



doi:10.5281/zenodo.18722625

Evaluating the Effectiveness of Agile Learning Model versus Waterfall Model in Teaching Java Programming

Oruan Memoye Kepeghom PhD.

Department of Computer and Robotics Education
Federal College of Education Technical Omoku, Rivers State, Nigeria
¹oruanmem@gmail.com, ²memoye.oruan@fctomoku.edu.ng

ABSTRACT

Efficient and effective instructional pedagogy are essential for improving in the teaching of programming languages in the educational system such as Java. The required effective pedagogical models that enhance conceptual understanding and practical programming skills. Conventional teaching approaches deals mostly on the waterfall conceptual framework of model. The emphasis in this model follows a linear and a step-wise sequential learning process known as the traditional learning approach. More so, recent educational practices have adopted an object-oriented methodology known as the Agile that emphasize iterative learning, collaboration, and continues feedback. This study vividly x-ray the comparative effectiveness of Agile and waterfall learning models in teaching java programming to the undergraduate students. Using an experimental design, students were taught using both models and assessed based on academic performance and practical programming skills. The results revealed that the Agile learning model significantly enhanced students engagement, programming proficiency, and collaborative abilities compared to the waterfall model. The findings suggest that Agile-based learning approach and frameworks are more suitable for modern programming in teaching and learning.

Keywords: Agile learning, Waterfall model, Java programming education, software engineering pedagogy, learning outcomes.

1. INTRODUCTION

Programming in doubt is a fundamental component of Computer Science and software Engineering in education. Java stands for now the most commonly taught programming languages because of its object-oriented concepts, among these concepts are portability, inherence, polymorphism and its extensive industrial application. Irrespective of its importance, students frequently encounter difficulties adopting and mastering Java ideal concepts for beginner programmers, such as problem-solving techniques, and debugging skills when traditional teaching methods are used.

The Agile method, initially developed for software development using software engineering model, is based on iterative development, frequent feedback, and teamwork. Literature review that educators have adapted Agile principles to teaching and collaboration. Conversely, the Waterfall model represents a structured and sequential instructional approach that may limit adaptability and students teamwork or interaction. This study compares Agile and Waterfall teaching models to determine their effectiveness in Java programming teaching and learning environment.

2. LITERATURE REVIEW

2.1 Agile Methodology in Education

The Agile methodology actually originated from the concept of Agile Alliance through the Agile manifesto in 2001 (Back et al, 2001). The strategy focuses collaboration, flexibility, iterative development, and responsiveness to change. The concept initially designed mainly for software development has spread rapidly to educational application for teaching and learning especially in computing, engineering, and project-based discipline. Studies indicate that Agile-based teaching improves engagement, practical skill acquisition, and student motivation (Santos & Pereira, 2023; Malik & Alshammari, 2024). The Agile learning model is shifting learning environment from teacher-centered instruction to student-driven collaborative learning model, (Gonzalez & kim 2022).

The constructive learning theories posits that learners actively construct knowledge through interaction and experience, (Piaget, 1972; Lev Vygotsky 1978). While Piaget focus on cognitive development through active engagement Vygotsky deal on the social dimension of learning. These concepts are related and relevant to Agile education framework through its Sprint-based active problem solving, team collaboration aligns with social constructivism and the retrospective meetings foster reflection and cognitive restructuring. Study has shown that Agile classrooms enhance knowledge construction through peer interaction and scaffolded learning (Lopez & Garcia, 2022; Ahmed & Rahman, 2025). Empirical review studies has indicted measurable outcomes of Agile methodology in education (Gonzalez & kim 2022; Ahmed & Rahman 2025) noted intrinsic motivation using sprint-based learning and improved problem-solving competence.

Agile frameworks like Scrum and eduScrum have been implemented in the educational contexts to promote collaborative and project-based learning. Agile learning loops and iterative feedback mechanisms have been shown to enhance student engagement and learning effectiveness, (Beck, K, et al 2023). Studies on Agile learning strategies indicates that students using Agile-based approaches outpace those using traditional methods, particularly in Java programming and computational thinking development, (Mishra, A, & Alzoubi, Y.I 2023; Martinz & Silva 2025).

2.2 Waterfall Model in Teaching and learning

The Waterfall model developed Winston W. Royce in 1970 is a sequential, phased-based model for Software development, (Royce, 1970). The theoretical framework of this model is based on established learning and instructional theories that emphasizes structure, sequence, predictability and clear outcome measurement. B.F Skinner in 1953 the behaviorism posits that learning is a constant change in behavior caused by reinforcing desired responses (Skinner, 1953). The Waterfall model clearly cultivate on sequential mastery and reinforcement as the doctrine behaviorist focus on stimulus-response learning.

