ผลของแนวทางการเขียนโปรแกรมแบบ Vibe Coding ภายใต้บริบทปัญญาประดิษฐ์เชิงสร้างสรรค์ต่อประสิทธิภาพการพัฒนาซอฟต์แวร์และการเรียนรู้ของนักศึกษาสาขาเทคโนโลยีดิจิทัลและนวัตกรรม

Effects of Vibe Coding within the Context of Generative Artificial Intelligence on Software Development Performance and Learning among Digital Technology and Innovation Students

ชนินทร เฉลิมสุข[1*], ศิลป์ณรงค์ ฉวีพัฒน์[2] และ ธนบดี ศรีธนนันท์[3]

[1, 3] สาขาวิชาเทคโนโลยีดิจิทัลและนวัตกรรม คณะศิลปศาสตร์และวิทยาศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์

[2] สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏกำแพงเพชร

*ติดต่อ: chanintornc@sau.ac.th, เบอร์โทรศัพท์ : 064-170-0888

บทคัดย่อ

แม้ว่าเทคโนโลยีปัญญาประดิษฐ์เชิงสร้างสรรค์ (Generative AI) จะได้รับความนิยมอย่างแพร่หลายในวงการพัฒนาซอฟต์แวร์ แต่งานวิจัยเชิงประจักษ์เกี่ยวกับอิทธิพลทางการสอนในบริบทของการศึกษาระดับอุดมศึกษา โดยเฉพาะในประเทศไทย ยังคงมีอยู่อย่างจำกัด การศึกษาครั้งนี้มีวัตถุประสงค์เพื่อศึกษาอิทธิพลของรูปแบบการเขียนโปรแกรมแบบความร่วมมือระหว่างมนุษย์และปัญญาประดิษฐ์ ที่เรียกว่า "Vibe Coding" ต่อผลการปฏิบัติงานด้านการพัฒนาซอฟต์แวร์และการยอมรับเทคโนโลยีของนักศึกษาระดับปริญญาตรีในสาขาเทคโนโลยีดิจิทัลและนวัตกรรม

การวิจัยครั้งนี้ใช้รูปแบบการวิจัยกึ่งทดลอง โดยมีนักศึกษาจำนวน 40 คน แบ่งออกเป็นกลุ่มทดลองที่ใช้ Vibe Coding ร่วมกับปัญญาประดิษฐ์เชิงสร้างสรรค์ และกลุ่มควบคุมที่ใช้วิธีการเขียนโปรแกรมแบบดั้งเดิม เครื่องมือที่ใช้ในการเก็บรวบรวมข้อมูล ได้แก่ การประเมินผลการปฏิบัติงานจากภารกิจการเขียนโปรแกรม รูบริกประเมินคุณภาพโค้ด และแบบสอบถามตามแบบจำลองการยอมรับเทคโนโลยี (Technology Acceptance Model: TAM) ซึ่งมีค่าสัมประสิทธิ์แอลฟาของครอนบาค (Cronbach's alpha) เท่ากับ .87 สถิติที่ใช้ในการวิเคราะห์ข้อมูล ได้แก่ การทดสอบค่าทีแบบกลุ่มตัวอย่างอิสระ (Independent samples t-test) สหสัมพันธ์เพียร์สัน (Pearson correlation) และค่า Cohen's d

ผลการวิจัยพบว่า กลุ่มทดลองมีประสิทธิภาพในการปฏิบัติภารกิจการเขียนโปรแกรมได้รวดเร็วกว่ากลุ่มควบคุมอย่างมีนัยสำคัญทางสถิติ ($t(58) = 3.45$, $p < .01$, $d = 0.89$) โดยค่าขนาดอิทธิพล (effect size) อยู่ในระดับมาก ตามที่แสดงโดยค่า d เท่ากับ 0.89 ในทางตรงกันข้าม ผลการวิจัยแสดงให้เห็นว่าไม่มีความแตกต่างอย่างมีนัยสำคัญในด้านคุณภาพของโค้ดระหว่างกลุ่มทดลองและกลุ่มควบคุม ($p > .05$) ซึ่งแสดงว่าการนำปัญญาประดิษฐ์มาใช้ในการเขียนโปรแกรมไม่ส่งผลกระทบต่อคุณภาพของโค้ดแต่อย่างใด นอกจากนี้ ผลการวิจัยยังพบว่ามีความสัมพันธ์ในระดับปานกลางระหว่างการรับรู้ถึงประโยชน์ของปัญญาประดิษฐ์และความตั้งใจเชิงพฤติกรรมในการใช้งานในอนาคต ($r = .52$, $p < .01$)

