

การประยุกต์ใช้แนวคิด Agile จากอุตสาหกรรมซอฟต์แวร์สู่ห้องเรียน ในบริบทของการศึกษาระดับอุดมศึกษา

Applying Agile Principles from the Software Industry to the Classroom A Case within the Context of Higher Education

ดนุพล วันชัยสถิ¹, วชรพงษ์ ณ เชียงใหม่², นิลาวรรณ วงษ์ศิลปมรกด³, Chi Keong Goh⁴

¹คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยนอร์ท-เชียงใหม่, danuphon@gmail.com

²คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยนอร์ท-เชียงใหม่, wacharapong@northcm.ac.th

³คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยนอร์ท-เชียงใหม่, nilawan@northcm.ac.th

⁴คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยนอร์ท-เชียงใหม่, meitikor@gmail.com

บทคัดย่อ

บทความวิชาการนี้มีวัตถุประสงค์เพื่อเสนอแนวทางการประยุกต์ใช้แนวคิด Agile ซึ่งเป็นพื้นฐานสำคัญในกระบวนการผลิตซอฟต์แวร์โดยเป็นการนำแนวคิดไปประยุกต์ใช้ในการจัดการเรียนการสอนในบริบทของการศึกษาระดับอุดมศึกษา โดยได้นำองค์ประกอบหลักของ Agile มาใช้ ได้แก่ Agile Manifesto หลักการ 12 ข้อ และแนวทางของ Agile Methodology ได้แก่ Scrum, Kanban, XP และ Lean Software Development โดยนำเสนอการประยุกต์ใช้แนวคิดดังกล่าวเพื่อออกแบบกระบวนการจัดการเรียนการสอน ทั้งในรายวิชาทางทฤษฎีและรายวิชาที่เน้นทักษะเชิงปฏิบัติ และแนวทางเชิงวิเคราะห์เพื่อแสดงให้เห็นถึงความเป็นไปได้ในการประยุกต์ใช้แนวคิด Agile ในการออกแบบกิจกรรมการเรียนรู้ การบริหารจัดการชั้นเรียน และการส่งเสริมการเรียนรู้แบบมีปฏิสัมพันธ์ ซึ่งจะช่วยเพิ่มความคล่องตัวและประสิทธิภาพของกระบวนการเรียนรู้ ข้อเสนอแนะชี้ให้เห็นว่า แนวคิด Agile ช่วยเสริมสร้างความต่อเนื่อง ความยืดหยุ่น และการมีส่วนร่วมของผู้เรียนได้เป็นอย่างดี อย่างไรก็ตามการประยุกต์ใช้อย่างมีประสิทธิภาพจำเป็นต้องอาศัยความเข้าใจในหลักการของ Agile อย่างเพียงพอจากผู้สอนและนักออกแบบการเรียนการสอน เพื่อให้สามารถนำไปใช้ได้อย่างเหมาะสม และสอดคล้องกับเป้าหมายของการจัดการเรียนการสอนและบริบทของผู้เรียน

คำหลัก: อัจฉริยะ การจัดการเรียนการสอน แนวทางการเรียนรู้แบบยืดหยุ่น

Abstract

This academic article aims to propose a framework for applying the Agile methodology basic fundamental in software development processes to instructional design within the context of higher education. The article integrates key values of Agile, including the Agile Manifesto, 12

principles, and Agile methodologies such as Scrum, Kanban, Extreme Programming (XP), and Lean Software Development. It presents a practical application of these concepts in designing instructional processes across both theoretical and practice-oriented courses. The article provides an analytical perspective demonstrating the feasibility of using Agile in the development of learning activities, classroom management, and the promotion of interactive learning. Such an approach enhances the flexibility and responsiveness of the learning process. The findings suggest that Agile principles can foster continuity, adaptability, and active learner engagement. However, effective implementation requires instructors and instructional designers to possess sufficient understanding of Agile foundations to ensure alignment with the pedagogical goals and learner context.

Keywords: Agile, Learning and Teaching Frameworks, Flexible Learning Approaches

บทนำ

ในยุคที่เทคโนโลยีสารสนเทศมีการเปลี่ยนแปลงอย่างรวดเร็วและต่อเนื่อง องค์กรต่างๆ โดยเฉพาะในอุตสาหกรรมซอฟต์แวร์ จำเป็นต้องปรับตัวเพื่อให้สามารถตอบสนองต่อความต้องการของผู้ใช้งานที่หลากหลาย และเปลี่ยนแปลงอยู่ตลอดเวลา กระบวนการพัฒนาซอฟต์แวร์จึงไม่อาจยึดติดกับแนวทางแบบเดิมที่มีลักษณะเป็นขั้นตอนตายตัว เช่น Waterfall ได้อีกต่อไป เนื่องจากรูปแบบดังกล่าวมักไม่สามารถปรับเปลี่ยนแผนงานหรือโครงสร้างได้ทันต่อสถานการณ์ ส่งผลให้โครงการเกิดความล่าช้า ไม่สามารถส่งมอบคุณค่าได้อย่างแท้จริง และอาจสิ้นเปลืองทรัพยากรโดยไม่จำเป็น

จากการเผชิญกับความเปลี่ยนแปลงอย่างรวดเร็วและความไม่แน่นอนของสภาพแวดล้อมในการทำงาน จึงนำไปสู่การพัฒนาแนวคิดที่เรียกว่า Agile ซึ่งมุ่งเน้นการทำงานร่วมกัน การสื่อสารที่มีประสิทธิภาพ การส่งมอบงานอย่างต่อเนื่อง และการตอบสนองต่อการเปลี่ยนแปลงได้อย่างรวดเร็ว แนวคิด Agile ถือกำเนิดขึ้นในปี ค.ศ. 2001 จากการรวมตัวของนักพัฒนาซอฟต์แวร์ 17 คน ที่ร่วมกันร่างเอกสารสำคัญชื่อว่า Agile Manifesto ซึ่งประกอบด้วยอุดมการณ์หลัก 4 ข้อ (Beck และคณะ, 2001) ได้แก่

- 1) Individuals and Interactions over Processes and Tools
- 2) Working Software over Comprehensive Documentation
- 3) Customer Collaboration over Contract Negotiation
- 4) Responding to Change over Following a Plan

ปัจจุบันแนวคิด Agile ได้กลายเป็นกรอบแนวทางหลักในอุตสาหกรรมซอฟต์แวร์ และถูกนำไปปรับใช้ในหลายองค์กร ไม่เพียงแต่ลดความเสี่ยงของโครงการแต่ยังช่วยเพิ่มคุณภาพของผลิตภัณฑ์ ส่งเสริมการสื่อสารในทีม

