions come from sharing ideas.	3.727	1.421	4.727	0.786	2.236	0.049*
Collaboration	When working on solving a problem, I actively try to connect with other people.	3.182	1.401	4.182	0.874	2.057	0.067
Collaboration	It is very important to incorporate ideas suggested by other people to complete a solution.	4.182	1.250	4.273	1.104	0.232	0.821
Empathy	I can easily view a problem from the perspective of people who are troubled by the situation.	4.727	1.348	4.364	1.362	-1.000	0.341
Empathy	When people consult me about the impact of a problem, I can easily understand the situation from their perspective.	4.545	1.440	4.364	1.433	-0.559	0.588

Note. Paired-samples t-tests were conducted for each item ($n = 12$; $df = 11$). Differences were computed as Post−Pre. p values are two-tailed. * $p < .05$, ** $p < .01$.

In addition, three items that capture adaptive creativity-related behaviors and beliefs showed significant decreases from pre to post: the counterconditioning item "When I cannot come up with a new idea to improve something in my daily life, I pause and think about it at another time" ($t(11) = -3.317$, $p = .008$), the reinforcement management item "When I manage to think of a new idea to improve something in my daily life, I praise myself" ($t(11) = -3.068$, $p = .012$), and the reinforcement management item "I believe that thinking of new ideas to improve something in my daily life is a good thing" ($t(11) = -2.887$, $p = .016$). Although the significant effects differed in sign, they were consistent in valence. The negatively worded self-reappraisal item increased, and three positively worded items decreased. Taken together, these shifts indicate heightened self-criticism regarding everyday creativity-related behaviors and beliefs.

Table 2 presents participants' pre- and post-intervention responses to the Design Attitude Measurement, together with paired-samples t-test results for each item. Significant pre-post differences were found for two items. The optimism item "For any kind of problem, it is possible to create a solution" showed a significant increase from pre to post ($t(11) = 2.803$, $p = .019$). This suggests that after the learning activity, participants became more confident that solutions can be developed even for diverse types of problems. The collaboration item "I know that the best solutions come from sharing ideas" also increased significantly ($t(11) = 2.236$, $p = .049$), indicating stronger recognition of the value of idea sharing in problem solving.

The mini-c results suggest that participants developed a partially negative impression of their own creative activities following the learning activity. In contrast, the Design Attitude Measurement results suggest that participants became partially more positively oriented toward designing. At first glance, these findings may appear contradictory. Moreover, based on the mini-c results alone, it is also possible to interpret the learning activity as unsuccessful. However, examining the open-ended responses provides another perspective that integrates the mini-c and Design Attitude Measurement results.

In response to the open-ended question “Please tell us what you thought was important in the learning activity,” participants wrote:

- “I often do not get results that match what I imagined even when I give instructions to ChatGPT, but I learned that it is important not to give up and to give instructions in more specific wording.”
- “I realized both how difficult and how important it is to look at what I created objectively.”

In response to the open-ended question “If anything in your way of thinking changed through the learning activity, please describe it,” one participant wrote:

- “Even if I think what I made is the best, it might not be good for other people.”

In response to the open-ended question “Please freely write what you felt after experiencing pair programming with GenAI in the class and vibe coding alone in the interim assignment,” participants wrote:

- “When I work in a pair, I can see things that are hard to notice when I am alone.”
- “When I worked in a pair, even if we ran into difficulties, we could think things through together and I did not give up. Because we shared and discussed our ideas, I felt confident about the app we made. When I worked alone, I tended to give up or I could not feel confident, so I preferred working in a pair. At the same time, working alone was also easier in the sense that I could quickly type what I was thinking.”
- “Compared with doing vibe coding alone, pair programming often helped me because my partner pointed out problems, I would not notice by myself and gave me ideas for new features, so it felt stronger on the creativity side, like coming up with ideas.”
- “With pair programming, not only could I bring in ideas I would never think of on my own, but I could also code while talking, which was fun. Vibe coding alone is nice because I can do things exactly how I want, but it turns into silent work, so I felt it would be tough to keep doing it for a long time.”