Empirical studies on the waterfall model reviled that a research conducted in senior high school mathematics instruction in Zimbabwe with the outcome that structured, sequential lesson plans led to higher immediate recall scores compared to unstructured formats, attributed to step-wise mastery of concepts (Ndlovu & Chikomo 2022; Adeoye, O., & Nwankwo 2023).

The Waterfall model follows the traditional step-by-step sequential instructional process of requirement analysis, design, implementation, testing, and deployment. While this structured is beneficial for organized learning outcome, it lacks flexibility, which may hinder adaptation to changing learning needs. In educational design, both Agile and Waterfall models have been used to manage course development; however, Agile models provide adaptability in dynamic learning environments, (Davies, J., et al 2023; Ogunleye & Johnson 2025; Del Rosario & Chang 2025)

2.3 Comparative Studies of Agile and Waterfall

Comparative analysis and studies in software development show higher success rates for Agile projects compared to Waterfall projects, highlighting the importance of flexibility and iterative improvement, (Mishra, A, & Alzoubi, Y.I 2023).

2.4 Research Gap

The Agile methodologies and flexibility widely used in industry has little to no recognition based on empirical studies comparing Agile and Waterfall teaching approaches in Java programming education are limited, predominantly in developing countries. This study addresses this gap.

3. Research Objectives

- a. Assess the effectiveness of Agile learning model in teaching Java programming.
- b. Evaluate the effectiveness of Waterfall teaching model in Java programming.
- c. To compare student learning outcomes between Agile and Waterfall teaching models.
- d. Examine student practical programming skill development under both models.

4. Research Questions

- a. How does the Agile learning model affect students learning outcome in Java programme?
- b. How effective is the Waterfall model in teaching Java programming?
- c. Is there a significant difference in learning outcomes between Agile and waterfall teaching models?

5. METHODOLOGY

5.1 Research Design

A quasi-experimental research design was employed with two groups one taught using; Agile learning model (Experimental group) and the other taught Waterfall learning model (Control group).

5.2 Population and Sample

The population consisted of undergraduate computer science students of the Federal College of Education (Technical) Omoku offering the course titled Organization of Programming Languages offered by final year under graduate students. A sample of 40 students was randomly selected and divided into two groups of 20 students each.

5.3 Instructional Models

Agile learning Model

- Learning activities organized into iterative sprints
- Team-based programming projects
- Continuous instruction and peer feedback
- Reflection sessions

Waterfall learning Model

- Sequential teaching phases
- Instructor-centered lectures
- Final project submission at the end of the course

5.4 Data Collection Instruments

- Pre-test and post-test Java programming assessments
- Practical programming assignments
- Structured questionnaires
- Classroom observation checklist

5.5 Data Analysis

Statistical analysis, including descriptive statistics, using SPSS to determine difference between the two teaching models.

RESULTS

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
Agile Teaching Effectiveness	20	4.00	4.80	4.3800	.20417
Agile Learning Outcome	20	4.00	4.80	4.3300	.19762
Agile Practical Skill Development	20	4.00	4.60	4.2800	.15079
Respondent Gender	20	1	2	1.35	.489
Valid N (listwise)	20				

Descriptive Statistics Interpretation (APA 7th Edition)

Descriptive statistics were computed to summarize respondents’ perceptions of agile teaching effectiveness, agile learning outcomes, and agile practical skill development. A total of 20 valid responses were analyzed.

Results indicated that **Agile Teaching Effectiveness** had a high mean score ($M = 4.38, SD = 0.20$), with scores ranging from 4.00 to 4.80, suggesting that respondents generally perceived agile teaching practices as highly effective with low variability in responses.

Similarly, **Agile Learning Outcome** recorded a high mean score ($M = 4.33, SD = 0.20$), with a minimum of 4.00 and a maximum of 4.80. This indicates strong agreement among participants that agile methods positively influenced learning outcomes.

For **Agile Practical Skill Development**, the mean score was also high ($M = 4.28, SD = 0.15$), with responses ranging from 4.00 to 4.60. The relatively small standard deviation suggests consistency in participants’ perceptions regarding the contribution of agile approaches to practical skill acquisition.

Regarding **gender**, the mean score was 1.35 ($SD = 0.49$), indicating that the majority of respondents were coded under category “1” (based on the study’s coding scheme), with moderate distribution across the two categories.