การศึกษาครั้งนี้ให้หลักฐานเชิงประจักษ์ว่าการเขียนโปรแกรมร่วมกันระหว่างมนุษย์และปัญญาประดิษฐ์เป็นประโยชน์ต่อประสิทธิภาพของกระบวนการเขียนโปรแกรม โดยไม่ส่งผลเสียต่อคุณภาพของงานแต่อย่างใด

***คำหลัก**: การเขียนโปรแกรมเชิงสนทนา, ปัญญาประดิษฐ์เชิงสร้างสรรค์, การทำงานร่วมกันระหว่างมนุษย์กับปัญญาประดิษฐ์*

Abstract

While Generative Artificial Intelligence (AI) technology has gained popularity in software development, empirical research on its pedagogical influence on higher education settings, especially within the context of Thailand, has been limited. This current study seeks to investigate the influence of a collaborative programming paradigm between humans and AI, known as “Vibe Coding,” on the performance of software development and technology acceptance for undergraduate Digital Technology and Innovation students.

A quasi-experimental design with 40 students allocated into the experimental group that used Vibe Coding with Generative AI and the control group that used the traditional programming approach. Data collection tools used were the assessment of the students’ performance from the programming tasks, code quality rubric, and Technology Acceptance Model (TAM) survey instrument with Cronbach’s alpha value of .87. Independent samples t-test, Pearson correlation, and Cohen’s d were used as statistical tools.

The results showed that the performance of the experimental group was faster compared to the control group in the programming task, and the difference was statistically significant ($t(58) = 3.45$, $p < .01$, $d = 0.89$). The effect size was large, as shown by the value of d, which was 0.89. On the other hand, the results showed that there was no significant difference in the quality of the code in the experimental and control groups ($p > .05$); thus, the application of AI in programming does not affect the quality of the code. Moreover, the results showed that there was a moderate correlation between the usefulness of AI and the behavioral intention of using AI in the future ($r = .52$, $p < .01$).

This study provides empirical evidence that Human-AI co-programming is beneficial to the efficiency of the programming process without compromising the quality.

***Keywords**: Vide Coding, Generative Artificial Intelligence, Human-AI Collaboration*

1. บทนำ

การพัฒนาเทคโนโลยีปัญญาประดิษฐ์ประเภท Generative AI ได้เปลี่ยนแปลงกระบวนการพัฒนาซอฟต์แวร์อย่างมีนัยสำคัญ โดยเฉพาะการเกิดขึ้นของเครื่องมือที่สามารถสร้างและปรับปรุงโค้ดอัตโนมัติ เช่น Lovable, Replit, Bolt และ Manus เป็นต้น ส่งผลให้รูปแบบการทำงานของนักพัฒนาเปลี่ยนจากการเขียนโค้ดด้วยตนเองทั้งหมดไปสู่การทำงานร่วมกับระบบ AI ในลักษณะ Human–AI Collaboration

แม้จะมีงานวิจัยในต่างประเทศที่ศึกษาประสิทธิภาพของ AI-assisted programming เพิ่มขึ้นอย่างต่อเนื่อง แต่ในบริบทการศึกษาระดับอุดมศึกษาใน

ประเทศไทย ยังขาดหลักฐานเชิงประจักษ์ที่วิเคราะห์ผลกระทบอย่างเป็นระบบ โดยเฉพาะในมิติของประสิทธิภาพการพัฒนา คุณภาพผลงาน และการยอมรับเทคโนโลยีของผู้เรียน

นอกจากนี้ แนวคิด “Vibe Coding” ซึ่งสะท้อนรูปแบบการทำงานแบบร่วมสร้าง (co-creation) ระหว่างมนุษย์และ AI ยังไม่ได้รับการนิยามและทดสอบเชิงประจักษ์อย่างชัดเจนในบริบทการศึกษาไทย ส่งผลให้เกิดช่องว่างทางวิชาการ (Research Gap) ดังนี้:

1. ขาดการศึกษากึ่งทดลองที่เปรียบเทียบผลลัพธ์เชิงประสิทธิภาพ

2. ขาดการวิเคราะห์ผลกระทบต่อคุณภาพโค้ด

3. ขาดการเชื่อมโยงกับกรอบแนวคิดการยอมรับเทคโนโลยี

ดังนั้น งานวิจัยนี้จึงมุ่งศึกษาผลกระทบของแนวทาง Vibe Coding ภายใต้บริบท Generative AI ต่อประสิทธิภาพการพัฒนาซอฟต์แวร์ คุณภาพโค้ด และการยอมรับเทคโนโลยีของนักศึกษาด้านเทคโนโลยีดิจิทัลและนวัตกรรม