และสร้างวัฒนธรรมองค์กรที่โปร่งใสและเน้นผู้ใช้งานเป็นศูนย์กลาง แนวคิด Agile ไม่ได้เป็นเพียงแนวทางในการจัดการโครงการ แต่เป็นวิธีคิดที่สามารถนำไปประยุกต์ใช้ในหลายบริบท ทั้งในเชิงธุรกิจ การบริการสาธารณะ และโดยเฉพาะอย่างยิ่งในภาคการศึกษา (Schwaber & Sutherland, 2017)

ในช่วงทศวรรษที่ผ่านมา แนวทางการศึกษาเริ่มให้ความสนใจในการนำแนวคิด Agile มาประยุกต์ใช้ในการจัดการเรียนการสอน โดยเฉพาะในบริบทการเรียนรู้ที่เน้นผู้เรียนเป็นศูนย์กลาง การส่งเสริมการเรียนรู้แบบ Active Learning และการพัฒนาทักษะแห่งศตวรรษที่ 21 แนวทาง Agile ซึ่งให้ความสำคัญกับความยืดหยุ่น การมีส่วนร่วม การสะท้อนผล และการพัฒนาอย่างต่อเนื่อง จึงสอดคล้องกับแนวทางจัดการเรียนการสอนยุคใหม่อย่างมีนัยสำคัญ (สมพร อิศเรนา ณ อยุธยา และคณะ, 2561) การนำ Agile มาปรับใช้ในการออกแบบกระบวนการเรียนรู้ ช่วยสร้างสภาพแวดล้อมที่เปิดโอกาสให้ผู้เรียนได้ลงมือปฏิบัติจริง มีส่วนร่วมในกระบวนการเรียนรู้ทุกขั้นตอน และสามารถปรับเปลี่ยนทิศทางการเรียนรู้ตามผลลัพธ์ที่เกิดขึ้น ส่งเสริมการทำงานกลุ่ม การประเมินผลอย่างต่อเนื่อง และการเรียนรู้จากการสะท้อนผลจากผลลัพธ์การเรียนรู้ ซึ่งล้วนเป็นปัจจัยสำคัญที่นำไปสู่การพัฒนาทักษะเชิงกระบวนการและความเข้าใจที่ลึกซึ้งของผู้เรียน

บทความนี้มีวัตถุประสงค์เพื่อเสนอแนวทางการประยุกต์ใช้แนวคิด Agile ซึ่งมีจุดเด่นด้านความยืดหยุ่น การทำงานแบบเป็นรอบ และการปรับปรุงอย่างต่อเนื่อง เข้ากับการจัดการเรียนรู้ที่เน้นผู้เรียนเป็นศูนย์กลาง โดยมุ่งส่งเสริมให้ผู้เรียนมีส่วนร่วมในกระบวนการเรียนรู้ สามารถปรับเปลี่ยนกิจกรรมตามบริบทและผลลัพธ์ที่เกิดขึ้นจริงซึ่งเป็นบริบทที่สำคัญของการศึกษาระดับอุดมศึกษา พร้อมทั้งพัฒนาทักษะด้านการคิดวิเคราะห์ การทำงานร่วมกัน และการเรียนรู้ด้วยตนเองทั้งในรายวิชาทางทฤษฎีและรายวิชาที่เน้นทักษะเชิงปฏิบัติ นอกจากนี้ระบบการศึกษาระดับอุดมศึกษาในปัจจุบันยังประสบกับปัญหาเฉพาะหลายประการ เช่น การมีส่วนร่วมของผู้เรียนที่อยู่ในระดับต่ำภายใต้รูปแบบการเรียนการสอนแบบดั้งเดิม การประเมินผลที่ไม่ส่งเสริมการคิดวิเคราะห์เชิงลึก ตลอดจนความไม่สอดคล้องของหลักสูตรกับบริบทและความเปลี่ยนแปลงของโลกในอนาคต แนวคิด Agile จึงสามารถเข้ามาแก้ปัญหาในการรับมือกับปัญหาเหล่านี้ได้อย่างมีประสิทธิภาพ ผ่านการออกแบบกิจกรรมการเรียนรู้ที่มีความยืดหยุ่น ปรับเปลี่ยนได้ตามสถานการณ์ และเน้นการมีส่วนร่วมของผู้เรียนอย่างแท้จริง ซึ่งช่วยส่งเสริมการเรียนรู้เชิงรุกและพัฒนาทักษะสำคัญที่จำเป็นในศตวรรษที่ 21

แนวคิดและความสำคัญของ Agile

แนวคิดของ Agile มีจุดเริ่มต้นจากประสบการณ์ตรงของผู้พัฒนาซอฟต์แวร์ที่ต้องเผชิญกับปัญหาในกระบวนการพัฒนาแบบดั้งเดิมที่มีความตายตัว เช่น Waterfall ซึ่งมีการกำหนดลำดับขั้นตอนที่ชัดเจน ไม่สามารถปรับเปลี่ยนได้ง่าย เช่น การวางแผน การวิเคราะห์และออกแบบระบบ การพัฒนา และการทดสอบระบบ ที่ต้องทำตามลำดับเท่านั้น หากเกิดการเปลี่ยนแปลงในระหว่างกระบวนการ ต้องย้อนกลับไปแก้ไขในขั้นตอนก่อนหน้า ซึ่งใช้เวลาเพิ่มขึ้นและสิ้นเปลืองทรัพยากร จากปัญหานี้ นักพัฒนาซอฟต์แวร์ได้แลกเปลี่ยนประสบการณ์และ

หาวิธีพัฒนาแนวคิดใหม่ในการจัดการโครงการซอฟต์แวร์ โดยได้ร่วมกันคิดค้นแนวคิด ที่รู้จักกันในชื่อว่า Agile Manifesto ประกอบด้วยอุดมการณ์ 4 ข้อ ได้แก่

1. ให้ความสำคัญกับคนและการสื่อสารมากกว่าขั้นตอนหรือเครื่องมือ
2. เน้นซอฟต์แวร์ที่ใช้งานได้จริงมากกว่าการจัดทำเอกสารที่ละเอียดสมบูรณ์
3. มุ่งให้ลูกค้ามีส่วนร่วมมากกว่าการยึดติดกับสัญญา
4. พร้อมรับการเปลี่ยนแปลงมากกว่าการเดินตามแผนอย่างเคร่งครัด

นอกจากอุดมการณ์หลักทั้ง 4 ข้อ ยังมีหลักการของ Agile อีก 12 ข้อ ที่สนับสนุนการทำงานซึ่งเน้นความยืดหยุ่น การมีส่วนร่วม และการพัฒนาที่ยั่งยืน ดังนี้