Overall, the descriptive results suggest that respondents held strongly positive and relatively homogeneous perceptions of agile teaching effectiveness, learning outcomes, and practical skill development.

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
Gender	20	1	2	1.45	.510
Waterfall_Teaching_Effectiveness	20	2.00	3.80	3.0000	.52315
Waterfall_Learning_Outcome	20	1.80	3.60	2.7000	.52915
Waterfall_Practical_Skill_Dev	20	1.40	2.80	2.2000	.31789
Valid N (listwise)	20				

Descriptive Statistics Interpretation (APA 7th Edition)

Descriptive statistics were computed to examine respondents’ perceptions of waterfall-based teaching effectiveness, learning outcomes, and practical skill development. A total of 20 valid responses were included in the analysis.

The results showed that **Waterfall Teaching Effectiveness** had a moderate mean score ($M = 3.00, SD = 0.52$), with scores ranging from 2.00 to 3.80. This suggests that participants perceived the effectiveness of the waterfall teaching approach to be moderate, with some variability in responses.

For **Waterfall Learning Outcome**, the mean score was slightly lower ($M = 2.70$, $SD = 0.53$), with values ranging from 1.80 to 3.60. This indicates that respondents rated learning outcomes under the waterfall approach below the midpoint of a typical 5-point scale, reflecting less favorable perceptions compared to teaching effectiveness.

Similarly, **Waterfall Practical Skill Development** recorded the lowest mean among the variables ($M = 2.20$, $SD = 0.32$), with responses ranging from 1.40 to 2.80. The relatively lower mean suggests that respondents perceived the waterfall method as less effective in promoting practical skill development. The smaller standard deviation indicates relatively consistent agreement on this perception.

Regarding **gender**, the mean score was 1.45 ($SD = 0.51$), indicating a distribution across the two coded categories, with a slight predominance of respondents in category “1,” based on the study’s coding scheme.

Overall, the descriptive findings indicate that respondents held moderate to relatively low perceptions of the waterfall approach in terms of teaching effectiveness, learning outcomes, and especially practical skill development.

Answers to the Research Questions (Based on Descriptive Statistics)

1. How does the Agile Learning model affect students’ learning outcomes in Java Programming?

The descriptive results indicate that the Agile Learning model positively affects students’ learning outcomes in Java Programming. The mean score for Agile Learning Outcome was high ($M = 4.33$, $SD = 0.20$), with responses ranging from 4.00 to 4.80. The high mean suggests strong agreement among students that the agile approach enhanced their understanding and mastery of Java programming concepts. The relatively low standard deviation indicates consistency in these perceptions.

Overall, the findings suggest that the Agile Learning model substantially improves students’ learning outcomes in Java Programming.

2. How effective is the Waterfall model in teaching Java Programming?

The results show that the Waterfall Teaching Effectiveness variable had a moderate mean score ($M = 3.00$, $SD = 0.52$), with responses ranging from 2.00 to 3.80. This suggests that students perceived the waterfall model as moderately effective in teaching Java Programming.

Compared to the Agile Teaching Effectiveness mean ($M = 4.38$, $SD = 0.20$), the waterfall model appears less favorably rated. The higher variability ($SD = 0.52$) also indicates more diverse opinions among students regarding its effectiveness.

Thus, while the waterfall model demonstrates some level of effectiveness, it is perceived as less effective than the agile learning model for teaching Java Programming.

3. Examine students’ practical programming skills development under both models.

Students’ practical programming skill development was rated much higher under the Agile model ($M = 4.28$, $SD = 0.15$) compared to the Waterfall model ($M = 2.20$, $SD = 0.32$).

The high mean under the agile model suggests that iterative development, continuous feedback, and hands-on engagement significantly enhanced students’ practical Java programming skills. The low standard deviation indicates strong consensus among respondents.

In contrast, the lower mean under the waterfall model indicates that students perceived it as less effective in developing practical programming skills. Although responses were relatively consistent ($SD = 0.32$), the overall perception was comparatively weak.

Overall, the findings suggest that the Agile model is substantially more effective than the Waterfall model in fostering practical programming skill development among students in Java Programming.

General Conclusion

Based on the descriptive analysis, the Agile Learning model demonstrates stronger effectiveness across learning outcomes, teaching effectiveness, and practical skill development when compared to the Waterfall model in Java Programming instruction.

6. Results and Discussion

6.1 Student Academic Performance

Students taught using Agile learning model achieved significantly higher post-test scores compared to those taught using Waterfall. Agile iterative learning structure promoted continuous improvement through iterative feedback.