These comments suggest that in the pre-survey, participants may have answered by imagining their usual behavior in student life and may not have treated programming or application development as targets of self-evaluation, having only a vague image of them. In contrast, in the post-survey, through application development involving GenAI-supported pair programming, participants thought more concretely about what they themselves needed to do. As a result, they had an opportunity to make more concrete but stricter self-evaluations. In mini-c, which asks about one’s current self-perception in everyday creativity, this may have led to more negative responses. Meanwhile, because the Design Attitude Measurement asks what one should do when designing and solving problems, the activity may have confirmed that participants acquired this perspective, leading to more positive responses.

In our informal classroom observations, participants often preferred a cycle of revising prompts and repeatedly asking GenAI to generate code, rather than directly modifying program code. They repeatedly tried different prompts and effectively bet that continued trials might lead to success. This behavior can be understood as leveraging a key characteristic of GenAI, namely that it can rapidly generate some form of code in response to a given prompt. When issuing instructions in natural language is perceived as easier than writing code directly, iteratively rephrasing prompts until the output meets one’s expectations becomes a comparatively accessible strategy. In this sense, the threshold for obtaining a workable result is lowered.

In this setting, attention tended to shift from sharing and improving code itself to sharing and improving prompts, and participants sometimes discussed and exchanged prompt ideas with their partners. Indeed, we observed participants teaching each other what content to include in prompts. This is also consistent with a response to the open-ended question “If anything in your way of thinking changed through the learning activity, please describe it.” One participant wrote:

- “My awareness of articulating requirements in language increased.”

Working with their partner, participants engaged in a broader range of trial-and-error prompt revisions. This produced more prototypes and helped align the resulting applications with participants’ expectations. Compared with struggling alone to craft prompts while vibe coding, participants appeared to enjoy having GenAI generate program code more when working in pairs, as they could test a wider range of prompts from multiple perspectives.

Taken together, although the mini-c measure suggests a seemingly negative shift in participants’ self-perceptions of creative activity, the process of concretely understanding creative activity through vibe coding and GenAI-supported pair programming and engaging in reflection may have contributed to the more positive outcomes captured by the Design Attitude Measurement, such as increased confidence in creative activity as problem solving.

5. Conclusion

This study proposed a learning method that uses GenAI in pair programming as a framework enabling undergraduate programming beginners from non-computing disciplines to engage in application development. In this application development, we had participants envision applications that intended users would want to use, thereby encouraging them to create concrete and more authentic applications. The program included two 200-minute in-class learning sessions and a two-week period of extracurricular learning activity. The results suggest that, regarding creativity, participants came to view themselves more realistically and critically, as indicated by mini-c responses, which measure self-recognition of everyday creative learning. Regarding design attitudes, participants showed increased recognition of attitudes toward problem solving and the need to share ideas. Vibe coding may lower the barriers for novice learners to prototype applications. However, open-ended responses and observations also suggest that, to address problems that beginners cannot notice on their own, it becomes easier to resolve them by using GenAI within pair programming.
Several limitations should be noted. The study drew on a small sample from a single course at one university (n = 12), which limits the generalizability of the findings. Because no control condition was included, the observed pre-post differences cannot be interpreted as causal effects of GenAI-supported pair programming. The GenAI tools and development environments were not standardized across participants (e.g., ChatGPT vs. Gemini; use of Canvas feature vs. external editors), which may have introduced variability in the level and form of support available during development and influenced the learning process and outcomes.