1. ความพึงพอใจของลูกค้าเป็นสิ่งสำคัญอันดับแรกโดยการส่งมอบซอฟต์แวร์ที่ใช้งานได้จริงในช่วงเวลาสั้น ๆ อย่างต่อเนื่อง เพื่อให้ลูกค้าเห็นความก้าวหน้าและสามารถใช้งานได้ทันที
2. ยอมรับการเปลี่ยนแปลงแม้ในช่วงท้ายของการพัฒนาโดยมองว่าการเปลี่ยนแปลงคือโอกาส ไม่ใช่ปัญหา และควรมีความยืดหยุ่นในการปรับตัวเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด
3. ส่งมอบซอฟต์แวร์ที่ใช้งานได้อย่างสม่ำเสมอโดยกำหนดรอบเวลาที่แน่นอนเพื่อให้เกิดความต่อเนื่องในการพัฒนาและการประเมินผล
4. ผู้มีส่วนเกี่ยวข้องกับผู้ธุรกิจและนักพัฒนาควรร่วมมือกันอย่างใกล้ชิดเพื่อให้เข้าใจเป้าหมายร่วมกัน และสามารถปรับกระบวนการได้อย่างตรงจุดและร่วมกันแก้ไขปัญหา
5. สร้างแรงจูงใจและสนับสนุนบุคลากรในทีมโดยเชื่อมั่นในความสามารถของทีมและให้เครื่องมือหรือสภาพแวดล้อมที่เอื้อต่อความสำเร็จ
6. การสื่อสารแบบพบปะโดยตรงเป็นวิธีที่มีประสิทธิภาพที่สุด เพราะช่วยลดความเข้าใจผิดและเพิ่มประสิทธิภาพในการตัดสินใจ
7. ซอฟต์แวร์ที่ใช้งานได้จริงคือตัวชี้วัดสำคัญของความก้าวหน้า ไม่ใช่เพียงรายงานหรือเอกสาร
8. รักษาระยะเวลาในการพัฒนาอย่างยั่งยืน ทั้งในมุมมองของนักพัฒนาและผู้ใช้ เพื่อไม่ให้เกิดความเหนื่อยล้าหรือคุณภาพลดลงในระยะยาว
9. ให้ความสำคัญกับคุณภาพทางเทคนิคและการออกแบบที่ดี เพื่อให้ซอฟต์แวร์สามารถดูแลรักษาและพัฒนาเพิ่มเติมได้ง่าย
10. ความเรียบง่ายเป็นสิ่งสำคัญ โดยเน้นที่การลดงานที่ไม่จำเป็นและเลือกวิธีการที่ตรงเป้าหมายมากที่สุด
11. ทีมที่จัดการตัวเองได้จะมีประสิทธิภาพมากที่สุด เพราะสามารถตัดสินใจได้รวดเร็วและรับผิดชอบต่อผลลัพธ์ร่วมกัน
12. ทีมควรสะท้อนผลและปรับปรุงกระบวนการอย่างสม่ำเสมอ เพื่อเรียนรู้จากประสบการณ์และพัฒนาปรับปรุง ให้ดีขึ้นในทุกๆ รอบการทำงาน

ตาราง 1 แสดงประเด็นเปรียบเทียบกระบวนการผลิตซอฟต์แวร์ ระหว่าง Agile และ Waterfall

ประเด็นเปรียบเทียบ	Agile	Waterfall Model
ลำดับขั้นการพัฒนา	มีลักษณะยืดหยุ่น สามารถปรับเปลี่ยนขั้นตอนตามสถานการณ์หรือข้อมูลใหม่ที่เกิดขึ้นระหว่างการดำเนินงาน	ดำเนินการตามลำดับขั้นที่วางไว้ล่วงหน้าอย่างเคร่งครัด ไม่สามารถย้อนกลับได้ง่าย
การส่งมอบงาน	ส่งมอบผลงานเป็นระยะอย่างต่อเนื่องในลักษณะรอบสั้น (Iteration) เพื่อรับข้อเสนอแนะและปรับปรุงทันที	ส่งมอบผลงานเมื่อสิ้นสุดโครงการ โดยไม่มีการส่งมอบคุณระหว่างทาง
การมีส่วนร่วมของลูกค้า	ส่งเสริมการมีส่วนร่วมของผู้ใช้หรือผู้มีส่วนได้ส่วนเสียอย่างต่อเนื่องตลอดกระบวนการ	มีการติดต่อเฉพาะช่วงต้นโครงการ (เก็บความต้องการ) และช่วงท้าย (ส่งมอบงาน)
การตอบสนองต่อการเปลี่ยนแปลง	สามารถปรับเปลี่ยนแผนงานหรือคุณสมบัติของระบบได้อย่างรวดเร็วตามข้อเสนอแนะหรือสถานการณ์ใหม่	การเปลี่ยนแปลงทำได้ยากและมีต้นทุนสูง เนื่องจากมีการกำหนดแผนงานไว้ล่วงหน้าอย่างแน่นอน
การทดสอบระบบ	มีการทดสอบควบคู่ไปกับกระบวนการพัฒนา เพื่อให้สามารถตรวจพบและแก้ไขข้อผิดพลาดได้ทันที	ทำการทดสอบเมื่อการพัฒนาเสร็จสมบูรณ์เท่านั้น ซึ่งอาจทำให้ข้อผิดพลาดสะสมและแก้ไขได้ยาก

จากตาราง 1 การเลือกใช้แนวทางการพัฒนาระบบซอฟต์แวร์มีผลอย่างมากต่อประสิทธิภาพของโครงการ โดยเฉพาะสองแนวทางหลักที่ได้รับความนิยม ได้แก่ Agile และ Waterfall ซึ่งมีลักษณะและปรัชญาการทำงานที่แตกต่างกันอย่างชัดเจน แนวทาง Waterfall เน้นความเป็นขั้นตอนที่แน่นอนและดำเนินการตามลำดับ ตั้งแต่การวิเคราะห์ความต้องการ ไปจนถึงการออกแบบ การพัฒนาระบบ และ การทดสอบระบบ ซึ่งข้อดีของ Waterfall คือความมีระเบียบแบบแผน แต่ข้อจำกัดสำคัญคือการขาดความยืดหยุ่น โดยเฉพาะอย่างยิ่งเมื่อต้องเผชิญกับการเปลี่ยนแปลงที่เกิดขึ้นในระหว่างดำเนินโครงการ (Serrador & Pinto, 2015)

ในทางตรงกันข้าม Agile เป็นแนวทางที่ให้ความสำคัญกับการปรับตัว การแบ่งงานเป็นรอบเล็กๆ และการส่งมอบงานอย่างต่อเนื่อง รวมถึงลูกค้าและผู้มีส่วนร่วมจะมีบทบาทร่วมกันตลอดกระบวนการพัฒนา ซึ่งช่วยให้ทีมสามารถตอบสนองต่อความเปลี่ยนแปลงได้อย่างรวดเร็ว ด้านการทดสอบระบบจะเกิดขึ้นควบคู่กับการพัฒนาระบบ จึงช่วยลดความผิดพลาดและเพิ่มคุณภาพของผลิตภัณฑ์ได้ดียิ่งขึ้น (Shastri, Hoda, & Amor, 2021) โดยมีข้อแตกต่างอย่างเด่นชัดในหลายด้าน เช่น ลำดับขั้นการพัฒนา การมีส่วนร่วมของลูกค้า และกระบวนการทดสอบระบบ ซึ่งชี้ให้เห็นว่า Agile มีความเหมาะสมกับโครงการที่มีความไม่แน่นอนสูง และมีการปรับเปลี่ยนบ่อยครั้ง ขณะที่ Waterfall เหมาะกับโครงการที่ข้อกำหนดชัดเจนตั้งแต่ต้น และไม่เปลี่ยนแปลงมากนัก