2. วัตถุประสงค์งานวิจัย

1. เพื่อเปรียบเทียบประสิทธิภาพการพัฒนาซอฟต์แวร์ระหว่างนักศึกษาที่ใช้แนวทางการเขียนโปรแกรมแบบ Vibe Coding ภายใต้บริบทปัญญาประดิษฐ์เชิงสร้างสรรค์กับนักศึกษาที่ใช้วิธีการเขียนโปรแกรมแบบดั้งเดิม

2. เพื่อเปรียบเทียบคุณภาพโค้ดระหว่างนักศึกษาทั้งสองกลุ่ม

3. เพื่อศึกษาความสัมพันธ์ระหว่างการรับรู้ประโยชน์ของปัญญาประดิษฐ์เชิงสร้างสรรค์กับความตั้งใจการใช้งานในอนาคตของนักศึกษา

4. เพื่อประเมินระดับการยอมรับเทคโนโลยีของนักศึกษาที่ใช้แนวทาง Vibe Coding

3. สมมติฐานการวิจัย

1. นักศึกษาที่ใช้แนวทาง Vibe Coding ภายใต้บริบทปัญญาประดิษฐ์เชิงสร้างสรรค์มีประสิทธิภาพการพัฒนาซอฟต์แวร์สูงกว่านักศึกษาที่ใช้วิธีการเขียนโปรแกรมแบบดั้งเดิมอย่างมีนัยสำคัญทางสถิติที่ระดับ .05

2. คุณภาพโค้ดของนักศึกษาที่ใช้แนวทาง Vibe Coding แตกต่างจากนักศึกษาที่ใช้วิธีการเขียนโปรแกรมแบบดั้งเดิมอย่างมีนัยสำคัญทางสถิติที่ระดับ .05

3. การรับรู้ประโยชน์ของปัญญาประดิษฐ์เชิงสร้างสรรค์มีความสัมพันธ์เชิงบวกกับความตั้งใจใช้งานในอนาคตอย่างมีนัยสำคัญทางสถิติที่ระดับ .05

4. นักศึกษากลุ่มทดลองมีระดับการรับรู้ประโยชน์ของเทคโนโลยีในระดับสูง

4. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การเขียนโปรแกรมเชิงสนทนาด้วย Vibe Coding หมายถึงกระบวนทัศน์การเขียนโปรแกรมโดยผู้ใช้สื่อสารเป้าหมายในรูปแบบภาษาธรรมชาติ และปัญญาประดิษฐ์เชิงสร้างสรรค์ (Generative AI: GenAI) จะสร้างโค้ดขึ้นมาแบบวนซ้ำ [1] [2] [3] แนวคิดนี้แตกต่างจากการเขียนโปรแกรมแบบดั้งเดิมในแง่ที่ผู้เรียนไม่จำเป็นต้องมีความรู้ด้านไวยากรณ์การเขียนโปรแกรมเป็นพื้นฐาน แต่เน้นการสื่อสารเจตนาและการประเมินผลลัพธ์เป็นหลัก

ผลของ Vibe Coding ต่อประสิทธิภาพและผลการเรียนรู้ หลักฐานเชิงประจักษ์จากการศึกษาทั้งในกรณีศึกษาและการปฏิบัติจริงพบว่า Vibe Coding สามารถเพิ่มความเร็วในการสร้างต้นแบบซอฟต์แวร์ได้ถึงร้อยละ 60–80 โดยเฉพาะสำหรับแอปพลิเคชันขนาดเล็ก [2] [4] [5] นักเรียนและผู้ใช้ที่ไม่มีประสบการณ์การเขียนโปรแกรมมาก่อนสามารถสร้างแอปพลิเคชันที่ซับซ้อนได้ อาทิ โปรแกรมจำลองการขับรถแบบสามมิติ แม้ว่าจะยังพบความยากลำบากในการแก้ไขข้อผิดพลาดเชิงตรรกะ [6] อย่างไรก็ดี งานวิจัยเชิงประจักษ์พบว่านักพัฒนาที่มี

ประสบการณ์สูงอาจมีประสิทธิภาพลดลงประมาณร้อยละ 19 เมื่อใช้เครื่องมือ Vibe Coding ในระยะแรก เนื่องจากภาระการตรวจสอบและแก้ไขโค้ดที่เพิ่มขึ้น [2]

ในด้านคุณภาพของโค้ดและความปลอดภัย งานวิจัยหลายชิ้นชี้ให้เห็นว่าโค้ดที่สร้างขึ้นโดย GenAI มักมีช่องโหว่ด้านความปลอดภัยและข้อผิดพลาดเชิงตรรกะอยู่บ่อยครั้ง จึงยังคงต้องการการตรวจสอบโดยมนุษย์ [7] ส่วนผลสัมฤทธิ์ทางวิชาการในระยะสั้นนั้นพบว่ามีความเป็นกลางหรือเพิ่มขึ้นเล็กน้อย [8] [9]