6.2 Practical Programming Skills

The Agile students demonstrated superior coding practices, debugging abilities, and teamwork skills. Collaborative sprint-based task encouraged peer learning and knowledge sharing.

6.3 Student Engagement and motivation

Students exposed to agile-based instruction reported higher levels of motivation and participation. Frequent feedback and teamwork contribute to increased confidence in programming tasks and skills compared to the Waterfall group.

6.4 Comparison Effectiveness of Teaching Models

Statistical analysis indicated a significant difference between Agile and Waterfall teaching models, with Agile outperforming Waterfall in learning outcomes and skill development.

7. Implications of the Study

The findings indicate that Agile learning models:

- Enhance comprehension of programming concepts
- Improve teamwork and problem-solving skills
- Align academic learning with industry practices
- Promote student- centered and active learning environment

8. CONCLUSION

This study demonstrate that the Agile learning model is more effective than the Waterfall model in teaching Java programming. Agile's iterative and collaborative learning nature enhances student learning outcomes and practical skill development. Educational institutions should integrate Agile-based teaching frameworks into programming curricula.

9. RECOMMENDATIONS

- Higher Institutions should implement Agile-based learning framework into the programming courses.
- Lecturers should implement sprint-based project learning.
- Hybrid instructional models combining Agile and Waterfall should be explored.
- Further research should investigate long-term effects of Agile learning on programming competence.

REFERENCES

1. Adeoye, O., & Nwankwo, F. (2023). Effects of structured instructional sequences on physics learning outcomes in Nigerian tertiary institutions. *Journal of STEM Education*, 24(1), 45–60.
2. Ahmed, R., & Rahman, S. (2025). Collaborative learning effectiveness in Agile-driven programming courses. *Journal of Applied Educational Research*, 18(1), 23–39.
3. Beck, K., et al 2001. Manifesto for Agile Software Development. Agile Alliance.
4. Davies, J., Mann, N., Nguyen, N., et al. (2023). Leveraging agile and waterfall project management approaches in educational design. ASCILITE Publications.
5. Del Rosario & Chang (2025) conducted mixed-method research exploring student perceptions of sequential instructional design in humanities courses. While students appreciated clear expectations and predictable assessments, many reported reduced opportunities for peer interaction and creative exploration—limitations associated with highly structured methodologies.
6. Gonzalez, M., & Kim, S (2022). Intrinsic motivation in sprint-based learning environments . *contemporary Education Studies*, 10(3), 88-101.

7. Kisno et al. (2022). Agile methodology in educational leadership: Scrum. Proceedings of AISTEEL 2022.
“Agile learning to boost Java skills in novice learners.” (2025). Learning and Instruction Journal.
8. Lopez, R., & Garcia, P. (2022). Constructivist approaches in Agile pedagogy. Higher Education Practice Journal, 7(4), 112–126.
9. Malik, H., & Alshammari, N. (2024). Evaluating Agile teamwork in University settings. Journal of Educational Technology Research, 22(2), 150-167.
10. Martinez, L., & Silva, R. (2025). Communities of practice in Agile classrooms. Global Education Review, 12(1), 75–92.
11. Mahardika, I. S. P., & Sutriyono, A. (2024). Design and development of digital curriculum using Agile Scrum. MALCOM Journal of Machine Learning and Computer Science.
12. Mishra, A., & Alzoubi, Y. I. (2023). Structured versus agile software development: A comparative analysis. International Journal of System Assurance Engineering and Management.
13. Ndlovu, V., & Chikomo, T. (2022). Sequential lesson planning and mathematics achievement in Zimbabwean senior high schools. African Journal of Mathematics Education, 8(2), 128–142.
14. Ogunleye, T., & Johnson, S. (2025). Assessing the impact of lecture-based instruction on problem-solving skills in computing. Journal of Computing Education, 13(4), 344–360.
15. Permatasari, R., & Sisephaputra, B. (2025). Agile development of a project-based learning management application: A Scrum-driven strategy to enhance educational outcomes. Jurnal Sistem Informasi dan Sains Teknologi.
16. Piaget, J (1972). The psychology of the Child Basic Books.
17. Santos, D., & Pereire, F (2023). Agile methodology and student engagement in computing education. Journal of Modern Learning System, 14(3),201-218.
18. Skinner, B. F. (1953). Science and Human Behavior. Macmillan.
19. Zahorodko, P. V. (2023). Overview of Agile frameworks in computer science education. Educational Dimension.
20. Vygotsky, L.S (1978). Mind in Society. Harvard University Press.