REFERENCES

[1] Knight Foundation, “The Emergence of Civic Tech: Investments in a Growing Field,” SlideShare. Accessed: Aug. 30, 2025. [Online]. Available: https://www.slideshare.net/slideshow/knight-civictech/28862111
[2] Liu C., Bao X., Zhang H., Zhang N., Hu H., Zhang X., and Yan M., “Improving ChatGPT Prompt for Code Generation,” arXiv:2305.08360. 2023. doi: 10.48550/arXiv.2305.08360.
[3] Liu J., Xia C.S., Wang Y., and Zhang L., “Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation,” Advances in Neural Information Processing Systems, vol. 36, pp. 21558–21572, 2023.
[4] Ge Y., Mei L., Duan Z., Li T., Zheng Y., Wang Y., Wang L., Yao J., Liu T., Cai Y., Bi B., Guo F., Guo J., Liu S., and Cheng X., “A Survey of Vibe Coding with Large Language Models,” arXiv:2510.12399. 2025. doi: 10.48550/arXiv.2510.12399.
[5] Lau S., and Guo P., “From ‘Ban It Till We Understand It’ to ‘Resistance is Futile’: How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot,” in Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER ’23), vol. 1. New York, NY, USA: Association for Computing Machinery, pp. 106–121, 2023. doi: 10.1145/3568813.3600138.
[6] Prather J., Leinonen J., Kiesler N., Benario J.G., Lau S., MacNeil S., Norouzi N., Opel S., Pettit V., Porter L., Reeves B.N., Savelka J., Smith D.H., Strickroth S., and Zingaro D., “Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools,” in Proceedings of 2024 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE 2024). New York, NY, USA: Association for Computing Machinery, pp. 300–338, 2025. doi: 10.1145/3689187.3709614.
[7] Hanks B., Fitzgerald S., McCauley R., Murphy L., and Zander C., “Pair programming in education: a literature review,” Computer Science Education, vol. 21, no. 2, pp. 135–173, 2011. doi: 10.1080/08993408.2011.579808.
[8] Cao L. and Xu P., “Activity Patterns of Pair Programming,” in Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005), Big Island, HI, USA: IEEE Computer Society, pp. 88a–88a, 2005. doi: 10.1109/HICSS.2005.66.
[9] Chaparro E.A., Yuksel A., Romero P., and Bryant S., “Factors Affecting the Perceived Effectiveness of Pair Programming in Higher Education,” in Proceedings of 17th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2005), Brighton, UK, pp. 5–18, 2005.

[10] Spinellis D., “Pair Programming with Generative AI,” IEEE Software, vol. 41, no. 3, pp. 16–18, 2024. doi: 10.1109/MS.2024.3363848.

[11] Simaremare M., Pardede C., Tampubolon I., Manurung P., and Simangunsong D., “Pair Programming in Programming Courses in the Era of Generative AI: Students’ Perspective,” in Proceedings of 2024 31st Asia-Pacific Software Engineering Conference (APSEC), pp. 507–511, 2024. doi: 10.1109/APSEC65559.2024.00069.

[12] Lyu W., Wang Y., Sun Y., and Zhang Y., “Will Your Next Pair Programming Partner Be Human? An Empirical Evaluation of Generative AI as a Collaborative Teammate in a Semester-Long Classroom Setting,” in Proceedings of the Twelfth ACM Conference on Learning @ Scale (L@S ’25). New York, NY, USA: Association for Computing Machinery, pp. 83–94, 2025. doi: 10.1145/3698205.3729544.

[13] Yaegashi K., Ando T., Goto S., and Morita T., “Developing the Tool for Measuring the Design Capability of Companies,” Journal of Design Science, vol. 1, pp. 103–118, 2022. doi: 10.34382/00016127. (in Japanese)

[14] Beghetto R.A., and Kaufman J.C., “Toward a broader conception of creativity: A case for ‘mini-c’ creativity,” Psychology of Aesthetics, Creativity, and the Arts, vol. 1, no. 2, pp. 73–79, 2007. doi: 10.1037/1931-3896.1.2.73.

[15] Kondo K., and Nagai Y., “Research on Enhancement of Creativity,” Journal of Japan Creativity Society, vol. 21, pp. 42–63, 2018. doi: 10.24578/japancreativity.21.0_42. (in Japanese)