Agile ไม่ได้เป็นเพียงกรอบแนวคิดหรือทฤษฎี แต่เป็นแนวทางการทำงานที่มีพื้นฐานจากอุดมการณ์ทั้ง 4 ข้อ และหลักการ 12 ข้อ โดยแนวคิดเหล่านี้ถูกนำมาประยุกต์ใช้อย่างเป็นระบบตามแนวปฏิบัติที่เรียกว่า Agile Methodology ซึ่งได้รับการออกแบบให้สอดคล้องกับบริบทการทำงานที่หลากหลาย ตัวอย่างเช่น

1. Scrum ถือเป็นรูปแบบการดำเนินงานที่ได้รับความนิยมอย่างแพร่หลายในกระบวนการพัฒนาซอฟต์แวร์ โดยมีลักษณะการทำงานแบบวนซ้ำในรอบสั้นๆที่เรียกว่า Sprint ซึ่งแต่ละรอบมักมีระยะเวลา 2-4 สัปดาห์ ภายในกระบวนการจะมีบทบาทสำคัญที่ชัดเจน ได้แก่ ผู้รับผิดชอบผลิตภัณฑ์ (Product Owner), ผู้นำกระบวนการ (Scrum Master) และทีมปฏิบัติงาน (Development Team) โดยมีกระบวนการหลักประกอบไปด้วย การวางแผนการทำงาน การประชุมติดตามความก้าวหน้าแบบรายวัน การทบทวนผลงานในแต่ละวัน และการสะท้อนผลเพื่อการปรับปรุงโครงการ ทั้งนี้ จุดเด่นของการส่งเสริมการประเมินผลและปรับปรุงกระบวนการอย่างต่อเนื่องในแต่ละรอบการดำเนินงาน (Schwaber & Sutherland, 2017)

2. Kanban ให้ความสำคัญกับการบริหารจัดการงานอย่างเป็นระบบ โดยใช้เครื่องมือในการมองเห็นกระบวนการทำงานผ่านบอร์ดแสดงสถานะของงาน ซึ่งแบ่งออกเป็นขั้นตอน เช่น รอดำเนินการ กำลังดำเนินการ และเสร็จสิ้น ลักษณะเด่น คือการจำกัดจำนวนงานที่อยู่ในระหว่างดำเนินการในแต่ละช่วงเวลา (Work in Progress Limit) เพื่อเพิ่มความคล่องตัว ลดความซับซ้อน และป้องกันความแออัดของภาระงาน ทั้งยังส่งเสริมวัฒนธรรมการทำงานที่โปร่งใสและตรวจสอบได้ (Ahmad, Markkula, & Oivo, 2013)

3. Extreme Programming (XP) เป็นแนวทางการพัฒนาซอฟต์แวร์ที่มุ่งเน้นการยกระดับคุณภาพของการเขียนโปรแกรม และการตอบสนองอย่างรวดเร็วต่อความต้องการที่เปลี่ยนแปลงอยู่เสมอ โดยเน้นการทำงานร่วมกันอย่างใกล้ชิดของทีมพัฒนา และการทดสอบอย่างต่อเนื่อง เทคนิคหลักที่สำคัญ ได้แก่ การเขียนโปรแกรมแบบจับคู่ (Pair Programming) การพัฒนาที่ขับเคลื่อนด้วยการทดสอบ (Test-Driven Development: TDD) การรวมโปรแกรมอย่างต่อเนื่อง (Continuous Integration) และการปรับปรุงโครงสร้างภายในของโปรแกรมโดยไม่เปลี่ยนพฤติกรรมภายนอก (Refactoring)

4. Lean Software Development มุ่งเน้นการลดขั้นตอนหรือกระบวนการที่ไม่จำเป็น เพื่อเพิ่มคุณค่า แนวทางนี้ให้ความสำคัญกับการทำงานอย่างมีประสิทธิภาพ การส่งมอบผลงานอย่างรวดเร็ว และการส่งเสริมหักยภาพของทีม โดยยึดหลักการสำคัญ เช่น การขจัดสิ่งที่ไม่จำเป็น (Eliminate Waste) การสร้างคุณภาพตั้งแต่ต้นทาง (Build Quality In) การส่งมอบอย่างรวดเร็ว (Deliver Fast) การเพิ่มอำนาจในการตัดสินใจให้กับทีม (Empower the Team) และการปรับปรุงระบบภาพรวมให้ดีที่สุด (Optimize the Whole)

การประยุกต์ใช้ในระบบการศึกษาศึกษา

หลักการทั้ง 12 ข้อของ Agile ไม่ได้จำกัดอยู่เพียงในแวดวงอุตสาหกรรมซอฟต์แวร์ แต่ยังสามารถประยุกต์ใช้ในระบบของการศึกษาศึกษาได้อย่างกว้างขวาง โดยเฉพาะในยุคที่การเรียนรู้จำเป็นต้องมีความยืดหยุ่น เน้นผู้เรียนเป็นศูนย์กลาง และส่งเสริมการพัฒนาทักษะแห่งศตวรรษที่ 21 โดยการวิเคราะห์ถึงแนวทางการประยุกต์ใช้หลักการทั้ง 12 ข้อ สำหรับการเรียนการสอน ในรายวิชาที่เน้นความรู้ทางทฤษฎี/เชิงเนื้อหา และวิชาที่เน้นทักษะเชิงปฏิบัติ แสดงได้ในตารางต่อไปนี้

ตาราง 2 การเปรียบเทียบการประยุกต์ใช้หลักการ Agile Manifesto ในรายวิชาทางทฤษฎีและปฏิบัติการ