ผลต่อการเรียนรู้ แรงจูงใจ และทักษะ การศึกษาเชิงอภิวิเคราะห์ครอบคลุมระดับการศึกษาขั้นพื้นฐานถึงระดับอุดมศึกษาพบว่า GenAI มีผลเชิงบวกต่อผลสัมฤทธิ์ทางการเรียนและแรงจูงใจในระดับปานกลางถึงสูง โดยเฉพาะในระดับอุดมศึกษาและการมีปฏิสัมพันธ์ผ่านข้อความ [10] ในรายวิชาการเขียนโปรแกรม การสนับสนุนด้วย GenAI ส่งผลให้ผลการทดสอบใกล้เคียงกับการเรียนผ่านสื่อวิดีโอ แต่ผู้เรียนรายงานระดับความรู้สึกเป็นอิสระสูงกว่าและรับรู้ความกดดันน้อยกว่า แม้ว่าสื่อวิดีโอจะสนับสนุนความรู้สึกสมรรถนะในตนเองได้ดีกว่า [8] นอกจากนี้ Vibe Coding ยังส่งผลให้ทักษะที่ผู้เรียนต้องพัฒนาเปลี่ยนแปลงไป โดยเน้นทักษะการออกแบบคำสั่ง (prompt engineering) การจัดการบริบท การประเมินโค้ดอย่างรวดเร็ว และการตัดสินใจว่าเมื่อใดควรแก้ไขโค้ดด้วยตนเอง [3] กรณีศึกษากับผู้เรียนภาษาต่างประเทศพบว่าความสำเร็จในการใช้ Vibe Coding ขึ้นอยู่กับนั่งร้านการเรียนรู้เชิงอภิภาษา (meta-languaging scaffolds) ที่ชัดเจน ได้แก่ การออกแบบคำสั่งอย่างมีโครงสร้าง การสะท้อนคิดเกี่ยวกับความเป็นเจ้าของผลงาน และการสร้างแบบจำลองทางความคิดเกี่ยวกับ AI [7]

นัยสำคัญต่อการศึกษาด้านเทคโนโลยี สำหรับผู้เรียนในสาขาเทคโนโลยี Vibe Coding และ GenAI มีศักยภาพในการลดอุปสรรคในการเริ่มต้นสร้างซอฟต์แวร์ อย่างไรก็ตาม มีความเสี่ยงที่ผู้เรียนจะเกิดความเข้าใจเพียงผิวเผิน พึ่งพาโค้ดที่ตนเองไม่เข้าใจ และสูญเสียทักษะการเขียนโปรแกรม หากใช้ GenAI เพื่อผลิตผลงานเพียงอย่างเดียวโดยไม่ใช้เป็นเครื่องมือเพื่อการเรียนรู้ [11]

5. กรอบแนวคิดการวิจัย

5.1 พื้นฐานเชิงทฤษฎี

งานวิจัยนี้พัฒนากรอบแนวคิดบนพื้นฐาน 2 แนวคิดหลัก ได้แก่

1. แนวคิด Human–AI Collaboration ซึ่งมองว่า AI ทำหน้าที่เป็นผู้ช่วยเชิงปัญญา (Cognitive Augmentation) สนับสนุนการทำงานของมนุษย์ แทนที่จะทำหน้าที่แทนทั้งหมด

2. กรอบแนวคิด Technology Acceptance Model (TAM) ซึ่งอธิบายว่าการรับรู้ประโยชน์ของเทคโนโลยีส่งผลต่อความตั้งใจในการใช้งาน

ภายใต้กรอบดังกล่าว แนวทาง “Vibe Coding” ถูกนิยามในงานวิจัยนี้ว่าเป็นรูปแบบการเขียนโปรแกรมที่ผู้เรียนทำงานร่วมกับ Generative AI ในลักษณะการร่วมสร้าง (co-creation) โดยผู้เรียนยังคงมีบทบาทหลักในการตัดสินใจ วิเคราะห์ และปรับปรุงโค้ด

5.2 โมเดลความสัมพันธ์ของตัวแปร

ตัวแปรต้น:

แนวทางการเขียนโปรแกรม (Vibe Coding และ Traditional Programming) และการรับรู้ประโยชน์ของเทคโนโลยี

ตัวแปรตาม:

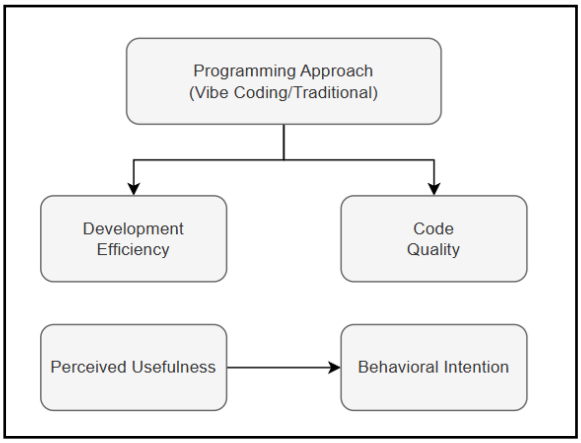
ประสิทธิภาพการพัฒนาซอฟต์แวร์ คุณภาพโค้ด และความตั้งใจใช้งานในอนาคต