หลักการ	รายวิชาทางทฤษฎี/เชิงเนื้อหา	รายวิชาทางปฏิบัติ/เชิงทักษะ
1. การส่งมอบคุณค่าอย่างต่อเนื่อง	มอบหมายงานย่อยเป็นระยะ เช่น บทความหรือรายงานย่อย เพื่อสะท้อนความเข้าใจอย่างต่อเนื่อง	แบ่งกระบวนการทำงานออกเป็นช่วง ส่งมอบชิ้นงานจริงหรือผลลัพธ์การปฏิบัติงานเป็นลำดับขั้น
2. ยอมรับการเปลี่ยนแปลงแม้ในช่วงท้าย	เปิดโอกาสให้ผู้เรียนปรับหัวข้อหรือมุมมองในการศึกษาได้เมื่อได้รับข้อมูลใหม่	ปรับแผนการทำงานหรือกลยุทธ์ตามข้อเสนอแนะหรือสถานการณ์การปฏิบัติงานที่เปลี่ยนไป
3. ส่งมอบงานที่และสั้น	มอบหมายแบบฝึกหัดสั้น ๆ อย่างสม่ำเสมอเพื่อกระตุ้นการคิด วิเคราะห์ และเขียน	ดำเนินงานเป็นรอบสั้น (Sprint) พร้อมการสรุปและส่งมอบงาน/ชิ้นงาน ทุกสัปดาห์
4. การทำงานร่วมกันระหว่างผู้มีส่วนเกี่ยวข้อง	สร้างกิจกรรมอภิปราย แลกเปลี่ยนความคิดเห็นระหว่างผู้เรียนกับผู้สอน	จัดเวิร์กช็อปที่ให้ผู้เรียนมีส่วนร่วม วางแผนและดำเนินโครงการร่วมกับผู้สอนหรือเพื่อนร่วมทีม
5. สร้างแรงบันดาลใจและความไว้วางใจ	ส่งเสริมให้ผู้เรียนเลือกหัวข้อที่สอดคล้องกับความสนใจส่วนบุคคล	สนับสนุนบทบาทผู้นำหรือผู้จัดการภายในกลุ่มเพื่อสร้างความรับผิดชอบและแรงจูงใจภายในทีม
6. การสื่อสารแบบพบปะโดยตรง	ใช้การอภิปราย หรือการเรียนรู้เพื่อส่งเสริมการสื่อสารเชิงลึก	จัดประชุมทีมแบบสั้นประจำวัน (Stand-up Meeting) เพื่อแก้ปัญหาและรายงานสถานะงาน
7. วัดผลจากผลลัพธ์ใช้งานได้	ประเมินความเข้าใจจากการวิเคราะห์เชิงลึก การนำเสนอ หรือข้อสอบสะท้อนแนวคิด	ประเมินจากผลงานจริง เช่น โครงการ ชิ้นงาน แอปพลิเคชัน หรือผลงานออกแบบ
8. การพัฒนาอย่างยั่งยืน	กระจายภาระงานตลอดภาคเรียน ไม่เร่งงานช่วงท้าย เพื่อสร้างความสมดุลในการเรียนรู้	วางแผนงานให้มีระยะเวลาปฏิบัติงานอย่างเหมาะสมต่อเนื่อง ไม่ก่อให้เกิดความเครียดหรือเร่งรีบเกินไป

หลักการ	รายวิชาทางทฤษฎี/เชิงเนื้อหา	รายวิชาทางปฏิบัติ/เชิงทักษะ
9. ใส่ใจในคุณภาพ	ส่งเสริมความถูกต้องของเนื้อหา การอ้างอิงแหล่งข้อมูล และการคิดอย่างมีระบบ	ตรวจสอบความถูกต้องของชิ้นงาน ฝึกการทบทวนและปรับปรุงคุณภาพของกระบวนการทำงาน
10. ความเรียบง่าย	ลดกิจกรรมหรือเนื้อหาที่ซ้ำซ้อน เน้นเฉพาะประเด็นสำคัญที่จำเป็น	ใช้เครื่องมือหรือทรัพยากรที่จำเป็นเท่านั้น มุ่งสู่ผลลัพธ์ที่ชัดเจนและใช้งานได้จริง
11. การจัดการตัวเองของทีม	ให้ผู้เรียนเลือกแนวทางการเรียนหรือหัวข้อย่อยตามความสนใจและเป้าหมายตนเอง	มอบหมายบทบาทในกลุ่ม เช่น หัวหน้าทีม ผู้ประสานงาน เพื่อฝึกการบริหารจัดการร่วมกัน
12. สะท้อนผลและปรับปรุง	จัดกิจกรรมเพื่อสะท้อนความเข้าใจของผู้เรียน	จัดกิจกรรมสะท้อนผลแบบเป็นทางการ เพื่อทบทวนและปรับปรุงกระบวนการทำงาน/ชิ้นงาน ต่อไป

จากตาราง 2 พบว่า หลักการทั้ง 12 ข้อของ Agile Manifesto มีศักยภาพในการยกระดับกระบวนการเรียนรู้ให้เกิดความคล่องตัวและต่อเนื่อง หลักการเหล่านี้สนับสนุนให้เกิดการเรียนรู้ที่มีความหมาย เน้นการลงมือทำ และการสะท้อนตนเองอย่างเป็นระบบ ซึ่งสอดคล้องกับผลการศึกษาของ Mironova และ คณะ (2020) ที่ระบุว่า การใช้แนวทาง Agile ช่วยเพิ่มความร่วมมือในชั้นเรียน และทำให้ผู้เรียนสามารถจัดการเวลาและงานได้อย่างมีประสิทธิภาพมากขึ้น งานวิจัยของสมพร อิศรเสนา ณ อยุธยา และคณะ (2561) สนับสนุนว่าการประยุกต์ใช้ Agile ช่วยพัฒนาทักษะการคิดวิเคราะห์ การทำงานเป็นทีม และการเรียนรู้ด้วยตนเองในผู้เรียนระดับอุดมศึกษา ขณะที่งานวิจัยของ Stewart และ คณะ (2009) และ Salza และคณะ (2019) ยืนยันว่า Agile เป็นแนวทางที่สามารถปรับใช้ได้ทั้งในระดับมัธยมศึกษาและอุดมศึกษา โดยเฉพาะในวิชาที่มีลักษณะการเปลี่ยนแปลงของเนื้อหาอย่างรวดเร็ว เช่น เทคโนโลยี การสื่อสาร หรือวิศวกรรม ดังนั้น การส่งเสริมให้ครูผู้สอนมีความเข้าใจและสามารถออกแบบกิจกรรมการเรียนรู้โดยยึดหลัก Agile จึงเป็นแนวทางสำคัญในการพัฒนาคุณภาพการศึกษาในยุคปัจจุบันและอนาคต

ในส่วนของรูปแบบ Agile Methodology ที่ได้รับความนิยมและสามารถประยุกต์ใช้ในกระบวนการเรียนรู้ ได้แก่ Scrum, Kanban, Extreme Programming (XP) และ Lean Software Development ซึ่งแต่ละแนวทางมีลักษณะเฉพาะที่สามารถเสริมสร้างกระบวนการเรียนรู้ให้มีประสิทธิภาพมากยิ่งขึ้น ดังแสดงในตารางต่อไปนี้

ตาราง 3 การเปรียบเทียบการประยุกต์ใช้หลักการ Agile Methodology ในรายวิชาทางทฤษฎีและปฏิบัติ