ตัวแปรควบคุม:

ความรู้พื้นฐานด้านการเขียนโปรแกรม ประสบการณ์การใช้เครื่องมือปัญญาประดิษฐ์มาก่อน ระดับชั้นปี และความซับซ้อนของงานที่มอบหมาย ทั้งนี้เพื่อป้องกันอิทธิพลแทรกซ้อนที่อาจส่งผลต่อประสิทธิภาพการพัฒนาและคุณภาพโค้ด

5.3. แผนภาพกรอบแนวคิด

รูปที่ 1 แสดงกรอบแนวคิดการวิจัยที่ใช้ในการศึกษาผลของแนวทางการเขียนโปรแกรมแบบ Vibe Coding ต่อประสิทธิภาพในการพัฒนาซอฟต์แวร์และการยอมรับเทคโนโลยี



แผนภาพกรอบแนวคิดนี้แสดงให้เห็นว่าการนำแนวทางการเขียนโปรแกรมแบบ Vibe Coding มาใช้ร่วมกับเทคโนโลยีปัญญาประดิษฐ์สามารถส่งผลต่อผลลัพธ์ด้านประสิทธิภาพและคุณภาพของการพัฒนาซอฟต์แวร์ ขณะเดียวกัน การรับรู้ประโยชน์ของเทคโนโลยียังมีบทบาทสำคัญต่อการยอมรับและความตั้งใจใช้งานเทคโนโลยีในอนาคต ซึ่งเป็นองค์ประกอบสำคัญของการพัฒนาทักษะด้านดิจิทัลและการเรียนรู้เชิงนวัตกรรมของนักศึกษาในยุคที่เทคโนโลยีปัญญาประดิษฐ์เข้ามามีบทบาทในกระบวนการพัฒนาซอฟต์แวร์อย่างต่อเนื่อง

6. วิธีดำเนินการวิจัย

6.1 รูปแบบวิจัย

งานวิจัยนี้ใช้รูปแบบ Quasi-Experimental Research Design แบบ Non-equivalent Control Group Design โดยแบ่งกลุ่มตัวอย่างออกเป็น

- กลุ่มทดลอง (Vibe Coding+Generative AI)
- กลุ่มควบคุม (Traditional Programming)

6.2 ประชากรและกลุ่มตัวอย่าง

นักศึกษาระดับปริญญาตรี สาขาเทคโนโลยีดิจิทัลและนวัตกรรม จำนวน 40 คน การคัดเลือกใช้วิธี Purposive Sampling โดยกำหนดคุณสมบัติ ดังนี้

- ผ่านรายวิชาพื้นฐานการเขียนโปรแกรม
- ไม่เคยอบรมเชิงลึกด้าน Generative AI

6.3 เครื่องมือวิจัย

1. แบบประเมินคุณภาพโค้ด (Code Quality Rubric) เครื่องมือที่ใช้ในการประเมินคุณภาพของโค้ด เป็นแบบ Rubric ที่ผู้วิจัยพัฒนาขึ้น ประกอบด้วยเกณฑ์การประเมิน 4 ด้าน ได้แก่

- Functional Correctness วัดความถูกต้องของโค้ดในการทำงานตามข้อกำหนด Code Structure วัดโครงสร้างและการจัดระเบียบของโค้ด
- Readability วัดความง่ายต่อการอ่านและทำความเข้าใจโค้ด
- Documentation วัดคุณภาพของคำอธิบายประกอบในโค้ด

2. แบบสอบถามการยอมรับเทคโนโลยี (Technology Acceptance Model Questionnaire) แบบสอบถามเพื่อวัดระดับการยอมรับเทคโนโลยีของกลุ่มตัวอย่างที่มีต่อแนวทางการเขียนโปรแกรมแบบ Vibe Coding ร่วมกับเครื่องมือ Generative AI

6.4 สถิติที่ใช้ในการวิเคราะห์

- Mean
- Standard Deviation
- Independent Samples t-test
- Pearson’s Correlation
- Cohen’s d Effect Size

7. ผลการวิจัย

1. ผลการเปรียบเทียบประสิทธิภาพการพัฒนาซอฟต์แวร์ เพื่อศึกษาผลของการใช้แนวทาง Vibe Coding ร่วมกับ Generative AI ต่อประสิทธิภาพในการพัฒนาซอฟต์แวร์ ผู้วิจัยได้ทำการเปรียบเทียบเวลาในการ

พัฒนาซอฟต์แวร์ของนักศึกษาระหว่างกลุ่มทดลองและกลุ่มควบคุม โดยใช้สถิติ Independent Samples t-test

ตารางที่ 1 ค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของเวลาในการพัฒนาซอฟต์แวร์

กลุ่ม	n	Mean (ชั่วโมง)	SD
Vibe Coding	20	2.85	0.72
Traditional Coding	20	3.74	0.81

จากตารางที่ 1 พบว่า กลุ่มทดลองที่ใช้แนวทาง Vibe Coding มีค่าเฉลี่ยเวลาในการพัฒนาซอฟต์แวร์เท่ากับ 2.85 ชั่วโมง ในขณะที่กลุ่มควบคุมมีค่าเฉลี่ยเท่ากับ 3.74 ชั่วโมง และผลการทดสอบด้วย Independent Samples t-test พบว่า เวลาในการพัฒนาซอฟต์แวร์ของทั้งสองกลุ่มแตกต่างกันอย่างมีนัยสำคัญทางสถิติที่ระดับ .01 ($t(58) = 3.45$, $p < .01$) นอกจากนี้ เมื่อคำนวณขนาดอิทธิพลด้วยค่า Cohen's d พบว่ามีค่าเท่ากับ 0.89 ซึ่งจัดอยู่ในระดับ สูง (large effect size) แสดงให้เห็นว่าการใช้แนวทาง Vibe Coding ร่วมกับเครื่องมือ Generative AI สามารถเพิ่มประสิทธิภาพในการพัฒนาซอฟต์แวร์ได้อย่างมีนัยสำคัญ

2. ผลการเปรียบเทียบคุณภาพโค้ด ผู้วิจัยได้ประเมินคุณภาพโค้ดของนักศึกษาทั้งสองกลุ่มโดยใช้เกณฑ์การประเมินที่ประกอบด้วย ความถูกต้องของโปรแกรม โครงสร้างของโค้ด ความอ่านง่ายของโค้ด และการจัดรูปแบบโค้ด

ตารางที่ 2 ผลการเปรียบเทียบคุณภาพโค้ด

กลุ่ม	n	Mean (ชั่วโมง)	SD
Vibe Coding	20	82.40	5.23
Traditional Coding	20	81.75	6.10

จากตารางที่ 2 พบว่า กลุ่มทดลองมีค่าเฉลี่ยคะแนนคุณภาพโค้ดเท่ากับ 82.40 คะแนน ในขณะที่กลุ่มควบคุมมีค่าเฉลี่ยเท่ากับ 81.75 คะแนน และผลการทดสอบด้วย Independent Samples t-test พบว่าคะแนนคุณภาพโค้ดของทั้งสองกลุ่ม ไม่แตกต่างกันอย่างมีนัยสำคัญทางสถิติ ($t(58) = 0.41$, $p > .05$) โดย

จากตารางที่ 2 ผลการวิจัยดังกล่าวสะท้อนให้เห็นว่า แม้ว่าการใช้ Generative AI จะช่วยลดเวลาในการพัฒนาซอฟต์แวร์ แต่ก็ไม่ได้ส่งผลให้คุณภาพของโค้ดลดลงเมื่อเปรียบเทียบกับการเขียนโปรแกรมแบบดั้งเดิม

3. ผลการยอมรับเทคโนโลยี ผู้วิจัยได้ศึกษาความสัมพันธ์ระหว่างตัวแปรตามกรอบแนวคิด Technology Acceptance Model (TAM) เพื่อวิเคราะห์ว่าการรับรู้ประโยชน์ของเทคโนโลยีมีผลต่อความตั้งใจใช้งานในอนาคตหรือไม่

ตารางที่ 3 ค่าสหสัมพันธ์ระหว่างตัวแปรตามกรอบแนวคิด TAM

ตัวแปร	PU	BI
Perceived Usefulness	1	
Behavioral Intention	.52**	1

หมายเหตุ $p < .01$

จากตารางที่ 3 ผลการวิเคราะห์ด้วยสถิติ Pearson's Correlation พบว่า การรับรู้ประโยชน์ของเทคโนโลยีมีความสัมพันธ์เชิงบวกกับความตั้งใจใช้งานในอนาคตในระดับปานกลางอย่างมีนัยสำคัญทางสถิติ ($r = .52$, $p < .01$) ผลดังกล่าวแสดงให้เห็นว่า เมื่อนักศึกษารับรู้ว่าเครื่องมือ Generative AI มีประโยชน์ต่อการพัฒนาซอฟต์แวร์มากขึ้น ก็จะส่งผลให้มีความตั้งใจใช้งานเทคโนโลยีดังกล่าวในอนาคตเพิ่มขึ้น