Agile Methodology	วิชาทางทฤษฎี/เชิงเนื้อหา	วิชาทางปฏิบัติ/เชิงทักษะ
Scrum	ให้ผู้เรียนอภิปรายหัวข้อสำคัญในแต่ละสัปดาห์ จัดทำรายงานหรือแบบฝึกหัดกลุ่มในรอบการเรียนรู้สั้นๆ	แบ่งการทำงานเป็นรอบ ฝึกปฏิบัติจริงในแต่ละรอบ โดยมีการประเมินผลงานรายรอบและสรุปการเรียนรู้ร่วมกัน
Kanban	ใช้บอร์ดช่วยในการจัดลำดับหัวข้อหรือประเด็นเนื้อหาที่จะเรียนรู้ พร้อมติดตามความก้าวหน้าของแต่ละหัวข้อ	จัดลำดับภารกิจและขั้นตอนการปฏิบัติงานอย่างชัดเจน โดยให้ผู้เรียนปรับเปลี่ยนสถานะงานของตนบนบอร์ดอย่างสม่ำเสมอ
Extreme Programming (XP)	ใช้กิจกรรมแบบจับคู่เพื่ออภิปรายแนวคิด ทบทวนความเข้าใจ และแลกเปลี่ยนข้อเสนอแนะในระดับเนื้อหา	เน้นการทำงานร่วมกันแบบคู่ในการเขียนชุดคำสั่ง / ชิ้นงานย่อย การทดสอบ และการปรับปรุงผลงานแบบต่อเนื่องด้วยการสะท้อนผลระหว่างการทำงาน
Lean Software Development	ตัดกิจกรรมซ้ำซ้อน เช่น แบบฝึกหัดที่ไม่มีผลต่อความเข้าใจ มุ่งเน้นเฉพาะเนื้อหาหลักและกิจกรรมที่สร้างคุณค่า	ออกแบบกระบวนการฝึกปฏิบัติให้มีประสิทธิภาพ โดยลดเวลาสูญเปล่า และส่งเสริมให้ผู้เรียนเน้นงานที่สำคัญต่อผลลัพธ์จริง

จากตาราง 3 พบว่า แนวทางของ Agile Methodology มีลักษณะการประยุกต์ใช้ที่เหมาะสมแตกต่างกันไปตามบริบทของรายวิชา หากเป็นรายวิชาที่มุ่งเน้นการสร้างความรู้ความเข้าใจในเนื้อหา การใช้ Scrum และ Kanban จะช่วยให้ผู้เรียนสามารถจัดระบบการเรียนรู้และติดตามความก้าวหน้าได้อย่างมีทิศทาง ขณะเดียวกัน การใช้ XP จะเพิ่มคุณค่าในการสร้างปฏิสัมพันธ์และการสะท้อนความคิดภายในกลุ่มผู้เรียน ส่วนแนวคิดแบบ Lean Software Development (หรือในบริบทการเรียนการสอนอาจใช้คำว่า Lean Learning) จะช่วยลดภาระที่ไม่จำเป็น และส่งเสริมให้เนื้อหาเข้มข้นขึ้นอย่างมีเป้าหมาย

ในขณะที่รายวิชาที่เน้นทักษะปฏิบัติ แนวทาง Agile สามารถเพิ่มประสิทธิภาพในการทำงานเป็นทีม การฝึกฝนซ้ำ และการวิเคราะห์ผลลัพธ์จริงในระยะสั้น ซึ่งเห็นได้ชัดใน Scrum และ XP ที่เน้นรอบของการฝึกปฏิบัติ และการสะท้อนผลอย่างต่อเนื่อง Kanban มีบทบาทในการวางแผนขั้นตอนงานและการติดตามสถานะของงานมอบหมายงาน ส่วนแนวคิดแบบ Lean Software Development จะช่วยจัดระเบียบและลดความซ้ำซ้อนในกระบวนการฝึกทักษะ ทำให้ผู้เรียนสามารถมุ่งเน้นกับงานที่มีคุณค่าได้ดียิ่งขึ้น

แนวทางทั้งหมดนี้แสดงให้เห็นถึงความยืดหยุ่นของ Agile ในการออกแบบการเรียนรู้ที่ตอบสนองต่อความแตกต่างของผู้เรียน ทั้งในด้านเป้าหมายการเรียนรู้ วิธีการ และผลลัพธ์ที่ต้องการ ซึ่งสอดคล้องกับข้อเสนอแนะของงานวิจัยหลายชิ้น เช่น งานวิจัยของ Salza และ คณะ (2019) ที่ชี้ให้เห็นว่า Scrum และ Kanban ช่วยเสริมสร้างความสามารถในการจัดการงานและความร่วมมือในกลุ่มผู้เรียนได้อย่างมีประสิทธิภาพ รวมถึงงานวิจัยของ Mironova และ คณะ (2020) ที่เน้นการใช้ XP เพื่อพัฒนาทักษะการเขียนโปรแกรมผ่านการทำงานแบบมีส่วนร่วม ขณะที่ Hoda และ คณะ (2018) ยังเสนอว่าแนวทาง Lean Software Development ส่งผลดีต่อการลดภาระงานที่ไม่จำเป็นในบริบทของการศึกษาและส่งเสริมการเรียนรู้ที่มีเป้าหมายชัดเจนในการออกแบบการเรียนรู้ที่ตอบสนองต่อความแตกต่างของผู้เรียน ทั้งในด้านเป้าหมายการเรียนรู้ วิธีการ และผลลัพธ์ที่ต้องการ

กรณีศึกษาการจัดการกิจกรรมการเรียนรู้ด้วยแนวคิด Agile Methodology

เพื่อส่งเสริมการจัดการเรียนรู้ที่มุ่งเน้นผู้เรียนเป็นศูนย์กลาง และสอดคล้องกับหลักการของ Agile Methodology ผู้เขียนได้ออกแบบการจัดการเรียนการสอน ในรายวิชาการเขียนโปรแกรมคอมพิวเตอร์ สำหรับนักศึกษาชั้นปีที่ 1 คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยนอร์ท-เชียงใหม่ โดยมีเป้าหมายเพื่อพัฒนาทั้งทักษะทางเทคนิคและทักษะการทำงานร่วมกันของผู้เรียนผ่านการเรียนรู้ที่ยืดหยุ่นและเน้นการมีส่วนร่วม ดังนี้