8. สรุปและอภิปลายผล

ผลการวิจัยครั้งนี้แสดงให้เห็นว่าแนวทางการเขียนโปรแกรมแบบ Vibe Coding สามารถเพิ่มประสิทธิภาพในการพัฒนาซอฟต์แวร์ของนักศึกษาได้อย่างมีนัยสำคัญ โดยนักศึกษาที่ใช้เครื่องมือ Generative AI สามารถพัฒนาโปรแกรมได้เร็วกว่าเมื่อเปรียบเทียบกับนักศึกษาที่ใช้วิธีการเขียนโปรแกรมแบบดั้งเดิม ผลลัพธ์ดังกล่าวสอดคล้องกับงานศึกษาระดับสากลเกี่ยวกับเครื่องมือช่วยเขียนโค้ดด้วย AI เช่น GitHub Copilot ซึ่งได้รับการพัฒนาโดย GitHub และ Microsoft โดยงานวิจัยหลายชิ้นพบว่า นักพัฒนาซอฟต์แวร์ที่ใช้

เครื่องมือ AI-assisted coding สามารถลดเวลาในการพัฒนาโปรแกรมลงได้อย่างมีนัยสำคัญ เนื่องจากระบบ AI สามารถช่วยสร้างโค้ดตัวอย่าง แนะนำโครงสร้างโปรแกรม และช่วยแก้ไขข้อผิดพลาดของโปรแกรมได้อย่างรวดเร็ว

นอกจากนี้ ผลการวิจัยยังสอดคล้องกับการใช้โมเดลภาษาขนาดใหญ่ เช่น ChatGPT ที่สามารถช่วยนักพัฒนาในการอธิบายโค้ด การสร้างอัลกอริทึม และการแก้ไขข้อผิดพลาดของโปรแกรม ซึ่งมีรายงานว่าช่วยเพิ่มประสิทธิภาพในการพัฒนาซอฟต์แวร์และช่วยให้ผู้เรียนเข้าใจแนวคิดการเขียนโปรแกรมได้รวดเร็วยิ่งขึ้น

อย่างไรก็ตาม แม้ว่าการใช้ Generative AI จะช่วยเพิ่มประสิทธิภาพด้านเวลา แต่ผลการวิจัยพบว่าคุณภาพของโค้ดไม่ได้แตกต่างกันอย่างมีนัยสำคัญระหว่างกลุ่มทดลองและกลุ่มควบคุม ผลลัพธ์นี้สะท้อนให้เห็นว่าการใช้ AI ไม่ได้ทำให้คุณภาพของผลงานลดลง แต่ทำหน้าที่เป็นเครื่องมือสนับสนุนกระบวนการพัฒนาซอฟต์แวร์

ในด้านการยอมรับเทคโนโลยี ผลการวิเคราะห์พบว่า การรับรู้ประโยชน์ของเทคโนโลยีมีความสัมพันธ์เชิงบวกกับความตั้งใจใช้งานในอนาคต ซึ่งสอดคล้องกับกรอบแนวคิด Technology Acceptance Model ที่เสนอว่า Perceived Usefulness เป็นปัจจัยสำคัญที่ส่งผลต่อการยอมรับเทคโนโลยีใหม่

ดังนั้น การบูรณาการเครื่องมือ Generative AI ในการเรียนการสอนด้านการพัฒนาซอฟต์แวร์อาจเป็นแนวทางที่ช่วยเพิ่มประสิทธิภาพการเรียนรู้ของนักศึกษาและเตรียมความพร้อมสำหรับการทำงานในอุตสาหกรรมซอฟต์แวร์ในยุคที่เทคโนโลยี AI มีบทบาทสำคัญมากขึ้น

9. ข้อเสนอแนะ

แม้ว่างานวิจัยครั้งนี้จะให้หลักฐานเชิงประจักษ์เกี่ยวกับผลของแนวทาง Vibe Coding ต่อประสิทธิภาพในการพัฒนาซอฟต์แวร์ของนักศึกษา แต่ยังมีข้อจำกัดบางประการที่ควรนำไปพิจารณาในการวิจัยครั้งต่อไป

ประการแรก งานวิจัยนี้ใช้กลุ่มตัวอย่างจำนวน 40 คน ซึ่งเป็นนักศึกษาระดับปริญญาตรีจากสาขาเทคโนโลยีดิจิทัลและนวัตกรรมในสถาบันการศึกษาเพียงแห่งเดียว ดังนั้น ผลการวิจัยอาจมีข้อจำกัดในการนำไปอธิบายหรืออ้างอิงกับกลุ่มตัวอย่างที่มีบริบททางการศึกษาแตกต่างกัน เช่น นักศึกษาจากสาขาวิศวกรรมซอฟต์แวร์หรือวิทยาการคอมพิวเตอร์ในสถาบันอื่น