1. Agile Methodology : Scrum

ออกแบบการเรียนรู้ในรายวิชาให้ใช้โครงสร้างแบบรอบการเรียนรู้ (Sprint) โดยแต่ละรอบ มีระยะเวลาประมาณ 1-2 สัปดาห์และมีหัวข้อที่ชัดเจน เช่น รอบการเรียนรู้ที่ 1 เรื่องตัวแปรและชนิดข้อมูล, รอบการเรียนรู้ที่ 2 เรื่องโครงสร้างควบคุม, รอบการเรียนรู้ที่ 3 เรื่องฟังก์ชันและโมดูล และรอบการเรียนรู้ที่ 4 เรื่องการประมวลผลข้อมูล โดยในแต่ละรอบการเรียนรู้ นักศึกษาจะได้มีส่วนร่วมในการจัดทำโครงงานกลุ่มขนาดเล็กเพื่อฝึกฝนการทำงานร่วมกันและพัฒนาทักษะเชิงปฏิบัติ โดยเริ่มจากการวางแผนงานร่วมกันในกิจกรรม การวางแผนรอบการเรียนรู้ (Sprint Planning) เช่น การแบ่งหน้าที่ และกำหนดเป้าหมายรายสัปดาห์ หลังจากนั้นเข้าสู่กระบวนการเขียนโปรแกรมแบบภายใต้เวลาจำกัด และในช่วงปลายสัปดาห์จะมีการนำเสนองาน พร้อมทั้งทำกิจกรรมสะท้อนการทำงานที่ผ่านมาของกลุ่มเกี่ยวกับสิ่งที่ได้เรียนรู้และสิ่งที่ควรปรับปรุงเพื่อสะท้อนกระบวนการทำงานร่วมกัน ทบทวนสิ่งที่ทำได้ดี และระบุจุดที่ควรปรับปรุงในรอบถัดไป ซึ่งกระบวนการเรียนรู้นี้ดังกล่าว พบว่าช่วยให้นักศึกษาได้ฝึกการวางแผนและดำเนินการเรียนรู้อย่างเป็นระบบ เรียนรู้การตั้งเป้าหมายที่ชัดเจนในแต่ละช่วงเวลารวมถึงบริหารเวลาให้สัมพันธ์กับภาระงานได้ดีขึ้น

2. Agile Methodology : Kanban

ออกแบบการเรียนรู้โดยได้มอบหมายให้นักศึกษาใช้ Kanban Board เพื่อแบ่งขั้นตอนการทำงานออกเป็น 3 ลำดับ ได้แก่ งานที่ต้องทำ งานที่กำลังดำเนินการ และงานที่เสร็จสมบูรณ์ โดยในแต่ละขั้นตอนจะสะท้อนถึงสถานะของกิจกรรมต่างๆ เช่น การพัฒนาโปรแกรมคำนวณค่าเฉลี่ยของข้อมูล การออกแบบฟังก์ชันตรวจสอบรหัสผ่าน และการจัดการข้อมูลในรูปแบบตาราง แนวทางนี้สอดคล้องกับหลักการคิดแบบคล่องตัวที่ให้ความสำคัญ

กับการแสดงผลลัพธ์ที่จับต้องได้เป็นระยะ และเน้นการทำงานที่มีโครงสร้างเรียบง่ายเพื่อเพิ่มประสิทธิภาพในการเรียนรู้ กระบวนการดังกล่าวพบว่ามีส่วนช่วยให้นักศึกษาได้พัฒนาทักษะสำคัญทั้งในด้านการวางแผน การลำดับความสำคัญของงาน การทำงานร่วมกับผู้อื่น และการติดตามความก้าวหน้าอย่างเป็นระบบ ส่งผลให้เกิดความรับผิดชอบต่องานที่ได้รับมอบหมายมากขึ้นพร้อมทั้งส่งเสริมการเรียนรู้แบบมีเป้าหมาย

3. Agile Methodology : Extreme Programming (XP)

ในบางช่วงของรอบการเรียนรู้ (Sprint) ได้มอบหมายให้นักศึกษาจับคู่เพื่อทำกิจกรรม/เขียนโปรแกรม (Pair Programming) โดยกำหนดบทบาท Driver และ Navigator และสลับกันทุกครั้งที่แก้ไขโปรแกรมสำเร็จ พร้อมทั้งทำกิจกรรมแลกเปลี่ยนและตรวจสอบชุดคำสั่งร่วมกันในกลุ่มย่อยและให้ข้อเสนอแนะในกลุ่มย่อยโดยใช้เกณฑ์ เช่น ความถูกต้อง ความชัดเจนของชุดคำสั่ง และการจัดโครงสร้างอย่างมีระบบ กิจกรรมเหล่านี้สอดคล้องกับหลักการที่เน้นการสร้างสิ่งแวดล้อมที่มีแรงจูงใจและสนับสนุนให้ทำงานได้จริง และส่งเสริมให้ทีมสามารถจัดการตัวเองและพัฒนาวิธีการทำงานร่วมกันได้อย่างมีประสิทธิภาพ กิจกรรมดังกล่าวพบว่านักศึกษาส่วนใหญ่สามารถแสดงออกถึงความเข้าใจในกระบวนการทำงานร่วมกันได้อย่างชัดเจน มีการแลกเปลี่ยนความคิดเห็นอย่างมีระบบ กล่าวตั้งคำถามและเสนอแนวทางแก้ไขปัญหาร่วมกับผู้อื่น

4. Agile Methodology : Lean Software Development

แนวคิด Lean Software Development ถูกนำมาใช้เป็นแนวทางในการออกแบบกิจกรรมการเรียนรู้ที่มุ่งเน้นการลดความซับซ้อน ลดขั้นตอนที่ไม่จำเป็น และส่งเสริมให้นักศึกษาพัฒนาชุดคำสั่งที่มีคุณค่าและสอดคล้องกับวัตถุประสงค์ของโปรแกรมอย่างแท้จริง กิจกรรมหลักที่ใช้ได้แก่ การออกแบบฟังก์ชันให้มีโครงสร้างกระชับ การแยกกระบวนการย่อย เช่น การประมวลผลข้อมูลหรือการตรวจสอบเงื่อนไข ออกจากฟังก์ชันหลัก เพื่อให้สามารถตรวจสอบและนำกลับมาใช้ซ้ำได้ง่าย รวมถึงการออกแบบส่วนจัดการข้อผิดพลาดให้มีความชัดเจนและมีประสิทธิภาพ และฝึกเลือกใช้แนวทางการเขียนโปรแกรมที่เรียบง่ายแต่ยังคงประสิทธิภาพของระบบไว้ได้ แนวปฏิบัตินี้สอดคล้องกับหลักการซึ่งเน้นความเป็นเลิศทางเทคนิค และให้ความสำคัญกับความเรียบง่ายในกระบวนการทำงาน ผลจากการดำเนินกิจกรรมดังกล่าวพบว่านักศึกษามีการพัฒนาทักษะการคิดอย่างมีระบบ เห็นความสำคัญของการเขียนโปรแกรมที่ชัดเจนและยั่งยืน และสามารถตัดสินใจเลือกโครงสร้างที่เหมาะสมกับงานแต่ละประเภทได้อย่างมีเหตุผล นอกจากนี้ยังแสดงให้เห็นถึงความเข้าใจที่ลึกซึ้งเกี่ยวกับความสัมพันธ์ระหว่างความเรียบง่ายของชุดคำสั่งกับประสิทธิภาพโดยรวมของระบบ ซึ่งสะท้อนถึงศักยภาพของผู้เรียนในการประยุกต์ใช้แนวคิดทางเทคนิคในสถานการณ์จริงได้อย่างเหมาะสม

สรุป

Agile เป็นแนวคิดที่ริเริ่มจากการพัฒนาระบบและซอฟต์แวร์ โดยมีเป้าหมายเพื่อเพิ่มประสิทธิภาพในการดำเนินงานให้สามารถตอบสนองต่อความเปลี่ยนแปลงได้อย่างรวดเร็ว ยืดหยุ่น และมีความต่อเนื่อง โดยเน้นการทำงานร่วมกันของทีม การส่งมอบงานเป็นรอบ และการปรับปรุงอย่างสม่ำเสมอ เมื่อประยุกต์ใช้แนวคิด Agile ใน

บริบทของการจัดการเรียนการสอน ทั้งในด้านของ Agile Manifesto และ หลักการ 12 ข้อ รวมถึง Agile Methodology พบว่าแนวคิดนี้สามารถสนับสนุนการเรียนรู้ที่มีผู้เรียนเป็นศูนย์กลาง ผ่านการออกแบบกิจกรรมที่ยืดหยุ่น การสะท้อนผลอย่างต่อเนื่อง และการเรียนรู้จากกระบวนการมากกว่าผลลัพธ์เพียงอย่างเดียว ซึ่งช่วยส่งเสริมการคิดวิเคราะห์ การแก้ปัญหา และการเรียนรู้ร่วมกันของผู้เรียนได้อย่างเป็นระบบ

จากเนื้อหาที่ได้นำเสนอในบทความนี้ จะเห็นได้ว่า Agile เป็นแนวคิดที่มีศักยภาพสูงในการยกระดับคุณภาพการจัดการเรียนรู้ ในรายวิชาทั้งในเชิงทฤษฎีและปฏิบัติ ในกรณีศึกษาการบูรณาการแนวคิด Agile Methodology ในการจัดการเรียนการสอนรายวิชาการเขียนโปรแกรมคอมพิวเตอร์ ได้ออกแบบให้ครอบคลุมทั้งด้านกระบวนการเรียนรู้และการพัฒนาทักษะผู้เรียนอย่างรอบด้าน โดยใช้แนวทาง Scrum ในการจัดโครงสร้างการเรียนรู้แบบรอบการเรียนรู้ (Sprint) เพื่อฝึกการวางแผน การตั้งเป้าหมาย และการสะท้อนผลการเรียนรู้ ใช้ Kanban Board เพื่อส่งเสริมการจัดลำดับความสำคัญของงานและการติดตามความคืบหน้าอย่างเป็นระบบ ผสานกิจกรรม Pair Programming และการแลกเปลี่ยนความคิดเห็นตามแนวทาง Extreme Programming (XP) เพื่อเสริมสร้างทักษะการทำงานเป็นทีม การสื่อสาร และการตรวจสอบชุดคำสั่งอย่างมีแบบแผน และนำแนวคิด Lean Software Development มาใช้ในการฝึกให้นักศึกษาคิดวิเคราะห์เชิงโครงสร้าง ลดความซับซ้อนของชุดคำสั่ง และมุ่งเน้นการพัฒนาโปรแกรมอย่างมีประสิทธิภาพและมีคุณค่า แนวปฏิบัติทั้งสี่ประการนี้มีส่วนสำคัญในการส่งเสริมการเรียนรู้ที่ยั่งยืน พัฒนาทักษะแห่งศตวรรษที่ 21 และสร้างผู้เรียนที่มีศักยภาพในการประยุกต์ใช้ความรู้ในบริบทจริงได้อย่างเหมาะสม

อย่างไรก็ตามการนำไปใช้ในบริบททางการศึกษาอาจต้องมีการปรับให้เหมาะสมกับบริบทของผู้เรียน ผู้สอน และสภาพแวดล้อมการเรียนรู้แต่ละแห่ง แต่ด้วยหลักการที่ยืดหยุ่นของ Agile ก็สามารถทำให้เกิดผลลัพธ์ที่ส่งเสริมการเรียนรู้ที่มีความหมายและยั่งยืนได้อย่างแท้จริง อย่างไรก็ตามแนวทางจัดการเรียนรู้ด้วย Agile ยังมีจุดที่ต้องให้ความสำคัญ โดยเฉพาะอย่างยิ่งในด้านความเข้าใจของผู้สอนหรือนักออกแบบการเรียนการสอน จำเป็นต้องมีความรู้ความเข้าใจอย่างลึกซึ้งเกี่ยวกับหลักการของ Agile จึงจะสามารถออกแบบกิจกรรมการเรียนรู้ให้สอดคล้องกับแนวคิดดังกล่าวได้อย่างมีประสิทธิภาพ

เอกสารอ้างอิง

- สมพร อิศรเสนา ณ อยุธยา, วิไลรัตน์ หาญศรีวัฒนา, & สมจิตต์ ศรีสมบูรณ์. (2561). การพัฒนาแนวทางการจัดการเรียนรู้แบบคล่องตัว (Agile Learning Management) สำหรับศตวรรษที่ 21. *วารสารวิชาการมหาวิทยาลัยราชภัฏวไลยอลงกรณ์ ในพระบรมราชูปถัมภ์*, 13(2), 14–27.
- Ahmad, M. O., Markkula, J., & Oivo, M. (2013). *Kanban in software engineering: A systematic literature review*. Proceedings of the 39th EUROMICRO Conference on Software Engineering and Advanced Applications, 9–16. <https://doi.org/10.1109/SEAA.2013.10>

- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved April 3, 2025, from <https://agilemanifesto.org/>
- Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2018). Systematic literature reviews in agile software development: A tertiary study. *Information and Software Technology*, 85, 60–70. <https://doi.org/10.1016/j.infsof.2017.10.008>
- Mironova, V., Pyroh, M., & Harko, I. (2020, December). *Methodology of building agile-education processes in higher education institutions*. In IT&I 2020: Information Technology and Interactions (pp. 409–417). CEUR Workshop Proceedings. <http://ceur-ws.org/Vol-2753/paper38.pdf>
- Salza, P., Guardabascio, F., & Carnevale, L. (2019). Teaching agile methods using a Scrum-based approach in higher education. *Journal of Applied Computing*, 15(4), 75–88.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Retrieved April 3, 2025, from <https://www.scrumguides.org/>
- Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- Shastri, Y., Hoda, R., & Amor, R. (2021). *Understanding the roles of artifacts in agile software development*. *Information and Software Technology*, 131, 106513. <https://doi.org/10.1016/j.infsof.2020.106513>
- Stewart, D., Smith, R., & Shields, J. (2009). Implementing agile in secondary school IT education. *Australian Educational Computing*, 24(1), 18–25.