ประการที่สอง การวิจัยครั้งนี้ใช้รูปแบบการทดลองแบบกึ่งทดลอง (Quasi-experimental design) ซึ่งแม้ว่าจะช่วยให้สามารถศึกษาผลของตัวแปรได้ในบริบทการเรียนการสอนจริง แต่ยังอาจมีปัจจัยแทรกซ้อนอื่นที่ไม่สามารถควบคุมได้ทั้งหมด เช่น ความแตกต่างของประสบการณ์การเขียนโปรแกรมของนักศึกษา หรือระดับความคุ้นเคยกับเครื่องมือ Generative AI

ประการที่สาม การประเมินผลการพัฒนาซอฟต์แวร์ในงานวิจัยนี้มุ่งเน้นที่เวลาในการพัฒนาโปรแกรมและคุณภาพของโค้ดเป็นหลัก ซึ่งอาจยังไม่ครอบคลุมมิติอื่นของประสิทธิภาพการพัฒนาซอฟต์แวร์ เช่น ความสามารถในการแก้ปัญหาที่ซับซ้อน ความคิดสร้างสรรค์ในการออกแบบโปรแกรม หรือความเข้าใจเชิงลึกเกี่ยวกับแนวคิดการเขียนโปรแกรมต่อการเรียนรู้ด้านการเขียนโปรแกรมมากที่สุด

10. เอกสารอ้างอิง

[1] Meske, C., Hermanns, T., Von Der Weiden, E., Loser, K., & Berger, T. (2025). Vibe Coding as a Reconfiguration of Intent Mediation in Software Development: Definition, Implications, and Research Agenda. IEEE Access, 13, 213242-213259.

[2] Bamil, V. (2025). Vibe Coding: Toward an AI-Native Paradigm for Semantic and Intent-Driven Programming. ArXiv, abs/2510.17842.

[3] Sarkar, A., & Drosos, I. (2025). Vibe coding: programming through conversation with artificial intelligence. ArXiv, abs/2506.23253.

[4] Hemdev, N. (2025). VIBECODING:A Mixed-MethodsCase Study on Conversational AIProgramming and ApplicationDevelopment. International Journal on Science and Technology.

[5] Moore, J., & Tatonetti, N. (2025). Vibe coding: a new paradigm for biomedical software development. BioData Mining, 18.

[6] Fortes-Ferreira, M., Alam, M., & Bazilinskyy, P. (2025). Vibe Coding in Practice: Building a Driving Simulator Without Expert Programming Skills. Adjunct Proceedings of the 17th International Conference on Automotive User Interfaces and Interactive

[7] Woo, D., Guo, K., & Yu, Y. (2025). A vibe coding learning design to enhance EFL students' talking to, through, and about AI. ArXiv, abs/2509.08854.

[8] Cubillos, C., Mellado, R., Cabrera-Paniagua, D., & Urra, E. (2025). Generative Artificial Intelligence in Computer Programming: Does It Enhance Learning, Motivation, and the Learning Environment?. IEEE Access, 13, 40438-40455.

[9] Fan, L., Deng, K., & Liu, F. (2025). Educational impacts of generative artificial intelligence on learning and performance of engineering students in China. Scientific Reports, 15.

[10] Liu, X., Guo, B., He, W., & Hu, X. (2025). Effects of Generative Artificial Intelligence on K-12 and Higher Education Students' Learning Outcomes: A Meta-Analysis. Journal of Educational Computing Research, 63, 1249 - 1291.

[11] Geng, F., Shah, A., Li, H., Mulla, N., Swanson, S., Raj, A., Zingaro, D., & Porter, L. (2025). Exploring Student-AI Interactions in Vibe Coding. ArXiv, abs/2507.22614.

ประวัติผู้เขียนบทความ

ผู้ช่วยศาสตราจารย์ชนินทร เฉลิมสุข
อาจารย์สาขาวิชาเทคโนโลยีดิจิทัลและนวัตกรรม
คณะศิลปศาสตร์และวิทยาศาสตร์
มหาวิทยาลัยเอเชียอาคเนย์

ผู้ช่วยศาสตราจารย์ศิลป์ณรงค์ ฉวีพัฒน์
อาจารย์สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
คณะวิทยาการและเทคโนโลยี
มหาวิทยาลัยราชภัฏกำแพงเพชร

นายธนบดี ศรีธนนันท์
อาจารย์สาขาวิชาเทคโนโลยีดิจิทัลและนวัตกรรม
คณะศิลปศาสตร์และวิทยาศาสตร์
มหาวิทยาลัยเอเชียอาคเนย์